

Pregrado

CARRERA: SISTEMAS Y GESTIÓN DE DATA

ASIGNATURA (UIC): EJECUCIÓN DE PROYECTOS EN TIC'S

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL

TÍTULO EN:

TECNÓLOGO UNIVERSITARIO EN SISTEMAS Y

GESTIÓN DE DATA

TEMA: DESARROLLO DE UNA APLICACIÓN WEB PARA

LA GESTIÓN Y CONTROL DE MUESTRAS MÉDICAS

EN LA EMPRESA HOSPIMEDIKKA.

AUTOR/S: STALIN VINICIO REASCOS MANTILLA

RICARDO ANDRÉS SANTAMARÍA ROSARIO

TUTOR: MG. DANNY SANTIAGO PÁEZ OSCULLO

FECHA: SEPTIEMBRE 2025



Autor:

Santamaría Rosario Ricardo Andrés



Título a obtener: Tecnólogo Universitario en Sistemas y Gestión de Data.

Matriz: Sangolquí -Ecuador

Correo electrónico: ricardo.santamaria@ister.edu.ec

Autor:

Reascos Mantilla Stalin Vinicio



Título a obtener: Tecnólogo Universitario en Sistemas y Gestión de Data.

Matriz: Sangolquí -Ecuador

Correo electrónico: stalin.reascos@ister.edu.ec

Dirigido por:

Páez Oscullo Danny Santiago



Título: Ing. Computación Gráfica.

Mg. Lógica, Computación e Inteligencia Artificial.

Matriz: Sangolquí -Ecuador

Correo electrónico: danny.paez@ister.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

©2025 Tecnológico Universitario

Rumiñahui SANGOLQUÍ –

ECUADOR

Reascos Mantilla Stalin Vinicio

Ricardo Andrés Santamaría Rosario

***DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN Y CONTROL DE
MUESTRAS MÉDICAS EN LA EMPRESA HOSPIMEDIKKA.***



CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.3

Sangolquí, 12 de septiembre del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

**INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE
UNIVERSITARIO**

Presente

Por medio de la presente, yo, Santamaría Rosario Ricardo Andrés declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado Desarrollo de un Sistema Web para la Gestión y Control de Muestras Médicas en la Empresa HOSPIMEDIKKA, de la Tecnología Superior Universitaria Sistemas y Gestión de Data; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,

RICARDO ANDRÉS SANTAMARÍA ROSARIO
C.I.: 1723820872

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero
Telf: 0960052734 / 023524576 / 022331628
www.ister.edu.ec / info@ister.edu.ec



CARTA DE CESIÓN DE DERECHOS DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-2-2.3

Sangolquí, 12 de septiembre del 2025

MSc. Elizabeth Ordoñez
DIRECTORA DE DOCENCIA

MSc. Mónica Loachamín
COORDINADORA DE TITULACIÓN

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Por medio de la presente, yo, Reascos Mantilla Stalin Vinicio declaro y acepto en forma expresa lo siguiente: Ser autor del trabajo de titulación denominado Desarrollo de un Sistema Web para la Gestión y Control de Muestras Médicas en la Empresa HOSPIMEDIKKA, de la Tecnología Superior Universitaria Sistemas y Gestión de Data; y a su vez manifiesto mi voluntad de ceder al Instituto Superior Tecnológico Rumiñahui con condición de Universitario los derechos de reproducción, distribución y publicación de dicho trabajo de titulación, en cualquier formato y medio, con fines académicos y de investigación.

Esta cesión se otorga de manera no exclusiva y por un periodo indeterminado. Sin embargo, conservo los derechos morales sobre mi obra.

En fe de lo cual, firmo la presente.

Atentamente,

REASCOS MANTILLA STALIN VINICIO
C.I.: 1719307967

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero
Telf: 0960052734 / 023524576 / 022331628
www.ister.edu.ec / info@ister.edu.ec



FORMULARIO PARA ENTREGA DE PROYECTOS EN BIBLIOTECA INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

CT-ANX-2025-ISTER-3

CARRERA: TECNOLOGÍA UNIVERSITARIA EN SISTEMAS Y GESTIÓN DE DATA

AUTOR/ES: STALIN VINICIO REASCOS MANTILLA – RICARDO ANDRÉS SANTAMARÍA ROSARIO

TUTOR: DANNY SANTIAGO PAEZ OSCULLO

CONTACTO ESTUDIANTE: 0983443313 - 0983862931

**CORREO ELECTRÓNICO: andrew14.94santamaria@outlook.com,
stalin_c2k@hotmail.com**

TEMA: DESARROLLO DE UN SISTEMA WEB PARA LA GESTIÓN Y CONTROL DE MUESTRAS MÉDICAS EN LA EMPRESA HOSPIMEDIKKA.

OPCIÓN DE TITULACIÓN: UNIDAD DE INTEGRACIÓN CURRICULAR

RESUMEN:

El presente proyecto de titulación tiene como finalidad el diseño e implementación de un sistema web para la gestión y control de muestras médicas en la empresa HOSPIMEDIKKA, dedicada a la comercialización de productos farmacéuticos. La problemática principal identificada se relaciona con la administración manual de muestras a través de Kardex físicos, procedimiento que genera errores en los registros, pérdida de información, falta de trazabilidad y demoras en la entrega de insumos a los representantes de ventas.

La solución propuesta consiste en el desarrollo de una aplicación web basada en la arquitectura Modelo–Vista–Controlador (MVC), empleando tecnologías como Laravel, MySQL, TailwindCSS y JavaScript, que permite registrar, controlar y monitorear en tiempo real el ingreso, egreso e inventario de las muestras médicas. Asimismo, el sistema genera

reportes automatizados que fortalecen la auditoría y la transparencia, la toma de decisiones y la transparencia en los procesos internos.

La metodología aplicada combina enfoques cuantitativos y cualitativos, a través de la recopilación de datos objetivos como tiempos de proceso y reducción de errores observación directa y validación con los usuarios finales. Se aplicaron pruebas funcionales y de usabilidad que confirmaron la eficacia del sistema, evidenciando mejoras significativas en la precisión de los registros, la rapidez de los procesos y la satisfacción de los usuarios.

La implementación de este sistema web contribuye a la transformación digital de la empresa, optimizando el flujo de trabajo, mejorando la satisfacción de los usuarios y asegurando un control más eficiente y confiable de los recursos estratégicos para la actividad comercial de HOSPIMEDIKKA.

PALABRAS CLAVE:

Sistema Web, Laravel, Gestión de Muestras Médicas, Usabilidad, Automatización, Base de Datos, Arquitectura MVC.

ABSTRACT:

The present graduation project aims to design and implement a web system for the management and control of medical samples at HOSPIMEDIKKA, a company dedicated to the marketing of pharmaceutical products. The main issue identified relates to the manual administration of samples through physical Kardex, a procedure that generates errors in records, loss of information, lack of traceability, and delays in delivering supplies to sales representatives.

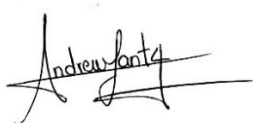
The proposed solution consists of developing a web application based on the Model-View-Controller (MVC) architecture, using technologies such as Laravel, MySQL, Tailwind

CSS, and JavaScript. This application allows for real-time registration, control, and monitoring of the entry, exit, and inventory of medical samples. Additionally, the system generates automated reports that enhance auditing, decision-making, and transparency in internal processes.

The methodology applied combines quantitative and qualitative approaches through the collection of objective data such as process times and error reduction, direct observation, and validation with end users. Functional and usability tests were conducted, confirming the system's effectiveness, demonstrating significant improvements in record accuracy, process speed, and user satisfaction.

The implementation of this web system contributes to the digital transformation of the company, optimizing workflow, enhancing user satisfaction, and ensuring more efficient and reliable control of strategic resources for HOSPIMEDIKKA's commercial activities.

Keywords: Web System, Medical Samples Management, Traceability, Usability, Automation, Digital Transformation.



Ricardo Andrés Santamaría Rosario

C.I.: 1723820872



Stalin Vinicio Reascos Mantilla

C.I.: 1719307967



SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 12 de septiembre del 2025

Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo, RICARDO ANDRÉS SANTAMARÍA ROSARIO, con C.I.: 1723820872 alumno de la Carrera TECNÓLOGO UNIVERSITARIO EN SISTEMAS Y GESTIÓN DE DATA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,

Firma del Estudiante

Ricardo Andrés Santamaría Rosario

C.I.: 1723820872

MATRIZ SANGOLQUÍ: Av. Atahualpa 1701 y 8 de Febrero

Telf: 0960052734 / 023524576 / 022331628

f i t t www.ister.edu.ec / info@ister.edu.ec



SOLICITUD DE PUBLICACIÓN DEL TRABAJO DE TITULACIÓN

CT-ANX-2025-ISTER-4

Sangolquí, 12 de septiembre del 2025

Sres.-

INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO

Presente

Yo, STALIN VINICIO REASCOS MANTILLA, con C.I.: 1719307967 alumno de la Carrera TECNÓLOGO UNIVERSITARIO EN SISTEMAS Y GESTIÓN DE DATA, cedo al INSTITUTO SUPERIOR TECNOLÓGICO RUMIÑAHUI CON CONDICIÓN DE UNIVERSITARIO, los derechos de publicaciones del presente trabajo de Titulación en el Repositorio Institucional para hacer uso de todos los contenidos con fines estrictamente académico o de investigación.

Atentamente,

Firma del Estudiante
Stalin Vinicio Reascos Mantilla
C.I.: 1719307967

AGRADECIMIENTOS

A Dios, por acompañarme en cada paso de este camino, dándome la fuerza, la sabiduría y la paciencia para nunca rendirme.

A mi familia, por ser mi pilar fundamental, por su amor incondicional y por enseñarme con su ejemplo el valor del esfuerzo y la dedicación. Gracias por creer siempre en mí y por apoyarme incluso en los momentos más difíciles.

A mi pareja, por estar a mi lado con comprensión, cariño y aliento en cada etapa de este proceso. Tu compañía ha sido un motor de motivación y una razón más para superarme día a día.

A mi ahijada, que con su ternura y alegría me recuerda constantemente la importancia de seguir luchando por mis sueños y ser un ejemplo para ella.

A mis profesores, quienes con paciencia y vocación compartieron sus conocimientos y me guiaron durante mi formación académica. Sus enseñanzas han dejado una huella imborrable en mi vida profesional y personal.

Y a mis compañeros de clases, con quienes compartí aprendizajes, esfuerzos y momentos inolvidables. Gracias por su amistad, apoyo y por hacer de este camino una experiencia enriquecedora y especial.

Ricardo Andrés Santamaría Rosario

Quiero expresar mis sinceros agradecimientos a todas las personas que hicieron posible la culminación de este proyecto de grado, el cual representa no solo un logro académico, sino también un desarrollo personal.

Primero quiero agradecer a mi familia quienes han sido mi principal fuente de apoyo durante todo este proceso, con su comprensión, paciencia y respaldo incondicional fueron fundamentales para no rendirme incluso cuando hubo momentos difíciles.

Como agradecimiento especial, dedico este logro a mi hija, que ha sido mi mayor motivación y razón para seguir adelante cada día.

Adicional agradezco profundamente al profesor Danny Páez (tutor de este trabajo) por su orientación, disposición y valiosas sugerencias que enriquecieron el desarrollo del proyecto en donde su acompañamiento fue clave fundamental para lograr los objetivos propuestos.

También reconozco el esfuerzo que implicó compaginar el trabajo con los estudios, no fue un camino fácil por recorrer, pero cada sacrificio ha valido la pena, este proyecto es también testimonio de que, con disciplina y compromiso es posible avanzar y cumplir las metas que uno se propone.

Finalmente, agradezco a todas las personas que de una u otra manera contribuyeron a la realización de este trabajo en donde cada palabra de aliento, cada gesto de ayuda, ha dejado una enorme huella en este arduo camino.

Gracias a todos por ser parte de este logro.

Stalin Vinicio Reascos Mantilla

DEDICATORIA

Dedico el siguiente proyecto a mi familia por brindarme su apoyo incondicional, por enseñarme que nunca hay que darse por vencido, que todo se lo soluciona con esfuerzo y trabajo constante, dedico este logro a mi pareja quien me ha motivado a ser mejor persona y que a pesar de todo sigue dándome la mano. A mi ahijada Briana quien fue el motor para conseguir esto, su dulzura me ayuda a ser mejor. A cada profesor que me ha brindado sus enseñanzas porque sin ellos igual no hubiese logrado esto.

Ricardo Andrés Santamaría Rosario

Dedico este proyecto a mi familia por su amor incondicional y por enseñarme a ser perseverante en mis sueños y metas, a mis docentes por toda la guía proporcionada durante el transcurso de la carrera, adicional y de manera especial a nuestro tutor Ing. Danny Paez que gracias a sus consejos y recomendaciones pudimos concluir de manera exitosa el desarrollo de nuestro proyecto.

Que este sistema web local contribuya a simplificar su gestión, a fortalecer la toma de decisiones y, sobre todo brinde una atención más eficiente en el registro, control y entrega de muestras medicas al personal Representante de Ventas dentro de la Empresa Hospimedikka Cía. Ltda.

Stalin Vinicio Reascos Mantilla

RESUMEN

El presente proyecto de titulación tiene como finalidad el diseño e implementación de un sistema web para la gestión y control de muestras médicas en la empresa HOSPIMEDIKKA, dedicada a la comercialización de productos farmacéuticos. La problemática principal identificada se relaciona con la administración manual de muestras a través de Kardex físicos, procedimiento que genera errores en los registros, pérdida de información, falta de trazabilidad y demoras en la entrega de insumos a los representantes de ventas.

La solución propuesta consiste en el desarrollo de una aplicación web basada en la arquitectura Modelo–Vista–Controlador (MVC), empleando tecnologías como Laravel, MySQL, TailwindCSS y JavaScript, que permite registrar, controlar y monitorear en tiempo real el ingreso, egreso e inventario de las muestras médicas. Asimismo, el sistema genera reportes automatizados que fortalecen la auditoría y la transparencia, la toma de decisiones y la transparencia en los procesos internos.

La metodología aplicada combina enfoques cuantitativos y cualitativos, a través de la recopilación de datos objetivos como tiempos de proceso y reducción de errores observación directa y validación con los usuarios finales. Se aplicaron pruebas funcionales y de usabilidad que confirmaron la eficacia del sistema, evidenciando mejoras significativas en la precisión de los registros, la rapidez de los procesos y la satisfacción de los usuarios.

La implementación de este sistema web contribuye a la transformación digital de la empresa, optimizando el flujo de trabajo, mejorando la satisfacción de los usuarios y asegurando un control más eficiente y confiable de los recursos estratégicos para la actividad comercial de HOSPIMEDIKKA.

PALABRAS CLAVE:

Sistema Web, Laravel, Gestión de Muestras Médicas, Usabilidad, Automatización, Base de Datos, Arquitectura MVC.

ABSTRACT

The present graduation project aims to design and implement a web system for the management and control of medical samples at HOSPIMEDIKKA, a company dedicated to the marketing of pharmaceutical products. The main issue identified relates to the manual administration of samples through physical Kardex, a procedure that generates errors in records, loss of information, lack of traceability, and delays in delivering supplies to sales representatives.

The proposed solution consists of developing a web application based on the Model-View-Controller (MVC) architecture, using technologies such as Laravel, MySQL, Tailwind CSS, and JavaScript. This application allows for real-time registration, control, and monitoring of the entry, exit, and inventory of medical samples. Additionally, the system generates automated reports that enhance auditing, decision-making, and transparency in internal processes.

The methodology applied combines quantitative and qualitative approaches through the collection of objective data such as process times and error reduction, direct observation, and validation with end users. Functional and usability tests were conducted, confirming the system's effectiveness, demonstrating significant improvements in record accuracy, process speed, and user satisfaction.

The implementation of this web system contributes to the digital transformation of the company, optimizing workflow, enhancing user satisfaction, and ensuring more efficient and reliable control of strategic resources for HOSPIMEDIKKA's commercial activities.

Keywords: Web System, Medical Samples Management, Traceability, Usability, Automation, Digital Transformation.

TABLA DE CONTENIDO

AGRADECIMIENTOS	11
DEDICATORIA.....	13
RESUMEN.....	14
ABSTRACT	15
ÍNDICE DE TABLAS	27
INTRODUCCIÓN	1
ANTECEDENTES.....	2
PROBLEMA	4
JUSTIFICACIÓN.....	5
Justificación Social.....	5
Justificación Técnica	5
OBJETIVOS.....	7
Objetivo General	7
Objetivos Específicos	7
ALCANCE	8
MARCOS REFERENCIALES	9
MARCOS LEGALES	10
MARCO POLÍTICO-SOCIAL	11
CAPÍTULO I.....	12
MARCO TEÓRICO	12

1.1	Trazabilidad y Gestión de Inventario	12
1.1.1	Trazabilidad.....	12
1.1.2	Importancia de la Trazabilidad.....	12
1.1.3	Trazabilidad en la Cadena de Suministros Farmacéutica.....	13
1.2	Sistemas de Inventarios	13
1.3	Tipos de Sistemas de Inventarios	13
1.3.1	Sistema de Inventario Perpetuo.....	13
1.3.2	Sistema de Inventario Periódico.....	14
1.4	Kardex	15
1.5	Arquitectura de Software para Aplicación de Gestión.....	15
1.5.1	El Patrón Modelo-Vista-Controlador (MVC)	15
1.5.2	Justificación del uso de Laravel	15
1.6	Stack Tecnológico para el Sistema de Gestión de Muestras Médicas	16
1.6.1	Visual Studio Code.....	16
1.6.2	Laragon.....	16
1.6.3	Backend PHP y el Framework Laravel	16
1.6.3.1	Framework	16
1.6.3.2	Laravel.....	17
1.6.4	Frontend	17
1.6.4.1	JavaScript y jQuery	17

1.6.4.2	TailwindCSS.....	17
1.6.5	Base de Datos MySQL	17
CAPÍTULO II		19
METODOLOGÍA		19
2.1	Tipo y Enfoque de Investigación	19
2.2	Método Cuantitativo.....	20
2.3	Método Cualitativo.....	21
2.4	Fases de Desarrollo del Proyecto	22
2.5	Estructura	22
2.5.1	Fase 1: Análisis y Levantamiento de Requisitos.....	22
2.5.2	Fase 2: Diseño del Sistema.....	23
2.5.3	Fase 3: Desarrollo e Implementación.....	23
2.5.4	Fase 4: Pruebas y Validación	23
2.5.5	Fase 5: Entrega y Capacitación	23
2.6	Población y Muestra.....	23
2.6.1	Población.....	23
2.6.2	Muestra.....	24
2.7	PLANTEAMIENTO DE LAS VARIABLES.....	24
2.7.1	Variable Independiente (Implementación del Sistema Web de Gestión)	24
2.7.1.1	Valores	24

2.7.2	Variable Dependiente (Eficiencia del Proceso de gestión de Muestras Médicas)	24
2.8	Indicadores Medibles	25
2.8.1	Tiempo de Proceso	25
2.8.2	Precisión del Registro.....	25
2.8.3	Nivel de Trazabilidad	25
2.9	Discusión del problema	25
2.10	PRUEBAS Y RESULTADOS	26
2.10.1	Pruebas de Usabilidad Ingreso, Consulta y Generación de Reporte.	27
2.10.2	Pruebas de Usabilidad del Sistema Web Local	28
2.11	Acceso Usuario Administrador	28
2.12	Acceso Usuario Bodega	34
	CAPITULO III	39
	PROPUESTA DEL DESARROLLO DEL PROYECTO TÉCNICO / ESTUDIO DE CASOS.	39
3.1	Modelo	39
3.2	Vista	40
3.3	Controlador.....	40
3.4	Justificación del uso de Laravel	40
3.5	Repercusión dentro del Proyecto.....	41
3.6	Diagrama Entidad Relación	41

3.7	DISEÑO DE LA INTERFAZ Y EXPERIENCIA DE USUARIO (UI/UX) ...	43
3.8	DESARROLLO DE MÓDULOS PRINCIPALES	44
3.8.1	Desarrollo de Funcionalidades CRUD	44
3.9	Archivos de Migraciones	45
3.10	Controladores	50
	CONCLUSIONES	55
	RECOMENDACIONES	57
	A futuro (expandir el diseño para un dispositivo móvil).....	57
	Práctica (capacitación al personal y levantamiento de un servidor).....	57
	BIBLIOGRAFÍA.....	58
	ANEXOS.....	60
	MANUAL DE USUARIO	60
	Login de Usuario	60
	Inicio de Sesión	60
	Pantalla de ingreso credenciales usuario	61
	Perfil de Usuario.....	63
	Editar datos Usuario	63
	Editar contraseña	64
	Eliminar cuenta de Usuario	65
	LogOut de Usuario	65

Cerrar Sesión	65
Administración de Módulos	66
Administración de Compañías	66
Opciones Administración Proveedores.....	67
Crear nuevo Proveedor	67
Editar Proveedor	69
Eliminar Proveedor.....	70
Reporte básico Proveedores	71
Administración de Visitadores	71
Opciones Administración Representantes	72
Crear nuevo Representante.....	73
Editar Representante.....	74
Eliminar Representante	75
Reporte básico Representantes	76
Administración de Zonas.....	77
Opciones Administración Zonas.....	78
Crear nueva Zona	79
Editar Zona	80
Eliminar Zona.....	81
Reporte básico Zonas	82

Administración de Productos	83
Opciones Administración Productos	83
Crear nuevo Producto	84
Editar Producto	86
Eliminar Producto.....	87
Reporte básico Productos	87
Administración de Inventario	88
Administración de Lotes.....	89
Opciones Administración Lotes	89
Crear nuevo Lote	90
Editar Lote	92
Eliminar Lote.....	93
Reporte básico de Lotes.....	94
Usuarios del sistema.....	95
Administración de Usuarios	95
Opciones Administración Usuarios	96
Crear nuevo Usuario.....	97
Editar Usuario.....	98
Actualizar contraseña Usuario.....	99
Eliminar Usuario	100

Reporte básico de Usuarios	101
Administración Muestras Médicas	102
Ingreso muestras médicas	102
Opciones Ingreso muestras médicas	102
Crear nuevo ingreso	103
Eliminar Ingreso	106
Reporte básico de Ingresos	107
Egreso muestras médicas	109
Opciones Egreso muestras médicas	109
Crear nuevo egreso	109
Eliminar Egreso	112
Reporte básico de Egresos	113
Reporte general stock	115
MANUAL DEL PROGRAMADOR	117
Tecnologías utilizadas	117
Requisitos del sistema	117
Hardware necesario	117
Software necesario	117
Instalación de software requerido	118
Instalación de Laragon	119

Instalación de Node.js	122
Instalación Visual Studio Code	125
Instalación Git	128
Instalación del proyecto	136
Clonación del repositorio	137
Abrir el proyecto con VS Code	138
Creación del archivo de configuración.....	140
Crea el archivo.env en la raíz del proyecto	140
Agregar variables de entorno.....	141
Instalación de dependencias	143
Instalación de composer	145
Inicializar servicios de Laragon.....	146
Creación de la base de datos.....	150
Creación del administrador del sistema.....	151
Código fuente Módulo Proveedores	155
Interfaz principal Proveedores.....	155
Código fuente	155
Interfaz crear proveedor	158
Código fuente	159
Interfaz editar proveedor	161

Código fuente	162
Interfaz eliminar proveedor	164
Código fuente	164
Interfaz buscar proveedor	165
Código fuente	165
Interfaz reporte de proveedores	166
Código fuente	167
Componente JavaScript	168
Código fuente	168
Nota interesante	170

ÍNDICE DE TABLAS

Tabla 1. Método Cuantitativo, Ventajas y Desventajas.	21
Tabla 2. Método Cualitativo, Ventajas y Desventajas.	22
Tabla 3. Desarrollo de Funcionalidades CRUD.	45
Tabla 4. Resultado de Pruebas de Usabilidad.	28
Tabla 5. Opciones Administración Proveedores.	67
Tabla 6. Opciones Administración Representantes.	73
Tabla 7. Opciones Administración Zonas.	79
Tabla 8. Opciones Administración Productos.	84
Tabla 9. Opciones Administración Lotes.	90
Tabla 10. Opciones Administración Usuarios.	97
Tabla 11. Opciones Ingreso muestras médicas.	103
Tabla 12. Opciones Egreso muestras médicas.	109

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Vista principal del Sistema Web.	27
Ilustración 2. Inicio de Sesión con Credenciales.....	28
Ilustración 3. Menú de Administración de Módulos.....	29
Ilustración 4. Formulario, Creación de Proveedor.	30
Ilustración 5. Lista de Proveedor Creado.	30
Ilustración 6. Formulario Creación de un Visitador.....	31
Ilustración 7. Lista de los Visitadores.	31
Ilustración 8. Formulario de Creación de Zonas.	32
Ilustración 9. Listado de las Zonas.....	32
Ilustración 10. Formulario para crear Productos.....	33
Ilustración 11. Listado de Productos.	33
Ilustración 12. Menú Administración Inventario.	34
Ilustración 13. Vista de Ingreso de Medicamentos.	35
Ilustración 14. Formulario para Ingresar Medicamentos.	35
Ilustración 15. Registro de Medicamentos ingresados.....	36
Ilustración 16. Listado de los Productos ingresados.	36
Ilustración 17. Vista de Egresos de Medicamentos.....	37
Ilustración 18. Selección de Medicamentos para Egresos.	37
Ilustración 19. Registro de Egresos.....	38
Ilustración 20. Vista Stock General de los Medicamentos.....	38
Ilustración 21. MVC del sistema web presentado.....	39
Ilustración 22. Diagrama E/R Base de Datos Pata de Gallo.	42
Ilustración 23. Prototipo (Mockup) del Sistema Web.....	43

Ilustración 24. Tablas a crear.	45
Ilustración 25. Código de migración para la tabla “users”	46
Ilustración 26. Código de migración para la tabla “companies”	47
Ilustración 27. Código de migración para la tabla “products”	47
Ilustración 28. Código de migración para la tabla “visitors y zones”	48
Ilustración 29. Código de migración para la tabla “batches”	48
Ilustración 30. Código de migración para la tabla “incomes”	49
Ilustración 31. Código de migración para la tabla “inventories”	49
Ilustración 32. Código de migración para la tabla “expenses”	50
Ilustración 33. Código del controlador de company.	51
Ilustración 34. Código del controlador de expense.	51
Ilustración 35. Código del controlador de income.	52
Ilustración 36. Código del controlador de inventory.....	52
Ilustración 37. Código del controlador de product.....	53
Ilustración 38. Código del controlador de user.	53
Ilustración 39. Código del controlador de visitor.....	54
Ilustración 40. Código del controlador de zone.	54
Ilustración 41. Vista Principal del Sistema de Control y Gestión de Muestras Médicas.	60
Ilustración 42. Pantalla Ingreso de Credenciales.	61
Ilustración 43. Vista Administración Módulo.....	62
Ilustración 44. Vista Administración de Inventario	62
Ilustración 45. Vista Administración de Usuarios	63
Ilustración 46. Módulo de edición de perfil.	63

Ilustración 47. Vista del Formulario de edición de Perfil.	64
Ilustración 48. Formulario Actualización de Contraseña.	64
Ilustración 49. Eliminación de Cuenta de Usuario.	65
Ilustración 50. Cierre de Sesión Usuario.	65
Ilustración 51. Vista Principal Administración Proveedores.	66
Ilustración 52. Formulario Creación Proveedor.	68
Ilustración 53. Detalle de Proveedores registrados.	68
Ilustración 54. Formulario edición Proveedor.	69
Ilustración 55. Eliminación de Proveedores.	70
Ilustración 56. Reporte Listado Proveedores.	71
Ilustración 57. Vista Principal Administración Visitadores.	72
Ilustración 58. Formulario Registro Representante.	74
Ilustración 59. Detalle de Representantes Registrados.	74
Ilustración 60. Formulario edición Representante.	75
Ilustración 61. Eliminación de Representantes.	76
Ilustración 62. Reporte Listado Representantes.	77
Ilustración 63. Vista Principal Administración Zonas.	78
Ilustración 64. Formulario Registro Zona.	79
Ilustración 65. Detalle de Zonas Registradas.	80
Ilustración 66. Formulario edición Zona.	81
Ilustración 67. Eliminación Zona	82
Ilustración 68. Reporte Listado Zonas.	82
Ilustración 69. Vista Principal Administración Productos.	83
Ilustración 70. Formulario Registro Producto.	85

Ilustración 71. Detalle de Productos Registrados.....	85
Ilustración 72. Formulario edición Producto.....	86
Ilustración 73. Eliminación Producto.....	87
Ilustración 74. Reporte Listado Zonas.	88
Ilustración 75. Vista Principal Administración Lotes.	89
Ilustración 76. Formulario Creación Lote.	91
Ilustración 77. Detalle de Lotes Registrados.....	92
Ilustración 78. Formulario edición Lote.....	93
Ilustración 79. Eliminación de Lote.	94
Ilustración 80. Reporte Listado Lotes.	95
Ilustración 81. Vista Principal Administración Usuarios.....	96
Ilustración 82. Formulario Creación Usuario.....	98
Ilustración 83. Detalle de Usuarios Registrados.	98
Ilustración 84. Formulario edición Usuario.	99
Ilustración 85. Actualizar contraseña Usuario.	100
Ilustración 86. Eliminación de Usuario.....	101
Ilustración 87. Reporte Listado Usuarios.....	102
Ilustración 88. Interfaz Ingresos.....	104
Ilustración 89. Datos fundamentales del Ingreso.	104
Ilustración 90. Visualización detalles del ingreso.....	105
Ilustración 91. Detalle de Ingresos registrados.	106
Ilustración 92. Eliminación de Ingresos.....	107
Ilustración 93. PDF Ingreso.	108
Ilustración 94. Vista Previa Ingreso.	108

Ilustración 95. Interfaz Egresos.....	110
Ilustración 96. Datos principales Egreso.....	110
Ilustración 97. Buscador de stock.	111
Ilustración 98. Visualización detalles del ingreso.....	111
Ilustración 99. Detalle de Egresos registrados.	112
Ilustración 100. Eliminación de Egresos.....	113
Ilustración 101. PDF Egreso.	114
Ilustración 102. Vista Previa Egreso.....	114
Ilustración 103. DASHBOARDS Inventario.	115
Ilustración 104. Stock Disponible.	115
Ilustración 105. Sistema Operativo.....	118
Ilustración 106. Especificaciones Hardware.	118
Ilustración 107. Directorio Descargas.....	119
Ilustración 108. Ejecución como administrador.....	119
Ilustración 109. Selección Idioma.....	120
Ilustración 110. Selección directorio instalación.	120
Ilustración 111. Opciones por defecto.....	120
Ilustración 112. Inicio instalación Laragon.....	121
Ilustración 113. Proceso instalación Laragon.	121
Ilustración 114. Finalización instalación.....	121
Ilustración 115. Ejecutar instalador node.....	122
Ilustración 116. Aceptar Términos.....	122
Ilustración 117. Directorio de instalación.	123
Ilustración 118. Opciones de instalación.....	123

Ilustración 119. Opciones de instalación.....	123
Ilustración 120. Inicio de instalación.	124
Ilustración 121. Progreso de instalación.	124
Ilustración 122. Instalación finalizada.	124
Ilustración 123. Directorio descargas.....	125
Ilustración 124. Aceptación de términos.....	125
Ilustración 125. Directorio de instalación VSC.	126
Ilustración 126. Opciones predeterminadas.	126
Ilustración 127. Opciones predeterminadas.	126
Ilustración 128. Inicio instalación VSC.	127
Ilustración 129. Progreso instalación VSC.	127
Ilustración 130. Instalación finalizada.	128
Ilustración 131. Directorio descargas.....	128
Ilustración 132. Opciones de instalación.....	129
Ilustración 133. Directorio de instalación.	129
Ilustración 134. Opciones de instalación.....	130
Ilustración 135. Opciones de instalación.....	130
Ilustración 136. Selección de VSC como editor por defecto para Git.	131
Ilustración 137. Opciones de instalación.....	131
Ilustración 138. Opciones de instalación.....	132
Ilustración 139. Opciones de instalación.....	132
Ilustración 140. Opciones de instalación.....	133
Ilustración 141. Opciones de instalación.....	133
Ilustración 142. Símbolo del sistema como terminal por defecto.	134

Ilustración 143. Opciones de instalación.....	134
Ilustración 144. Opciones de instalación.....	135
Ilustración 145. Inicio instalación Git.	135
Ilustración 146. Progreso instalación Git.	136
Ilustración 147. Finalización instalación Git.....	136
Ilustración 148. Directorio del proyecto.	137
Ilustración 149. Directorio donde se clonará el repositorio.	137
Ilustración 150. Proceso de clonación del repositorio.....	138
Ilustración 151. Directorio con el proyecto clonado.	138
Ilustración 152. Abrir Proyecto.	139
Ilustración 153. Seleccionar ubicación del proyecto.....	139
Ilustración 154. Estructura de archivos del proyecto.	140
Ilustración 155. Crear archivo .env	141
Ilustración 156. Archivo .env	141
Ilustración 157. Instalación Node.js	143
Ilustración 158. Instalación terminada.	144
Ilustración 159. Corrección de vulnerabilidades en dependencias.	144
Ilustración 160. Instalación de composer.	145
Ilustración 161. Instalación terminada composer.....	145
Ilustración 162. Ejecutar Servidor.....	146
Ilustración 163. Búsqueda de Laragon.	147
Ilustración 164. Activar servicios.....	147
Ilustración 165. Verificación de servicios.....	148
Ilustración 166. Dirección 127.0.0.1:8000.....	149

Ilustración 167. Dirección muestrasmédicas-app.test.	149
Ilustración 168. Apagar servidor.	150
Ilustración 169. Creación base de datos.	150
Ilustración 170. Base de datos creada.	151
Ilustración 171. Creación administrador del sistema.	152
Ilustración 172. Opción base de datos.	152
Ilustración 173. Administrador BD.	152
Ilustración 174. Ingreso al sistema como administrador.	153
Ilustración 175. Login.	154
Ilustración 176. Interfaz principal del sistema.	154
Ilustración 177. Interfaz principal Proveedores.	155
Ilustración 178. Views Companies.	155
Ilustración 179. Directorio de componentes Company.	158
Ilustración 180. Interfaz crear proveedor.	159
Ilustración 181. Interfaz editar proveedor.	161
Ilustración 182. Interfaz eliminar proveedor.	164
Ilustración 183. Interfaz búsqueda proveedor.	165
Ilustración 184. Reporte proveedores.	167

INTRODUCCIÓN

La gestión eficiente de los recursos dentro del sector farmacéutico constituye un factor clave para el adecuado funcionamiento de las organizaciones, especialmente cuando se trata de elementos estratégicos como las muestras médicas gratuitas. Las muestras, aunque no representen un valor comercial directo, cumplen un rol fundamental en la promoción y posicionamiento de los productos en el mercado, siendo utilizadas por los representantes de ventas para fortalecer la relación con los puntos de distribución (farmacias) y profesionales de la salud.

La empresa HOSPIMEDIKKA, se dedica a la comercialización de medicamentos, donde enfrenta una problemática relacionada con la administración de sus muestras médicas. Actualmente, el proceso de registro, entrega y almacenamiento de dichas muestras se realiza de manera manual, a través de un sistema de Kardex que opera de forma independiente al sistema digital encargado de gestionar el resto del inventario. Esta separación ha generado un modelo de gestión paralelo, ineficiente y susceptible a errores, lo cual dificulta la realización de auditorías internas y compromete la transparencia en la distribución de estos recursos.

Frente a esta situación, se establece el desarrollo de un sistema automatizado (aplicación web) que permita controlar y administrar eficientemente las muestras médicas, integrando esta funcionalidad al sistema principal de la empresa. Esta solución no solo busca optimizar los procesos internos y reducir los márgenes de error, sino también fortalecer la labor de los representantes de ventas y mejorar la toma de decisiones mediante el acceso a información confiable.

ANTECEDENTES

Hospimedikka, es una empresa ecuatoriana con más de 25 años de trayectoria en el mercado, dedicada a la importación y comercialización de productos farmacéuticos, como parte de su estrategia comercial, la empresa dota a sus Representantes de Ventas de muestras médicas con el fin de que puedan presentarlas directamente a consultorios y médicos particulares, promoviendo así la prescripción de los productos que comercializa.

Actualmente, el proceso de control de ingreso y salida de estas muestras médicas se realiza de forma manual, a través de Kardex físicos llenados por los asistentes de bodega, para el proceso de ingreso de muestras se recibe un documento enviado por el encargado del área de Importaciones que detalla el proveedor, la descripción del producto y la cantidad a ingresar.

En la práctica este proceso generalmente no presenta inconvenientes, ya que la información proviene directamente del proveedor y se encuentra bien estructurada, sin embargo, las dificultades surgen en el proceso de salida de las muestras médicas, en donde el área de Análisis de Mercado elabora un listado con la distribución de las muestras por cada Representante de Venta, de acuerdo con los productos que estos deben proporcionar en sus visitas.

Este listado es entregado al jefe de bodega, quien lo revisa y escoge a uno de sus asistentes los cuales separan las cantidades detalladas en el listado y proceden a registrar manualmente en los Kardex físicos la salida de las muestras para cada representante.

Esta metodología ha generado diversos problemas operativos, principalmente relacionados con errores humanos en el llenado manual de los Kardex, esto ha llevado a presentar registros incorrectos, información ilegible, omisiones, y retrasos en la elaboración de los documentos, como consecuencia, se presentan demoras en la entrega de las muestras a los

Representantes de Ventas, lo que afecta directamente la efectividad de sus visitas, ya que no siempre cuentan con el material promocional a tiempo para cumplir con sus objetivos comerciales, como consecuencia las ventas particulares y privadas de los productos que se comercializan han disminuido provocando una mínima rotación de los mismos en el mercado ecuatoriano.

Ante esta problemática, se plantea el desarrollo de un sistema web que permita digitalizar y automatizar el control y gestión de las muestras médicas, con el fin de reducir errores, optimizar tiempos y garantizar una trazabilidad clara y eficiente en los procesos de ingreso y salida de productos desde bodega.

PROBLEMA

HOSPIMEDIKKA, es una empresa dedicada a la comercialización de medicamentos, realiza además la entrega de muestras médicas gratuitas a sus representantes de ventas para su distribución en distintos puntos de venta (PDV).

Actualmente, el proceso de registro, entrega y almacenamiento de estas muestras se gestiona de forma manual mediante el uso de un sistema de *Kardex*, independiente del sistema digital que administra el resto de los productos médicos de la empresa. Este sistema principal no contempla las muestras médicas, ya que estas no generan un valor comercial directo, lo que ha llevado a una gestión paralela, ineficiente y propensa a errores. La ausencia de un control automatizado limita la trazabilidad, dificulta la auditoría de los movimientos de stock de muestras y pone en riesgo la transparencia del proceso.

Esta situación evidencia la necesidad de desarrollar un sistema automatizado que permita administrar y controlar eficientemente las muestras médicas, integrando esta información a los procesos internos de la empresa y optimizando la labor de los representantes de ventas.

JUSTIFICACIÓN

Justificación Social

El desarrollo del sistema web para el control y administración de muestras médicas tiene un impacto social significativo, al optimizar la trazabilidad, control y disponibilidad de muestras en la empresa Hospimedikka, contribuye a la gestión más eficiente y segura de los insumos fundamentales para la visita médica a clínicas o consultorios, permitiendo una respuesta más rápida y organizada ante necesidades médicas.

Además, al reducir errores humanos y mejorar el acceso a la información en tiempo real, el sistema beneficia directamente al personal de bodega, encargado del control y almacenamiento de las muestras médicas. El desarrollo de este sistema web promueve la digitalización de procesos esenciales en el área logística, alineándose con las necesidades actuales. Asegura que los profesionales de la salud reciban el material promocional correcto y a tiempo, fortaleciendo la relación de confianza con la farmacéutica y garantizando que la información sobre nuevos medicamentos fluya eficientemente, lo cual es de interés para la salud pública.

Justificación Técnica

Desde el punto de vista técnico, esta tesis se justifica por la necesidad de modernizar y optimizar los procesos internos relacionados con la gestión de muestras médicas en la empresa HOSPI MEDIKKA. El proceso actual, basado en un Kardex manual y desconectado del sistema digital principal, representa una solución obsoleta y vulnerable, que limita la eficiencia operativa, la trazabilidad de los recursos y la capacidad de respuesta ante auditorías internas o requerimientos regulatorios.

La propuesta de un sistema web responde a esta necesidad mediante el desarrollo de una herramienta tecnológica que integre la gestión de muestras al ecosistema digital de la empresa. Esto permitirá centralizar la información, reducir errores humanos, agilizar los procesos y generar reportes en tiempo real que faciliten la toma de decisiones.

Este proyecto representa una oportunidad para aplicar metodologías modernas de análisis, diseño y desarrollo de sistemas, así como tecnologías actuales que aseguren escalabilidad, seguridad y facilidad de uso. Además, sienta las bases para futuras integraciones con otras áreas de la empresa, alineando la gestión de muestras con una visión de transformación digital sostenible.

OBJETIVOS

Objetivo General

Implementar un sistema web para el control y gestión de las muestras médicas en la empresa HOSPIMEDIKKA, con el fin de optimizar la trazabilidad, eficiencia operativa y transparencia en los procesos de registro, control y entrega.

Objetivos Específicos

- Analizar detalladamente el proceso actual de entrega, registro y control de muestras médicas.
- Determinar los requerimientos funcionales y técnicos necesarios para el desarrollo del sistema automatizado.
- Diseñar una solución tecnológica que permita el registro eficiente, seguimiento detallado y control centralizado de las muestras médicas.
- Desarrollar el sistema web utilizando la arquitectura MVC con Laravel y MySQL para automatizar los procesos de registro y control.
- Validar el funcionamiento del sistema web mediante pruebas técnicas y de usuario, asegurando que cumple con los requerimientos definidos.

ALCANCE

La presente tesis abarca el diseño e implementación de un sistema automatizado de gestión de muestras médicas para la empresa HOSPIMEDIKKA, con el objetivo de optimizar los procesos de registro, control, almacenamiento y entrega de dichas muestras. El alcance del proyecto comprende la mejora en el proceso de gestión de muestras médicas, partiendo de un análisis del sistema actual que evidencia limitaciones como la falta de trazabilidad y el control manual del inventario. Se procedió con el levantamiento de requerimientos funcionales y no funcionales en colaboración con las áreas involucradas, para diseñar y desarrollar un sistema web automatizado que permita registrar el ingreso de muestras por producto, proveedor y cantidad; controlar su entrega con trazabilidad; administrar el inventario en tiempo real; y generar reportes para auditoría.

El sistema se valida mediante pruebas funcionales y de usuario, y fue desarrollado con tecnologías que garanticen su escalabilidad y seguridad. Aunque no se prevé para aplicaciones móviles, el sistema establece una base sólida para futuras mejoras, y está orientado a optimizar la eficiencia operativa y el control interno de la empresa.

MARCOS REFERENCIALES

El presente marco referencial tiene como propósito sustentar la necesidad, viabilidad y beneficios de desarrollar un sistema web para el control y gestión de muestras médicas dentro de la empresa Hospimedikka Cía. Ltda. Enmarcado dentro de las buenas prácticas de distribución, almacenamiento y de gestión de medicamentos con el fin de contribuir a la eficiencia operativa y al cumplimiento de los objetivos comerciales de la empresa.

MARCOS LEGALES

La regulación del manejo de productos farmacéuticos en Ecuador está formada por la Agencia Nacional de Regulación, Control y Vigilancia Sanitaria (ARCSA), la cual establece lineamientos para el control de medicamentos y su distribución, incluyendo las muestras médicas. También se deben considerar las disposiciones del Código Orgánico de la Salud (COS) y las normativas relacionadas con la promoción ética de productos farmacéuticos (ARCSA, 2017).

MARCO POLÍTICO-SOCIAL

En Ecuador, el acceso a medicamentos y la promoción responsable de productos farmacéuticos son temas prioritarios en la política pública de salud. Las empresas deben garantizar prácticas éticas en la distribución de muestras médicas, evitando el uso indiscriminado y asegurando que lleguen a quienes realmente las necesitan. La automatización de este proceso contribuye a la rendición de cuentas, mejora la transparencia y responde a la creciente demanda de digitalización en el sector salud.

La relación que existe entre el sistema a implementar y la normativa de ARCSA responde precisamente en digitalizar y controlar rigurosamente el proceso de gestión de muestras médicas, garantizando que estas lleguen a los destinatarios correctos (profesionales de la salud autorizados) en la fecha estipulada, en el momento oportuno y con un respaldo de trazabilidad. Esto no solo agiliza el proceso del flujo de trabajo de la empresa, también promueve la transparencia, la rendición de cuentas y la ética en la promoción de medicamentos, tal y como lo exige la normativa nacional vigente (ARCSA, 2017).

El proceso que anteriormente era manual y vulnerable a errores al ser automatizado el sistema fortalecerá la capacidad de auditoría, reduciendo así riesgos asociados al uso indiscriminado o inadecuado de las muestras. Todo esto contribuye a la digitalización responsable del sector salud, un objetivo clave para el desarrollo tecnológico del país.

CAPÍTULO I

MARCO TEÓRICO

Esta sección tiene como objetivo fundamentar técnicamente los principales conceptos, herramientas y tecnologías que sustentan este proyecto, así como las bases metodológicas y arquitectónicas sobre las que se construirá la solución informática. Así mismo, se analizan las buenas prácticas empleadas en el desarrollo de software, gestión de inventarios y la seguridad informática, asegurando un sistema eficaz y confiable con los requerimientos de la organización.

1.1 Trazabilidad y Gestión de Inventario

1.1.1 Trazabilidad

Se entiende como trazabilidad la capacidad de seguir cada muestra médica desde su ingreso hasta su entrega final. Laravel, mediante migraciones y relaciones Eloquent, permite registrar cada evento asociado a las muestras sea (fecha, responsable, destino) con precisión y eficiencia. La base de datos relacional MySQL establece relaciones entre tablas clave como muestras, usuarios, representantes de ventas y reportes de entrega, garantizando la integridad permitiendo así generar reportes personalizados y exportables para auditorías.

1.1.2 Importancia de la Trazabilidad

La importancia de la trazabilidad en la distribución de muestras médicas representa un componente esencial ya que garantiza seguridad, cumplimiento normativo y la transparencia en su distribución. El rastreo es crucial desde el ingreso a bodega hasta la entrega final a los representantes de ventas, esto es esencial para evitar pérdidas, errores logísticos o uso indebido de estos productos, que, aunque no se comercialicen directamente, cumplen una función estratégica en la promoción médica. La implementación de un sistema de trazabilidad permitirá

identificar, registrar y verificar cada movimiento de las muestras, fortaleciendo los controles internos y reduciendo significativamente los riesgos asociados a la falsificación o desvío de productos farmacéuticos.

1.1.3 Trazabilidad en la Cadena de Suministros Farmacéutica

Desde una vista regulatoria y cumpliendo con normas establecidas por ARCSA en Ecuador, la trazabilidad desempeña un rol importante en la cadena de suministro farmacéutica, identifica y sigue el trayecto de cualquier producto desde su origen hasta su destino final. Con las muestras médicas, la trazabilidad asegura la localización de cada muestra así mismo será registrada y verificada en cada proceso logístico garantizando que las muestras lleguen correctamente a los profesionales de la salud, reduciendo errores, pérdidas o entregas indebidas.

La trazabilidad, fortalece la seguridad al prevenir el uso inadecuado de productos, y mejora la eficiencia de los procesos internos al facilitar auditorías y reportes en tiempo real.

1.2 Sistemas de Inventarios

Se entiende como sistemas de inventarios a la parte operacional donde existen normas, métodos y procedimientos que organizan y controlan el stock de productos o materiales dentro de una empresa (entradas, salidas, fechas, lotes, entre otros), pueden ser manuales o automatizados. Su principal función es asegurar la disponibilidad de insumos para las operaciones, evitando excesos o faltantes que afecten el funcionamiento del negocio.

1.3 Tipos de Sistemas de Inventarios

1.3.1 Sistema de Inventario Perpetuo

Este sistema realiza un seguimiento continuo del inventario disponible, registra cada movimiento de entrada y salida en tiempo real. Se apoya en tecnologías como computadoras y lectores de códigos de barras, permitiendo así una actualización inmediata del inventario cada

vez que se realiza una transacción. El sistema está diseñado para generar pedidos automáticamente cuando la posición de inventario (suma del stock disponible y las recepciones programadas menos los pedidos pendientes) alcanza un punto mínimo definido, conocido como punto de reorden. La cantidad a solicitar (Q) es fija, pero los tiempos entre pedidos pueden variar según el consumo o demanda.

Esto facilita la toma de decisiones en tiempo real, mejora la trazabilidad del stock y es ideal para empresas que requieren un alto control de sus productos, como es el caso de las muestras médicas en HOSPIMEDIKKA, donde la disponibilidad inmediata y el control detallado son fundamentales para la eficiencia del proceso (Safetyculture, 2024).

1.3.2 Sistema de Inventario Periódico

Al contrario del sistema perpetuo, este sistema no mantiene registros en tiempo real. En su lugar, el conteo se lo realiza al final de cada período establecido ya sea (mensual, trimestral, etc.), y a partir de estos datos se determina el inventario final y el costo de las mercancías vendidas. Este modelo, también llamado sistema físico, suele aplicarse en negocios con productos de bajo costo unitario o menor rotación.

Un ejemplo práctico de este método es el sistema de revisión periódica (P), en el que el inventario se inspecciona a intervalos fijos, y se genera un nuevo pedido con base en las necesidades detectadas. Aunque simplifica la logística, este enfoque puede conllevar mayor riesgo de faltantes si la demanda varía bruscamente entre revisiones.

En contextos donde la trazabilidad y precisión son prioritarias, como en el manejo de insumos médicos, el sistema periódico resulta menos eficaz frente a las ventajas que ofrece un sistema automatizado y perpetuo (Loja, 2015).

1.4 Kardex

Es un documento tradicional que lleva el control del inventario de un almacén o empresa, así mismo es de carácter administrativo donde se anota de manera manual los movimientos de entrada y salida de toda la mercancía.

Este registro lleva información necesaria como la cantidad, su medida, su precio y valor total. En base a estos datos se los organiza según su tipo, categoría, similitudes facilitando su localización y gestión.

1.5 Arquitectura de Software para Aplicación de Gestión

1.5.1 El Patrón Modelo-Vista-Controlador (MVC)

El sistema propuesto sigue una arquitectura web cliente-servidor, donde el backend desarrollado en Laravel maneja la lógica del negocio y la persistencia de datos en una base de datos MySQL. Esta arquitectura permite una gestión segura y centralizada de las operaciones de entrada y salida de muestras médicas en la empresa HOSPIMEDIKKA.

Laravel sigue el patrón MVC (Modelo-Vista-Controlador), que separa claramente la lógica de negocio, la presentación y el acceso a datos, facilitando el mantenimiento y escalabilidad del sistema (Laravel, s.f.). Laravel Breeze, se utiliza para la autenticación de usuarios el cuál proporciona un sistema de login, registro, verificación simple y seguro, con integración de rutas protegidas y middlewares (Laravel, s.f.).

1.5.2 Justificación del uso de Laravel

El uso del framework Laravel para el desarrollo del sistema web se fundamenta en su robusta implementación del patrón de arquitectura Modelo-Vista-Controlador (MVC), el cual permite una separación clara entre la lógica del negocio, la interfaz de usuario y el acceso a los datos. Esta estructura fomenta el orden, la escalabilidad y el mantenimiento del código en el

desarrollo de sistemas informáticos que requieren trazabilidad, seguridad y control como en el caso de la gestión de muestras médicas.

Laravel provee herramientas integradas para la autenticación, validación, migración de base de datos, enrutamiento y protección contra vulnerabilidades comunes. Así mismo garantiza que el sistema desarrollado sea estructurado, seguro y conforme con las buenas prácticas del desarrollo de software moderno. Por lo tanto, responde a las necesidades técnicas del proyecto en base a los objetivos requeridos (Altube, 2021).

1.6 Stack Tecnológico para el Sistema de Gestión de Muestras Médicas

1.6.1 Visual Studio Code

Editor de código gratuito, es de código abierto y multiplataforma. Permite editar texto de los archivos de código para programación, posee distintos lenguajes de programación facilitando a los desarrolladores manejar diversos lenguajes permitiendo escribir y editar el código paso a paso. VSC, ofrece ayudas a los programadores como resaltar la sintaxis o a su vez autocompletar el código (Zuñiga, 2024).

1.6.2 Laragon

Es una herramienta orientada a la creación de múltiples entornos de desarrollo, permitiendo trabajar de forma eficiente con distintos lenguajes de programación del lado del servidor como PHP, NodeJS, Ruby y Python (Zúñiga, 2024).

1.6.3 Backend PHP y el Framework Laravel

1.6.3.1 Framework

Un framework es un entorno que facilita el desarrollo de aplicaciones web o de escritorio, proporcionando herramientas como bibliotecas, plantillas y gestión de recursos que simplifican el trabajo del programador (Hostalia, Hostalia, s.f.).

1.6.3.2 Laravel

El sistema propuesto sigue una arquitectura web cliente-servidor, donde el backend desarrollado en Laravel maneja la lógica del negocio y la persistencia de datos en una base de datos MySQL. Esta arquitectura permite una gestión segura y centralizada de las operaciones de entrada y salida de muestras médicas en la empresa HOSPIMEDIKKA.

Laravel sigue el patrón MVC (Modelo-Vista-Controlador), que separa claramente la lógica de negocio, la presentación y el acceso a datos, facilitando el mantenimiento y escalabilidad del sistema (Laravel, s.f.). Laravel Breeze, se utiliza para la autenticación de usuarios el cuál proporciona un sistema de Login, registro, verificación simple y seguro, con integración de rutas protegidas y middlewares (Laravel, s.f.).

1.6.4 Frontend

1.6.4.1 JavaScript y jQuery

Hablamos de JavaScript como un lenguaje de programación y jQuery como una biblioteca rápida en distintas funciones, permite al programador crear una interfaz dinámica y responsiva para el usuario facilitando así el registro rápido de muestras o consultoría de reportes en sitios web o aplicaciones web.

1.6.4.2 TailwindCSS

Es un framework utilitario de estilos, incorporando funcionalidades que ofrecer una experiencia de desarrollo más eficiente y cómoda al momento de crear interfaces atractivas de manera ágil (Tailwindcss, s.f.).

1.6.5 Base de Datos MySQL

Es un sistema de base de datos relacional totalmente integrada, es conocido por su estabilidad, rendimiento y facilidad de integración con aplicaciones desarrolladas en PHP. Al

ser de código abierto junto con el soporte de una gran comunidad y actualizaciones constantes, lo convierten en una opción fiable, segura ante fallos y bloqueos para entornos de producción (Erickson, 2024).

Entre las muestras médicas, MySQL permite estructurar y relacionar eficientemente tablas que contienen información clave como productos, lotes, inventarios, usuarios, registros de ingresos y egresos asegurando la integridad y la consistencia de los datos. Al ser compatible Laravel, facilita la implementación de migraciones y consultas optimizadas, reduciendo el riesgo de errores y garantizando un acceso rápido a la información en tiempo real.

CAPÍTULO II

METODOLOGÍA

Este capítulo detalla el marco metodológico utilizado para el diseño y desarrollo de la aplicación web de gestión de muestras médicas para HOSPIMEDIKKA. Se describe el tipo de investigación, la metodología de gestión de proyectos seleccionada y las fases secuenciales seguidas, desde el análisis de requisitos hasta la validación del sistema. Para ello, se revisó el manejo actual mediante Kardex, se recopiló información del personal involucrado desarrollando así una herramienta digital que permita mejorar el control, seguimiento y organización de las muestras, de acuerdo con los objetivos del proyecto. Además, durante la implementación del sistema se aplicó el método Kanban como técnica de gestión de tareas, el cual permite organizar de forma visual las actividades, como el seguimiento al progreso del proyecto, priorizando los avances de manera ordenada. Esta parte del capítulo se eligió esta metodología porque permite el uso de tableros divididos en columnas, representando las siguientes actividades, “Terminados”, “En Desarrollo” y “Pendientes”, permitiendo identificar con claridad el estado de cada tarea asignada. Esta técnica ayuda al equipo de desarrollo visualizar fácilmente las actividades pendientes, la toma de decisiones rápidas, adaptación a los cambios asignando responsabilidades de forma eficiente evitando así cuellos de botella durante las etapas críticas del proyecto, mejorando así los puntos clave que logran una implementación ordenada y exitosa del sistema.

2.1 Tipo y Enfoque de Investigación

El proyecto compete a una investigación aplicada, la cual busca una solución a una problemática real detectada en la empresa HOSPIMEDIKKA, mediante el diseño e implementación de un sistema web para la gestión y control de muestras médicas. Como

resultado, optimizar procesos internos y mejorando la trazabilidad de la información, con resultados directamente utilizables por la empresa.

Dentro del enfoque de investigación, se utilizó un enfoque mixto, el cuál combina métodos cuantitativos y cualitativos. El componente cuantitativo permite recopilar y analizar datos objetivos del proceso actual, entre ellos tiempos de entrega, errores de registro y niveles de trazabilidad antes y después de la implementación del sistema.

El componente cualitativo se centra en entrevistas, observaciones directas y análisis de la experiencia de los usuarios, identificando necesidades específicas mejorando la usabilidad de la herramienta.

2.2 Método Cuantitativo

El siguiente proyecto se enfoca en un método cuantitativo, basándose en la recopilación y análisis de datos específicos del proceso actual de gestión de muestras médicas, tales como registros físicos (Kardex), tiempos de respuesta y errores frecuentes. A partir de esta información, se identificaron requerimientos funcionales y no funcionales que permitieron así el desarrollo del sistema web. Dentro del análisis se evaluaron los siguientes puntos:

- Errores de registros.
- Tiempo de entrega de las muestras médicas.
- Entrega de la información de muestras médicas por usuario.
- Niveles de trazabilidad antes y después de la implementación.

Método	Ventajas	Desventajas
Cuantitativo	Obtener datos objetivos y medibles.	No considera los aspectos subjetivos del usuario como la experiencia y la satisfacción.
	Comparación entre el sistema manual y aplicativo web.	Se requiere de registros claros para un análisis confiable.
	Generar resultados concretos apoyando así la toma de decisiones.	

Tabla 1. Método Cuantitativo, Ventajas y Desventajas.

Esta información servirá como base para evaluar la eficiencia del sistema web hablando en términos de optimización y reducción de errores.

2.3 Método Cualitativo

El enfoque cualitativo se aplica en investigaciones para estudiar información no estructurada ni cuantificable, buscando una comprensión detallada y profunda del fenómeno analizado. Este método posee diversas características que ayudan al desarrollo efectivo del proyecto realizado, por ejemplo:

- Entrevista al personal: representantes de ventas, asistentes de bodega y jefe de logística.
- Observación directa del flujo operativo del manejo de muestras en bodega.
- Análisis de documentación interna y reportes manuales.

Método	Ventajas	Desventajas
Cualitativo	Comprende la percepción, necesidad y experiencia del usuario.	Los resultados no se generan fácilmente.
	Detecta problemas que no pueden ser visibles con los datos numéricos.	La interpretación puede ser influenciada en base a la perspectiva del investigador.
	Obtención de ideas para la mejora de usabilidad del sistema.	

Tabla 2. Método Cualitativo, Ventajas y Desventajas.

Este método permite facilitar las dificultades recurrentes y las necesidades de los usuarios finales, ajustando así el diseño del sistema web y mejorando su usabilidad y la adaptabilidad en el entorno laboral.

2.4 Fases de Desarrollo del Proyecto

Para el desarrollo del proyecto se pasó por distintas fases que permitieron construir el trabajo de una manera ordenada, garantizando el cumplimiento de los objetivos planteados. A continuación, se detalla de forma general las distintas fases que se tomó en cuenta para este proceso.

2.5 Estructura

2.5.1 Fase 1: Análisis y Levantamiento de Requisitos

En esta etapa se identificaron todas las necesidades del área de bodega de la empresa, mediante reuniones y levantamiento de información con los usuarios clave. Se definen las funcionalidades principales, los actores del sistema y procesos que deben ser automatizados.

2.5.2 Fase 2: Diseño del Sistema

Obtenido los requisitos, se procede con el diseño de la estructura lógica del sistema. Se elaboraron diagramas de casos de uso, diagramas entidad-relación y una arquitectura base de la aplicación. Se definió la estructura de la base de datos y los roles de los usuarios.

2.5.3 Fase 3: Desarrollo e Implementación

En esta fase se implementó el sistema utilizando tecnologías como Laravel, PHP, JavaScript y MySQL. Los módulos fueron desarrollados, así mismo modelos, controladores y vistas dependiendo de los requerimientos. Así mismo, se integró un sistema de autenticación, políticas de acceso y se aplicó un diseño visual usando TailwindCSS.

2.5.4 Fase 4: Pruebas y Validación

Se realizaron pruebas manuales que permitieron validar el funcionamiento del sistema. Un representante del área de bodega se encargó de ingresar información y verificar que los procesos respondan de forma correcta. Las pruebas realizadas permitieron detectar errores, realizar ajustes y asegurar que la aplicación web cumpliera con lo solicitado.

2.5.5 Fase 5: Entrega y Capacitación

Verificado que el funcionamiento del sistema haya sido el correcto, se prepara su entrega para su uso final. Se realizó una capacitación básica para los usuarios que van a intervenir y adicional se dejó documentado el uso general del sistema.

2.6 Población y Muestra

2.6.1 Población

La población de estudio está compuesta por el personal del área de bodega y logística de la empresa HOSPIMEDIKKA.

2.6.2 Muestra

Para el levantamiento de requisitos y las pruebas de usabilidad, se trabajó con una muestra por conveniencia que incluye al jefe de bodega y a los asistentes responsables del manejo del Kardex.

2.7 PLANTEAMIENTO DE LAS VARIABLES

2.7.1 Variable Independiente (Implementación del Sistema Web de Gestión)

Esta variable es independiente y corresponde a la utilización de la aplicación web desarrollada específicamente para automatizar el proceso de gestión de muestras médicas. Esta implementación sustituye el sistema manual basado en registros físicos (Kardex) por una solución digital que centraliza, organiza y controla las operaciones de ingreso, entrega e inventario de muestras médicas.

2.7.1.1 Valores

La variable será evaluada en dos partes:

- Proceso manual: antes de la implementación del sistema web.
- Proceso automatizado: después de la implementación del sistema web

2.7.2 Variable Dependiente (Eficiencia del Proceso de gestión de Muestras Médicas)

Se refiere al grado alcanzado en el proceso de gestión de muestras médicas como los resultados de la automatización. Incluye aspectos clave como la rapidez en los registros, la reducción de errores y la trazabilidad completa del ciclo de vida de las muestras, desde su ingreso hasta su entrega final al representante de ventas.

2.8 Indicadores Medibles

2.8.1 Tiempo de Proceso

Tiempo promedio (en minutos/horas) requerido para registrar la salida de un lote de muestras para un representante.

2.8.2 Precisión del Registro

Porcentaje de reducción de errores o discrepancias entre el registro del sistema y el conteo físico del inventario de muestras.

2.8.3 Nivel de Trazabilidad

Tiempo promedio (en segundos) para localizar el historial completo de una muestra o lote específico a través del sistema.

2.9 Discusión del problema

La problemática inicial de HOSPI MEDIKKA se centraba en la gestión manual de las muestras médicas mediante Kardex físicos, lo cual generaba errores frecuentes, retrasos en la entrega de insumos y ausencia de trazabilidad. Estos factores afectaban la transparencia en la distribución y disminuían la eficacia de los representantes de ventas.

Los resultados obtenidos tras la implementación del sistema web evidencian una mejora significativa frente a esta situación. En las pruebas de usabilidad, los tiempos de registro y consulta se redujeron considerablemente (por ejemplo, consultas en 25 segundos y generación de reportes en 35 segundos), lo que contrasta con los procesos manuales que eran más lentos y propensos a errores. Asimismo, se logró una reducción casi total de inconsistencias en los registros, confirmando que la digitalización mejora la precisión y confiabilidad de la información.

En términos comparativos, el sistema automatizado superó ampliamente al método manual, ofreciendo trazabilidad en tiempo real, reportes automáticos y facilidad de auditoría. Esto implica que la empresa ahora cuenta con una herramienta tecnológica que fortalece la toma de decisiones, incrementa la eficiencia operativa y garantiza mayor control sobre recursos estratégicos.

Sin embargo, también se identifican limitaciones. El sistema fue diseñado para entorno web local, por lo que su acceso se restringe al espacio de trabajo interno, limitando la movilidad de los representantes de ventas. Además, aunque las pruebas funcionales y de usabilidad fueron exitosas, aún se requiere capacitación constante al personal y una futura implementación en servidores externos o plataformas móviles para asegurar escalabilidad y disponibilidad.

En síntesis, la discusión confirma que la solución propuesta responde de manera efectiva a la problemática planteada, generando un impacto positivo en la trazabilidad, transparencia y eficiencia, aunque persisten retos relacionados con la expansión tecnológica y la adopción por parte de los usuarios.

2.10 PRUEBAS Y RESULTADOS

El presente capítulo expone el proceso de validación del sistema web mediante la aplicación de pruebas funcionales y pruebas de usabilidad, con el fin de garantizar su correcto desempeño y la satisfacción de los usuarios finales. Las pruebas funcionales permitieron verificar que cada módulo cumpla con los requerimientos establecidos, asegurando el registro, control y gestión adecuada de las muestras médicas.

Posteriormente, las pruebas de usabilidad se enfocaron en evaluar la interacción de los usuarios con la aplicación, analizando aspectos como la facilidad de navegación, la claridad de las interfaces y la eficiencia en la ejecución de tareas. Los resultados obtenidos en ambas etapas

confirman la operatividad del sistema, evidenciando mejoras significativas en la trazabilidad, la reducción de errores y la optimización de los procesos internos de la empresa.

En la ilustración 21 se visualiza la pantalla principal donde nos permite ver el logo de la empresa y así mismo la opción para ingresar como administrador y usuario.

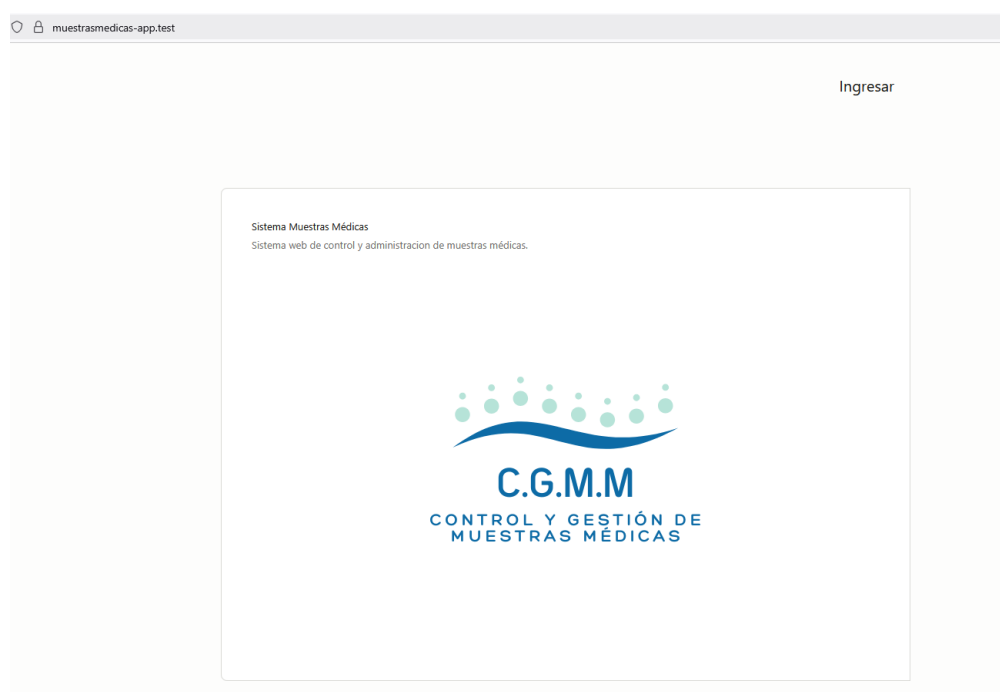


Ilustración 1. Vista principal del Sistema Web.

2.10.1 Pruebas de Usabilidad Ingreso, Consulta y Generación de Reporte.

Las pruebas de usabilidad se realizaron con el propósito de evaluar la interacción de los usuarios con el sistema web, verificando que las interfaces sean intuitivas, los tiempos de respuesta adecuados y las funciones ejecutadas sin errores.

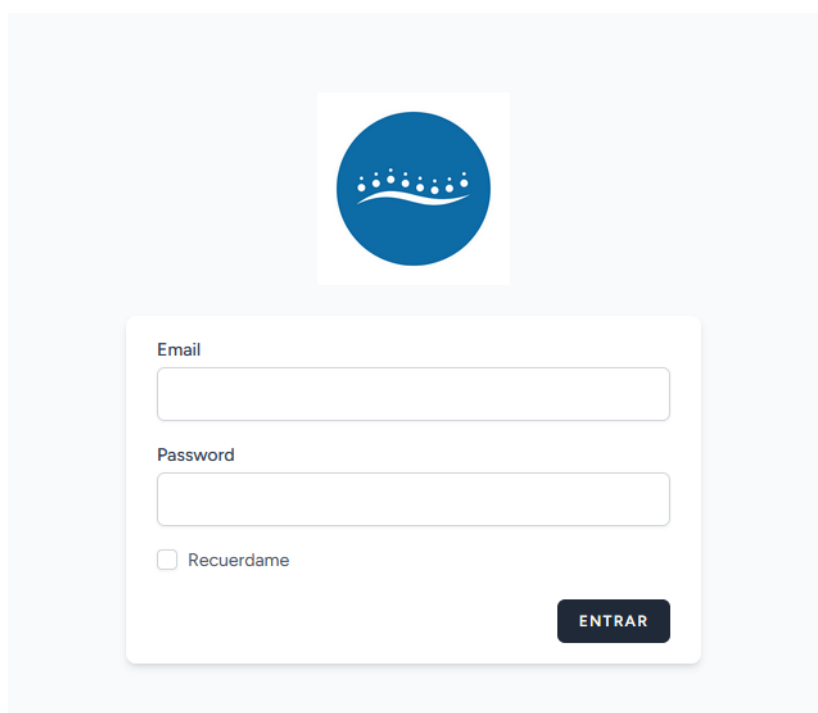
En la tabla 4 se presentan los resultados obtenidos de las pruebas de usabilidad aplicadas al sistema web. Cada caso de prueba describe la tarea realizada, el tiempo de ejecución y los resultados obtenidos, lo que permite evidenciar la eficiencia del sistema y su facilidad de uso por parte de los usuarios finales.

Caso de Prueba	Descripción	Resultado
Ingreso de muestras	Registrar un nuevo lote o existente en el inventario	Tiempo: 50 seg – 1 min, sin errores. Confirmación visual exitosa.
Consulta de inventario	Buscar un producto por nombre y verificar su stock y ubicación	Tiempo: 25 segundos. Resultados precisos y en orden.
Generación de reporte PDF	Filtrar por rango de fechas y exportar el informe	Tiempo: 35 segundos. Documento generado correctamente.

Tabla 3. Resultado de Pruebas de Usabilidad.

2.10.2 Pruebas de Usabilidad del Sistema Web Local

“En la ilustración 22 se muestra el proceso de inicio de sesión con las credenciales de los usuarios (administrador y bodega), obteniéndose un tiempo de respuesta de 5 segundos, tras lo cual el acceso se considera exitoso y el proceso queda completado.”



The image shows a login form on a light blue background. At the top center is a circular logo with a blue background and white dots forming a wave-like pattern. Below the logo is a white rectangular box containing the login fields. Inside this box, there is an 'Email' label above a text input field, a 'Password' label above another text input field, and a checkbox labeled 'Recuerdame'. At the bottom right of the white box is a dark blue button with the word 'ENTRAR' in white capital letters.

Ilustración 2. Inicio de Sesión con Credenciales.

2.11 Acceso Usuario Administrador

En la ilustración 23 se presenta el menú de administración de módulos, en el cual se encuentran distintos apartados que permiten al administrador y al usuario de bodega gestionar

la creación de compañías, visitantes, zonas y productos. En este apartado se evaluará el tiempo de respuesta y el flujo del proceso correspondiente.

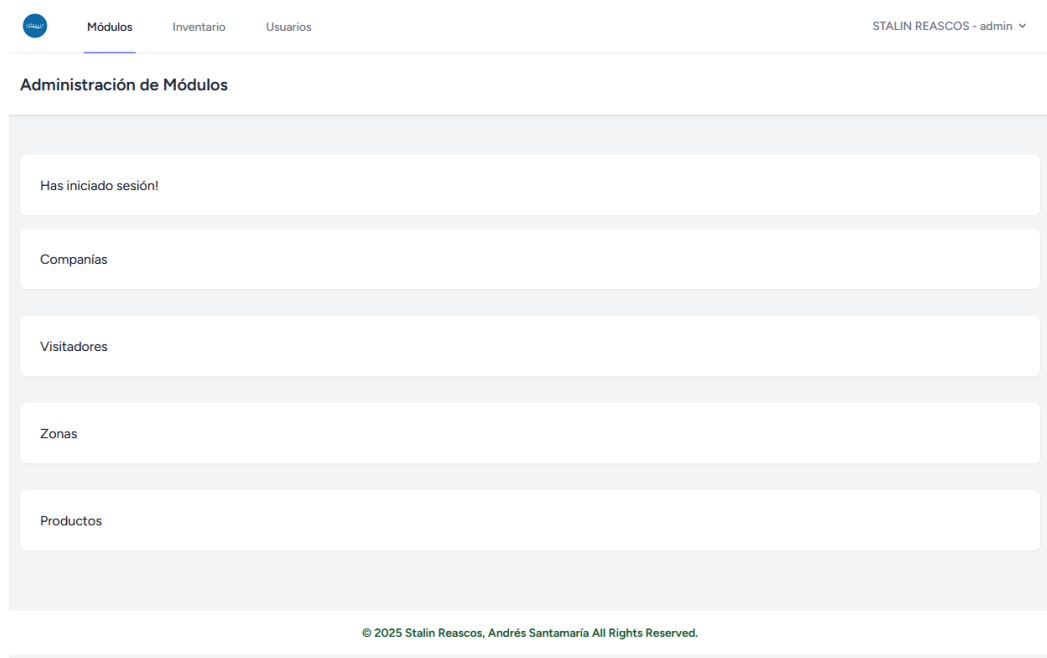


Ilustración 3. Menú de Administración de Módulos.

En la ilustración 24 se muestra el formulario correspondiente a la creación de proveedores, mientras que en la ilustración 25 se visualiza el registro generado dentro del módulo de compañías. El tiempo de creación oscila entre 25 y 30 segundos, tras lo cual el proceso se considera exitoso y completado.

Nueva Empresa

* Código

Example P(RUC) - P000000000000

Debe empezar con P seg

* Ruc

Example 151713681931001

* Razón Social

Nombre o Razon Social

Dirección

Dirección compania

Celular

Número celular

Cancelar

Guardar

Ilustración 4. Formulario, Creación de Proveedor.

muestrasmedicas-app.test/dashboard/company

Módulos Inventario Usuarios

STALIN REASCOS - admin

ADMINISTRACIÓN PROVEEDORES

Buscar...

CÓDIGO	RAZÓN SOCIAL/NOMBRE	DIRECCIÓN	TELÉFONO	TIPO	CREADO POR	ACCIONES
P1719307967001 Ruc 1719307967001	DENTAL AYALA	AMERICA Y COLON SN 25	0983443313	Proveedores	STALIN REASCOS 2025-06-17 15:17:22	Editar Eliminar
P1719307975001 Ruc 1719307975001	DENTAL ALTAMIRANO	COLON Y MERCADILLO	0000000000	Proveedores	STALIN REASCOS 2025-06-17 15:39:41	Editar Eliminar
P1709217457001 Ruc 1709217457001	DENTAL DENTSPLAY	AMRICA 27-29 Y LAS CASAS	0998403968	Proveedores	STALIN REASCOS 2025-06-17 15:41:21	Editar Eliminar
P1719307967002 Ruc 1719307967001	DENTAL AYALA 2		0000000000	Proveedores	STALIN REASCOS 2025-06-17 16:48:33	Editar Eliminar

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 5. Lista de Proveedor Creado.

En la ilustración 26 se muestra el formulario correspondiente a la creación de visitantes, mientras que en la ilustración 27 se visualiza el registro generado dentro del módulo de Visitadores. Mencionar que los visitantes se encuentran ubicados por Zonas mediante un

código, siendo 1 la zona Sierra, 2 la zona Costa y 3 la zona Oriente. El tiempo de creación oscila entre 25 y 30 segundos, tras lo cual el proceso se considera exitoso y completado.

Nuevo Representante

* Código

Example VEN 1

* Nombres y Apellidos

Nombres y apellidos del representante

Email

example@example.com

Celular

Numero celular

Cancelar

Guardar

Ilustración 6. Formulario Creación de un Visitador.

Módulos

Inventario

Usuarios

STALIN REASCOS - admin

ADMINISTRACIÓN REPRESENTANTES

Buscar...

CÓDIGO	NOMBRES	CELULAR	ESTADO	CREADO POR	ACCIONES
AMA 1	ALEJANDRO MANTILLA amantilla@yahoo.com	0998066862	Activo	STALIN REASCOS 2025-06-17 17:10:43	Editar Eliminar
EJM 1	EJEMPLO 1 PRUEBAS INGRESO	0000000000	Activo	STALIN REASCOS 2025-06-17 17:12:03	Editar Eliminar
DZQ 1	DISTRIBUIDORA QUITO	0000000000	Activo	STALIN REASCOS 2025-06-19 10:19:08	Editar Eliminar
DZQ 2	DISTRIBUIDORA GUAYAQUIL	0000000000	Activo	STALIN REASCOS 2025-06-19 10:19:46	Editar Eliminar
DZC 3	DISTRIBUIDORA CUENCA	0000000000	Activo	STALIN REASCOS 2025-06-19 10:20:09	Editar Eliminar
TEST1	PRUEBAS DE ELIMINACION	0000000000	Activo	STALIN REASCOS 2025-06-19 10:33:54	Editar Eliminar

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 7. Lista de los Visitadores.

Nueva Zona

* Código

Ejemplo 3004

^

v

* Descripción

Describe el nombre de la zona

* Seleccione Representante

AMA 1 - ALEJANDRO MANTILLA

v

Cancelar

Guardar

ADMINISTRACIÓN ZONAS

Ilustración 9. Listado de las Zonas.

productos. El tiempo de creación oscila entre 25 y 30 segundos, tras lo cual el proceso se considera exitoso y completado.

Nuevo Producto

* Código

Example COD01

* Descripción

Láminas de cera

Código de Barras

Example 010321654879

* Proveedor

P1709217457001 - DENTAL DENTSPRAY

Imagen

Examinar...

No se ha sele...ngún archivo.

Cancelar

Guardar

Ilustración 10. Formulario para crear Productos.

Módulos

Inventario

Usuarios

STALIN REASCOS - admin ▾

ADMINISTRACIÓN PRODUCTOS

Buscar...

CÓDIGO	DESCRIPCIÓN	CÓDIGO DE BARRAS	IMAGEN	CREADO POR	ACCIONES
LAC01 Proveedor: DENTAL AYALA	LAMINAS DE CERA	0000000000	Sin imagen	STALIN REASCOS 2025-06-24 11:07:34	Editar Eliminar
LAC02 Proveedor: DENTAL DENTSPLAY	LAMINAS DE CERA 60CM	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:52:37	Editar Eliminar
LAC03 Proveedor: DENTAL ALTAMIRANO	LAMINAS DE CERA 60 CM COLOR NARANJA	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:53:33	Editar Eliminar
COLO1 Proveedor: DENTAL DENTSPLAY	COLORIMETRO IVOSTAR A1	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:54:00	Editar Eliminar
COLO2 Proveedor: DENTAL DENTSPLAY	COLORIMETRO IVOSTAR A2	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:54:27	Editar Eliminar
TAB01 Proveedor: DENTAL ALTAMIRANO	TABLETA DIENTES SUP A1	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:54:52	Editar Eliminar
TAB02 Proveedor: DENTAL ALTAMIRANO	TABLETA DIENTES INF A1	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:55:08	Editar Eliminar

Showing 1 to 7 of 9 results

◀

1

2

▶

Ilustración 11. Listado de Productos.

2.12 Acceso Usuario Bodega

En la ilustración 32 se presenta el menú de administración de inventarios, en el cual se encuentran distintos apartados que permiten al administrador y al usuario de bodega gestionar la creación de Ingresos, Egresos, editar un Lote, visualizar el Kardex y el Stock General de las muestras médicas. En este apartado se evaluará el tiempo de respuesta y el flujo del proceso correspondiente.

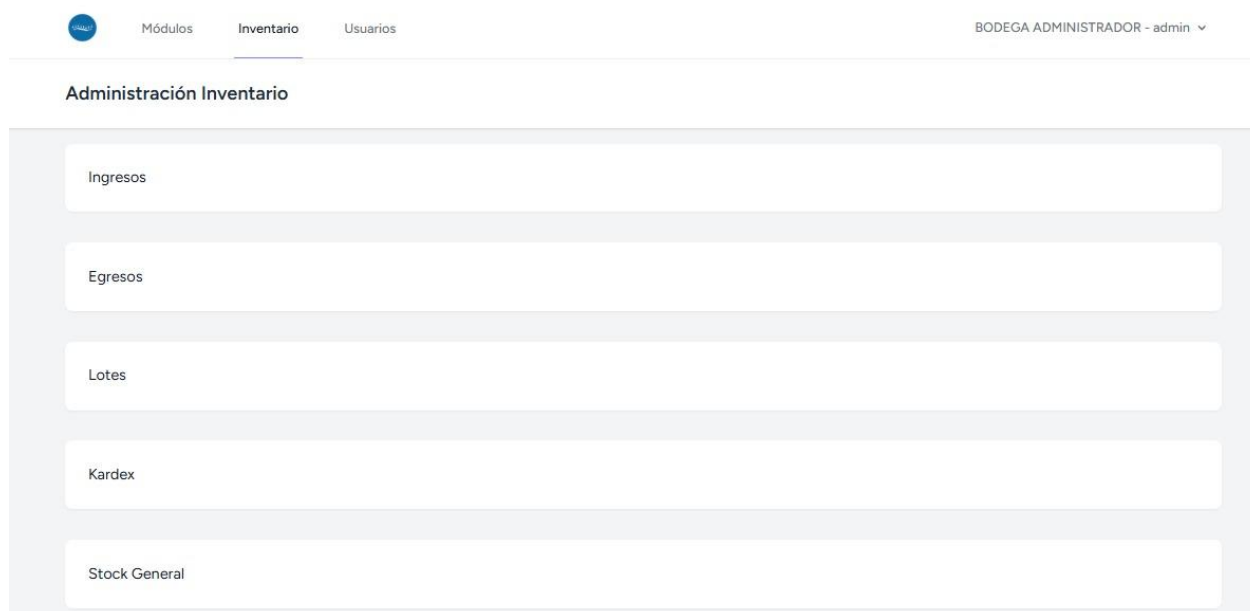


Ilustración 12. Menú Administración Inventario.

En la ilustración 33 se visualiza el módulo general para el registro de los medicamentos, la ilustración 34 muestra el formulario correspondiente a la creación de ingresos de medicamentos, mientras que en la ilustración 35 se visualiza el registro generado dentro del módulo de ingresos. El tiempo de creación oscila entre 25 segundos en adelante ya que depende de la magnitud de medicamentos que ingresan por lotes, tras lo cual el proceso se considera exitoso y completado. Cabe mencionar que los datos solicitados son campos obligatorios.

MódulosInventarioUsuarios

STALIN REASCOS - admin

DATOS ENTRADA

FECHA ENTRADA

21 / 07 / 2025

PROVEEDOR

Seleccione un proveedor

PRODUCTO

Seleccione un producto

Cantidad

1

Lote

NUMERO LOTE

Fabricación

dd / mm / aaaa

Vencimiento

dd / mm / aaaa

Agregar

Observaciones

Escriba sus comentarios aquí...

DETALLES

IDPROD	PRODUCTO	LOTE	CANTIDAD	F. FABRICACIÓN	F. VENCIMIENTO	ACCIÓN
TOTAL UNIDADES: 0						

Cancelar

Crear Entrada

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 13. Vista de Ingreso de Medicamentos.

DATOS ENTRADA

* FECHA ENTRADA

21 / 07 / 2025

* PROVEEDOR

DENTAL DENTSPLAY

* PRODUCTO

COLORIMETRO IVOSTAR A1

* Cantidad

100

* Lote

COLIV00100

* Fabricación

21 / 07 / 2025

* Vencimiento

31 / 07 / 2029

Agregar

Observaciones

COMPRA COLORIMETROS IVOSTAR A1

Ilustración 14. Formulario para Ingresar Medicamentos.

DETALLES

IDPROD	PRODUCTO	LOTE	CANTIDAD	F. FABRICACIÓN	F. VENCIMIENTO	ACCIÓN
4	COLORIMETRO IVOSTAR A1	coliv00100	100	2025-07-21	2029-07-31	Eliminar
5	COLORIMETRO IVOSTAR A2	coliv00102	100	2025-07-21	2027-07-31	Eliminar

TOTAL UNIDADES:
200

Cancelar

Crear Entrada

Ilustración 15. Registro de Medicamentos ingresados.

En la ilustración 36 se visualiza el catálogo completo de los productos que han sido ingresados.

ADMINISTRACIÓN INGRESOS						
Buscar...						
NUMERO ENTRADA	PROVEEDOR	FECHA	OBSERVACIONES	ACCIONES		
1	P1709217457001 DENTAL DENTSPLAY	2025-06-30	COMPRA MATERIALES DENTALES DENTAL DENTSPLAY	PDF	Visualizar	Eliminar
2	P1719307967001 DENTAL AYALA	2025-06-30	COMPRA MUESTRAS DE LAMINAS DE CERA DENTAL AYALA	PDF	Visualizar	Eliminar
3	P1719307975001 DENTAL ALTAMIRANO	2025-06-30	COMPRA TABLETAS DE DIENTES COLOR A2 INFERIORES A DENTAL ALTAMIRANO	PDF	Visualizar	Eliminar
4	P1719307967001 DENTAL AYALA	2025-06-30		PDF	Visualizar	Eliminar
5	P1709217457001 DENTAL DENTSPLAY	2025-06-30	COMPRA DE MUESTRAS MEDICAS DENTAL DENTSPLAY	PDF	Visualizar	Eliminar
6	P1709217457001 DENTAL DENTSPLAY	2025-07-21	COMPRA COLORIMETROS IVOSTAR A1 Y A2	PDF	Visualizar	Eliminar

Ilustración 16. Listado de los Productos ingresados.

En la ilustración 37 se visualiza el módulo general para el registro de los medicamentos, la ilustración 38 muestra el formulario correspondiente a la creación de egresos de medicamentos donde se selecciona el medicamento que será entregado al visitador, mientras que en la ilustración 39 se visualiza el registro generado dentro del módulo de egresos. El tiempo de creación oscila entre 25 segundos en adelante ya que depende de la magnitud de

medicamentos que serán entregados a los visitantes, tras lo cual el proceso se considera exitoso y completado. Cabe mencionar que los datos solicitados son campos obligatorios.

MódulosInventarioUsuarios

STALIN REASCOS - admin

DATOS SALIDA

* REPRESENTANTE DE VENTA

Seleccione representante

* FECHA SALIDA

21/07/2025

* STOCK DISPONIBLE

Buscar en la tabla

#	CODIGO	PRODUCTO	LOTE	VENCIMIENTO	STOCK	CANTIDAD	ACCIÓN
1	COL01	COLORIMETRO IVOSTAR A1	COLIVA1001	2029-06-30	8	Digite cantidad	Asignar
2	COL02	COLORIMETRO IVOSTAR A2	COLIVA2001	2029-06-30	2	Digite cantidad	Asignar
3	LAC01	LAMINAS DE CERA	LADA0001	2030-06-30	5	Digite cantidad	Asignar
4	TAB04	TABLETA DIENTES INF A2	TABDINFA2	2028-12-30	70	Digite cantidad	Asignar
5	LAC01	LAMINAS DE CERA	LAMC0102	2029-02-02	150	Digite cantidad	Asignar
6	LAC02	LAMINAS DE CERA 60CM	LAMCDD001	2050-10-10	300	Digite cantidad	Asignar
7	LAC02	LAMINAS DE CERA 60CM	LAMCDD002	2050-10-10	300	Digite cantidad	Asignar

Observaciones

Escribe tus observaciones...

DETALLES

CODE	PRODUCTO	LOTE	VTO	CANTIDAD	ACCIÓN
Total Unidades					
0					
Cancelar		Crear Salida			

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 17. Vista de Egresos de Medicamentos.

* STOCK DISPONIBLE

Buscar en la tabla

#	CODIGO	PRODUCTO	LOTE	VENCIMIENTO	STOCK	CANTIDAD	ACCIÓN
1	COL01	COLORIMETRO IVOSTAR A1	COLIVA1001	2029-06-30	8	Digite cantidad	Asignar
2	COL02	COLORIMETRO IVOSTAR A2	COLIVA2001	2029-06-30	2	Digite cantidad	Asignar
3	LAC01	LAMINAS DE CERA	LADA0001	2030-06-30	5	Digite cantidad	Asignar
4	TAB04	TABLETA DIENTES INF A2	TABDINFA2	2028-12-30	70	Digite cantidad	Asignar
5	LAC01	LAMINAS DE CERA	LAMC0102	2029-02-02	150	Digite cantidad	Asignar
6	LAC02	LAMINAS DE CERA 60CM	LAMCDD001	2050-10-10	300	Digite cantidad	Asignar
7	LAC02	LAMINAS DE CERA 60CM	LAMCDD002	2050-10-10	300	Digite cantidad	Asignar

Ilustración 18. Selección de Medicamentos para Egresos.

DETALLES					
CODE	PRODUCTO	LOTE	VTO	CANTIDAD	ACCIÓN
COL01	COLORIMETRO IVOSTAR A1	COLIVA1001	2029-06-30	8	Eliminar
COL02	COLORIMETRO IVOSTAR A2	COLIVA2001	2029-06-30	2	Eliminar
LAC01	LAMINAS DE CERA	LADA0001	2030-06-30	5	Eliminar
Total Unidades					
15					
Cancelar Crear Salida					

Ilustración 19. Registro de Egresos.

En la ilustración 40 se visualiza el stock general de los medicamentos que posee bodega, de igual manera se puede tener un reporte del detalle total. El tiempo de impresión oscila entre 5 segundos a 1 minutos, depende mucho de la carga total del stock tras lo cual el proceso se considera exitoso y completado.

STOCK GENERAL						
Buscar en la table						
#	CODIGO	PRODUCTO	PROVEEDOR	LOTE	VENCIMIENTO	STOCK
1	TAB04	TABLETA DIENTES INF A2	DENTAL ALTAMIRANO	TABDINF A2	2028-12-30	70
2	LAC01	LAMINAS DE CERA	DENTAL AYALA	LAMC0102	2029-02-02	150
3	LAC02	LAMINAS DE CERA 60CM	DENTAL DENTSPLAY	LAMCDD001	2050-10-10	300
4	LAC02	LAMINAS DE CERA 60CM	DENTAL DENTSPLAY	LAMCDD002	2050-10-10	300
5	LAC02	LAMINAS DE CERA 60CM	DENTAL DENTSPLAY	LAMCDD003	2050-10-10	300
6	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV0010	2050-10-10	300
7	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV0011	2050-10-10	300
8	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV0013	2050-10-10	300
9	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV00100	2025-05-31	100
10	COL02	COLORIMETRO IVOSTAR A2	DENTAL DENTSPLAY	COLIV00102	2027-07-31	100

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 20. Vista Stock General de los Medicamentos.

CAPITULO III

PROPUESTA DEL DESARROLLO DEL PROYECTO TÉCNICO / ESTUDIO DE CASOS.

El sistema web desarrollado para la gestión y control de muestras médicas en la empresa HOSPIMEDIKKA maneja una arquitectura Modelo–Vista–Controlador (MVC) según la ilustración 1, la cual separa de forma clara la lógica de negocio, la presentación de la información y el acceso a los datos.

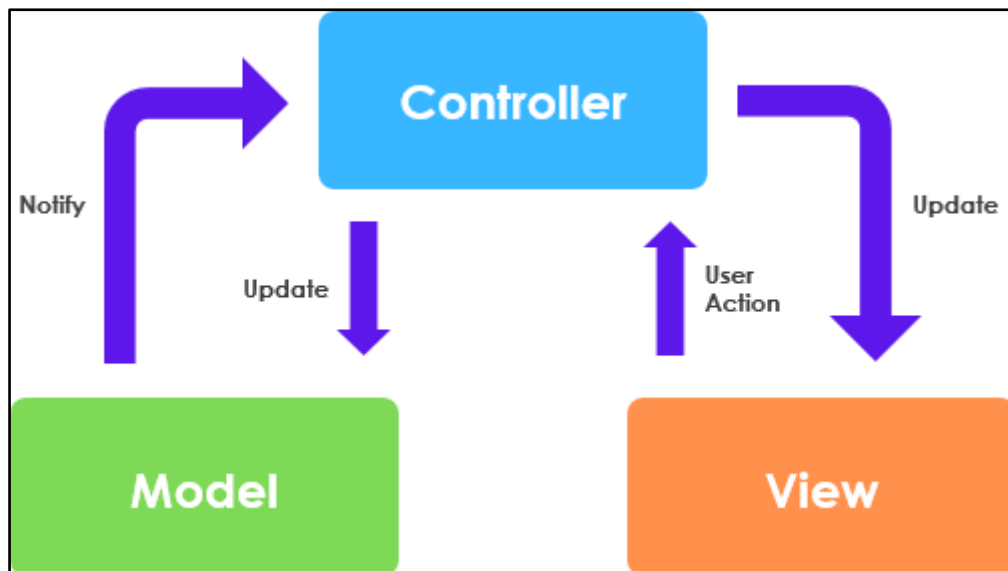


Ilustración 21. MVC del sistema web presentado.

3.1 Modelo

Gestiona la parte lógica de los datos, conectándose con la base de datos MySQL en donde almacena información sobre usuarios, productos, lotes, ingresos y egresos de las muestras médicas, garantizando integridad y consistencia. En el proyecto, esta etapa se encarga de la integridad, migraciones y consultas optimizadas necesarias para garantizar trazabilidad y reportes confiables.

3.2 Vista

Parte gráfica que permite la interacción con el usuario. Se implementa TailwindCSS para mantener un diseño moderno y JavaScript/jQuery para mejorar la interactividad. En esta sección se implementan las interfaces de usuario: dashboards, formularios de ingreso/egreso, generación de reportes en PDF y vistas de trazabilidad.

3.3 Controlador

Intermediario entre la Vista y el Modelo donde gestiona las solicitudes de los usuarios, procesando la lógica y enviando las respuestas necesarias. Por ejemplo, (se registra un ingreso o consulta el inventario), se validan los datos, interactúan con el Modelo para recuperar o almacenar información y devuelven la respuesta adecuada a la Vista.

Esta arquitectura facilita el mantenimiento, la escalabilidad y la organización del código permitiendo así futuras ampliaciones como la integración en dispositivos móviles.

3.4 Justificación del uso de Laravel

Se eligió Laravel como framework principal por las siguientes razones las cuales la convierten como favoritas:

- **Implementación Robusta del Patrón MVC:** Laravel proporciona una estructura clara y organizada, agiliza el desarrollo y facilita la colaboración en equipo.
- **Sistema de Migraciones:** Define y versiona la estructura de la base de datos, asegurando consistencia y facilidad de despliegue.
- **Seguridad:** Brinda seguridad ante vulnerabilidades comunes como inyección SQL y ataques CSRF

- **Autenticación y Control de Acceso:** Laravel Breeze como herramienta, simplifica la implementación del registro, inicio de sesión y manejo de roles de usuario.
- **Escalabilidad y Comunidad Activa:** Al ser ampliamente utilizado, cuenta con una gran comunidad y documentación extensa, asegurando soporte y posibilidades de mejora a largo plazo.

3.5 Repercusión dentro del Proyecto

La adopción del patrón Modelo–Vista–Controlador (MVC) aparte de definir la estructura del desarrollo, influyó directamente en la calidad, la seguridad y escalabilidad del sistema. Esta elección permitió que cada componente del proyecto tuviera responsabilidades claras, facilitando el trabajo colaborativo, reduciendo los tiempos de desarrollo y asegurando un producto final aceptable a las necesidades presentes y futuras de HOSPIMEDIKKA. En conjunto con Laravel se garantizó un sistema sólido, seguro, escalable y adaptable, alienable a las necesidades requeridas de la empresa, optimizando la trazabilidad y el control de las muestras médicas de forma eficiente y sostenible.

3.6 Diagrama Entidad Relación

En base a la ilustración 2, se muestra el Diagrama de Entidad Relación en donde se maneja la lógica de los datos siendo también el precursor de la base de datos. A continuación, se explica mediante el siguiente gráfico:

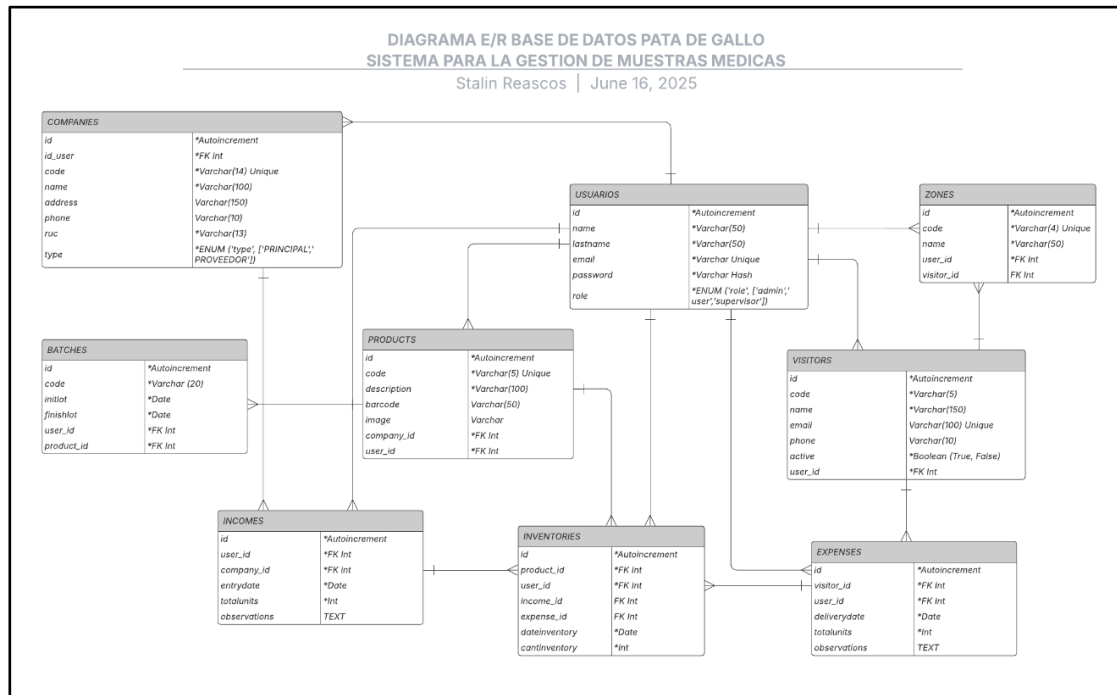


Ilustración 22. Diagrama E/R Base de Datos Pata de Gallo.

Un usuario se relaciona de forma uno a varios con zonas, visitantes, ingresos, egresos, inventarios, lotes y productos, permitiendo gestionar múltiples registros en cada entidad.

Las zonas se vinculan a un único usuario, mientras que los visitantes pueden estar asociados a varias zonas (relación varios a uno). Un visitante realiza múltiples entregas de muestras en distintas zonas (uno a varios).

Las compañías se relacionan uno a varios con productos e ingresos; cada compañía registra múltiples productos e ingresos, está asociada a un solo usuario (varios a uno). A su vez, los productos pertenecen a una compañía y a un usuario (varios a uno), y se relacionan uno a varios con inventarios y lotes.

Los ingresos y egresos se relacionan varios a uno con compañías, visitantes y usuarios, permitiendo múltiples registros por entidad. Cada egreso puede generar varios movimientos de inventario (uno a varios).

Los inventarios se vinculan varios a uno con productos, usuarios, ingresos y egresos, registrando todos los movimientos asociados. Los lotes se asocian varios a uno con productos y usuarios, permitiendo el control de múltiples lotes por producto.

3.7 DISEÑO DE LA INTERFAZ Y EXPERIENCIA DE USUARIO (UI/UX)

El diseño de la interfaz y la experiencia del usuario del sistema web para la gestión de muestras médicas se desarrolló con el objetivo de garantizar una interacción intuitiva, eficiente y accesible para todos los usuarios. Se tomó en cuenta la simplicidad en la navegación, la información clara y la adaptabilidad a distintos dispositivos. La propuesta de la interfaz se diseñó siguiendo principios de usabilidad y modernización, utilizando componentes visuales, elementos interactivos capaces de reducir errores y tiempos de aprendizaje.

Basándonos en la ilustración 3 se presenta un prototipo (Mockup) para el sistema web la cual define la distribución y funcionalidad de las principales pantallas, asegurando una navegación sencilla, clara y con experiencia fluida para el usuario.

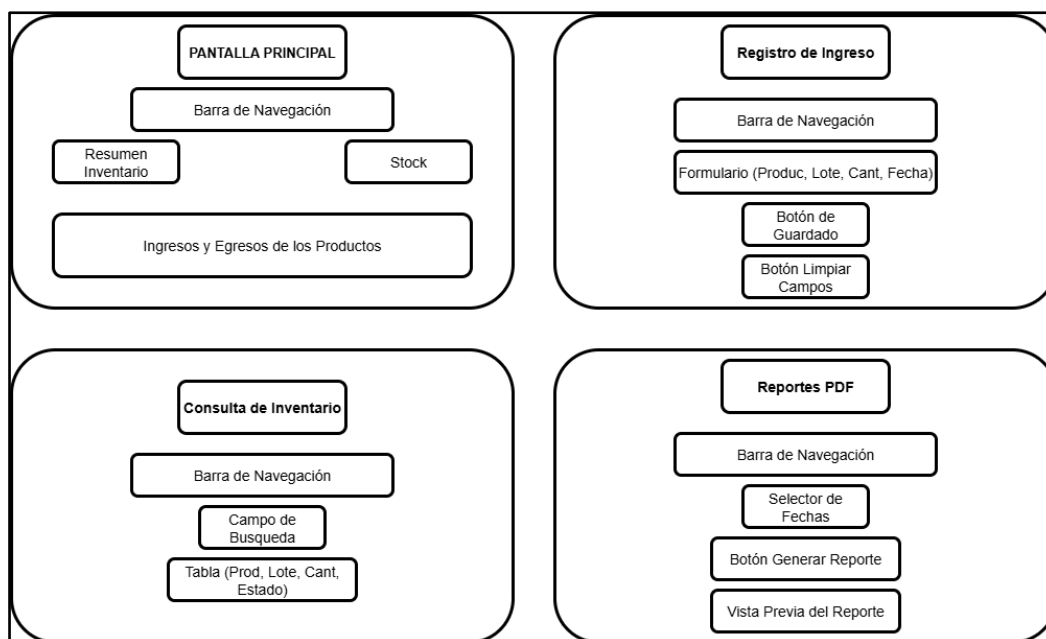


Ilustración 23. Prototipo (Mockup) del Sistema Web.

El prototipo sostiene que cada pantalla se enfoca en una función específica, con una distribución coherente y vistas consistentes contribuyendo al usuario una experiencia sencilla, eficiente y adaptable a las necesidades propuestas por la empresa HOSPIMEDIKKA.

3.8 DESARROLLO DE MÓDULOS PRINCIPALES

3.8.1 Desarrollo de Funcionalidades CRUD

En esta etapa se desarrollaron los distintos módulos para la gestión de las entidades, cada una tendrá sus respectivas funcionalidades y permisos.

Entidad	Funcionalidad	Estado
Visitadores	Crear Editar Eliminar (únicamente el usuario administrador)	Completado
Zonas	Crear Editar Eliminar (únicamente el usuario administrador)	Completado
Compañías	Crear Editar Eliminar (únicamente el usuario administrador)	Completado
Productos	Crear Editar Eliminar (únicamente el usuario administrador)	Completado
Ingreso Productos	Crear Editar Eliminar (únicamente el usuario administrador)	Completado

Egreso Productos	Crear	Completado
	Editar	
	Eliminar (únicamente el usuario administrador)	
Usuarios	Crear	Completado
	Editar	
	Eliminar (únicamente el usuario administrador)	
Lotes	Creación de lotes que posee un producto, con sus fechas de elaboración y vencimiento.	Completado

Tabla 4. Desarrollo de Funcionalidades CRUD.

3.9 Archivos de Migraciones

Las migraciones de nuestro proyecto serán las que definan la estructura de las tablas de la base de datos, así como también, las relaciones entre estas entidades y atributos.

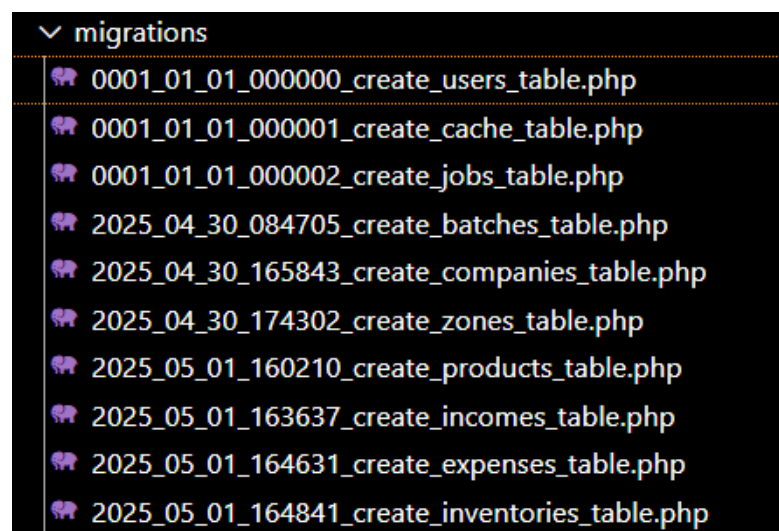


Ilustración 24. Tablas a crear.

A continuación, se representa desde la ilustración 4 hasta la ilustración 12, como se encuentran programadas las migraciones de la base de datos, las cuales serán fundamentales para un correcto almacenamiento de la información.

```
database > migrations > 0001_01_01_000000_create_users_table.php
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name',50);
17             $table->string('lastname',50);
18             $table->string('email')->unique();
19             $table->timestamp('email_verified_at')->nullable();
20             $table->string('password');
21             $table->enum('role',['admin','user','supervisor'])->default('user');
22             $table->softDeletes('deleted_at')->nullable();
23             $table->rememberToken();
24             $table->timestamps();
25         });
26 }
```

Ilustración 25. Código de migración para la tabla “users”

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('companies', function (Blueprint $table) {
15             $table->id();
16             $table->string('code', 14)->unique();
17             $table->string('name', 100);
18             $table->string('address', 150)->nullable();
19             $table->string('phone', 10)->nullable();
20             $table->string('ruc', 13);
21             $table->enum('type', ['major', 'supplier'])->default('supplier');
22             $table->foreignId('user_id')->constrained();
23             $table->timestamps();
24             $table->softDeletes('deleted_at')->nullable();
25         });
26     }
27
28     /**
29      * Reverse the migrations.
30      */
31     public function down(): void
32     {
33         Schema::dropIfExists('companies');
34     }
35 };
36

```

Ilustración 26. Código de migración para la tabla “companies”

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('products', function (Blueprint $table) {
15             $table->id();
16             $table->string('code')->unique();
17             $table->string('description', 100);
18             $table->string('barcode', 50)->nullable();
19             $table->string('image')->nullable();
20             $table->foreignId('company_id')->constrained();
21             $table->foreignId('user_id')->constrained();
22             $table->timestamps();
23             $table->softDeletes('deleted_at')->nullable();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      */
30     public function down(): void
31     {
32         Schema::dropIfExists('products');
33     }
34 };
35

```

Ilustración 27. Código de migración para la tabla “products”

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('visitors', function (Blueprint $table) {
15               $table->id();
16               $table->string('code', 5);
17               $table->string('name', 150);
18               $table->string('email')->unique()->nullable();
19               $table->string('phone', 10)->nullable();
20               $table->boolean('active')->default(true);
21               $table->foreignId('user_id')->constrained();
22
23               $table->timestamps();
24               $table->softDeletes('deleted_at')->nullable();
25           });
26
27           Schema::create('zones', function (Blueprint $table) {
28               $table->id();
29               $table->string('code', 4)->unique();
30               $table->string('name', 50);
31               $table->foreignId('user_id')->constrained();
32               $table->foreignId('visitor_id')->nullable()->constrained();
33               $table->timestamps();
34               $table->softDeletes('deleted_at')->nullable();
35           });
36       }
37

```

Ilustración 28. Código de migración para la tabla “visitors y zones”

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('batches', function (Blueprint $table) {
15               $table->id();
16               $table->string('code', 20);
17               $table->date('initlot');
18               $table->date('finishtlot');
19               $table->foreignId('user_id')->constrained();
20
21               $table->timestamps();
22               $table->softDeletes('deleted_at')->nullable();
23           });
24       }
25
26       /**
27        * Reverse the migrations.
28        */
29       public function down(): void
30       {
31           Schema::dropIfExists('batches');
32       }
33   };
34

```

Ilustración 29. Código de migración para la tabla “batches”

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('incomes', function (Blueprint $table) {
15               $table->id();
16               $table->foreignId('user_id')->constrained();
17               $table->foreignId('company_id')->constrained();
18               $table->date('entrydate');
19               $table->integer('totalunits');
20               $table->text('observations')->nullable();
21               $table->timestamps();
22               $table->softDeletes('deleted_at')->nullable();
23           });
24       }
25
26       /**
27        * Reverse the migrations.
28        */
29       public function down(): void
30       {
31           Schema::dropIfExists('incomes');
32       }
33   };
34

```

Ilustración 30. Código de migración para la tabla “incomes”

```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10       * Run the migrations.
11       */
12       public function up(): void
13       {
14           Schema::create('inventories', function (Blueprint $table) {
15               $table->id();
16               $table->foreignId('user_id')->constrained();
17               $table->foreignId('product_id')->constrained();
18               $table->foreignId('batch_id')->constrained();
19               $table->foreignId('income_id')->nullable()->constrained()->onDelete('cascade');
20               $table->foreignId('expense_id')->nullable()->constrained()->onDelete('cascade');
21               $table->date('dateinventory');
22               $table->integer('cantinventory');
23               $table->timestamps();
24               $table->softDeletes('deleted_at');
25           });
26       }
27
28       /**
29        * Reverse the migrations.
30        */
31       public function down(): void
32       {
33           Schema::dropIfExists('inventories');
34       }
35   };
36
37

```

Ilustración 31. Código de migración para la tabla “inventories”

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('expenses', function (Blueprint $table) {
15             $table->id();
16             $table->foreignId('user_id')->constrained();
17             $table->foreignId('visitor_id')->constrained();
18             $table->date('deliverydate');
19             $table->integer('totalunits');
20             $table->text('observations')->nullable();
21             $table->timestamps();
22             $table->softDeletes('deleted_at')->nullable();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('expenses');
32     }
33 };
34

```

Ilustración 32. Código de migración para la tabla “expenses”

3.10 Controladores

Desde la ilustración 13 hasta la ilustración 20 se muestra cierta porción del código de cada controlador, los controladores serán los encargados de realizar todo el proceso lógico de las solicitudes entre el Frontend con el backend de nuestro sistema web.

```

1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Company;
7 use App\Models\Income;
8 use Barryvdh\DomPDF\Facade\Pdf;
9 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\Auth;
12 use Illuminate\Support\Facades\Validator;
13
14 class CompanyController extends Controller
15 {
16     /**
17      * Display a listing of the resource.
18      */
19     use AuthorizesRequests;
20
21     public function index()
22     {
23         $companies = Company::orderBy('code', 'asc')->paginate(7);
24         return view('dashboard.companies.index', compact('companies'));
25     }
26
27     public function companyPdf(){
28         $companies = Company::orderBy('code', 'asc')->get();
29         $pdf = Pdf::loadView('dashboard.companies.listpdf', compact('companies'));
30         return $pdf->stream('list_company.pdf');
31     }
32 }
33

```

Ilustración 33. Código del controlador de company.

```
1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Expense;
7 use App\Models\Inventory;
8 use App\Models\Product;
9 use App\Models\Visitor;
10 use Barryvdh\DomPDF\Facade\Pdf;
11 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
12 use Illuminate\Http\Request;
13 use Illuminate\Support\Facades\Auth;
14 use Illuminate\Support\Facades\DB;
15 use Illuminate\Support\Facades\Validator;
16
17 class ExpenseController extends Controller
18 {
19     /**
20      * Display a listing of the resource.
21      */
22     use AuthorizesRequests;
23
24     public function index()
25     {
26         $visitors = Visitor::orderBy('code', 'asc')->get();
27         $products = Product::orderBy('code', 'asc')->get();
28         $expenses = Expense::orderBy('id', 'asc')->get();
29         return view('expenses.index', compact('expenses','visitors','products'));
30     }
31
32     /**
33      * Show the form for creating a new resource.
34      */
35     public function create()
36     {
37         $visitors = Visitor::orderBy('code', 'asc')->get();
```

Ilustración 34. Código del controlador de expense.

```

1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Batch;
7 use App\Models\Company;
8 use App\Models\Income;
9 use App\Models\Inventory;
10 use App\Models\Product;
11 use Barryvdh\DomPDF\Facade\Pdf;
12 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
13 use Illuminate\Http\Request;
14 use Illuminate\Support\Facades\Auth;
15 use Illuminate\Support\Facades\DB;
16
17 class IncomeController extends Controller
18 {
19     /**
20      * Display a listing of the resource.
21      */
22     use AuthorizesRequests;
23
24     public function index()
25     {
26         $incomes = Income::orderBy('id','asc')->paginate(10);
27         $companies = Company::has('product')->get();
28         return view('incomes.index',compact('incomes','companies'));
29     }
30
31
32
33     public function entryPdf($id){
34
35         $id = intval($id);
36         $entry = Income::where('id',$id)->first();
37         $pdf = Pdf::loadView('incomes.listpdf', compact('entry'));

```

Ilustración 35. Código del controlador de income.

```

1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Inventory;
7 use App\Models\Product;
8 use Barryvdh\DomPDF\Facade\Pdf;
9 use Illuminate\Database\Eloquent\SoftDeletes;
10 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
11 use Illuminate\Http\Request;
12
13 class InventoryController extends Controller
14 {
15     /**
16      * Display a listing of the resource.
17      */
18     use SoftDeletes;
19     use AuthorizesRequests;
20     public function index()
21     {
22         $inventories = Inventory::groupBy('id')->paginate(7);
23         return view('inventories.index', compact('inventories'));
24     }
25
26     public function inventoryPdf(){
27
28         $inventories = Inventory::orderBy('dateinventory', 'asc')->get();
29         $pdf = Pdf::loadView('inventories.listpdf', compact('inventories'));
30         return $pdf->stream('list_inventory.pdf');
31     }
32
33
34     public function getStock(Product $product)
35     {
36         $entradas = Inventory::where('product_id', $product->id)->whereNotNull('income_id')->sum('cantinventory');
37         $salidas = Inventory::where('product_id', $product->id)->whereNotNull('expense_id')->sum('cantinventory');

```

Ilustración 36. Código del controlador de inventory.

```

1  <?php
2
3  namespace App\Http\Controllers\Dashboard;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Company;
7  use App\Models\Product;
8  use Barryvdh\DomPDF\Facade\Pdf;
9  use Illuminate\Database\Eloquent\SoftDeletes;
10 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
11 use Illuminate\Http\Request;
12 use Illuminate\Support\Facades\Auth;
13 use Illuminate\Support\Facades\Validator;
14
15 class ProductController extends Controller
16 {
17     /**
18      * Display a listing of the resource.
19      */
20     use AuthorizesRequests;
21
22     public function index()
23     {
24         $companies = Company::orderBy('code', 'asc')->get();
25         $products = Product::orderBy('code', 'asc')->paginate(7);
26         return view('dashboard.products.index', compact('products', 'companies'));
27     }
28
29     public function productPdf(){
30
31         $products = Product::with('company')->orderBy('code', 'asc')->get();
32         $pdf = Pdf::loadView('dashboard.products.listpdf', compact('products'));
33         return $pdf->stream('list_products.pdf');
34     }
35 }
36
37 /**

```

Ilustración 37. Código del controlador de product.

```

1  <?php
2
3  namespace App\Http\Controllers\Dashboard;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\User;
7  use Illuminate\Http\Request;
8
9  class UserController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      */
14     public function index()
15     {
16         $user = User::find(1);
17         dd($user->zone[0]->name);
18     }
19
20     /**
21      * Show the form for creating a new resource.
22      */
23     public function create()
24     {
25         //
26     }
27
28     /**
29      * Store a newly created resource in storage.
30      */
31     public function store(Request $request)
32     {
33         //
34     }
35
36     /**
37      * Display the specified resource.

```

Ilustración 38. Código del controlador de user.

```

1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Visitor;
7 use Barryvdh\DomPDF\Facade\Pdf;
8 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
9 use Illuminate\Http\Request;
10 use Illuminate\Support\Facades\Auth;
11 use Illuminate\Support\Facades\Validator;
12
13 class VisitorController extends Controller
14 {
15
16     use AuthorizesRequests;
17
18     public function index()
19     {
20         $visitors = Visitor::orderBy('code', 'asc')->paginate(7);
21         return view('dashboard.visitors.index', compact('visitors'));
22     }
23
24     public function visitorPdf(){
25
26         $visitors = Visitor::with('zone')->orderBy('code', 'asc')->get();
27         $pdf = Pdf::loadView('dashboard.visitors.listpdf', compact('visitors'));
28         return $pdf->stream('list_visitors.pdf');
29     }
30
31     /**
32      * Show the form for creating a new resource.
33      */
34     public function create()
35     {
36         return view('dashboard.visitors.create');
37     }
38 }

```

Ilustración 39. Código del controlador de visitor.

```

1 <?php
2
3 namespace App\Http\Controllers\Dashboard;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Visitor;
7 use App\Models\Zone;
8 use Barryvdh\DomPDF\Facade\Pdf;
9 use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\Auth;
12 use Illuminate\Support\Facades\Validator;
13
14 class ZoneController extends Controller
15 {
16     /**
17      * Display a listing of the resource.
18      */
19     use AuthorizesRequests;
20
21     public function index()
22     {
23         $zones = Zone::orderBy('code', 'asc')->paginate(7);
24         $visitors = Visitor::where('active', true)->orderBy('code', 'asc')->get();
25         return view('dashboard.zones.index', compact('zones', 'visitors'));
26     }
27
28     /**
29      * Show the form for creating a new resource.
30      */
31     public function create()
32     {
33         $visitors = Visitor::where('active', true)->orderBy('code', 'asc');
34         $this->authorize('create', Zone::class);
35         return view('dashboard.zones.create', compact('visitors'));
36     }
37 }

```

Ilustración 40. Código del controlador de zone.

CONCLUSIONES

El desarrollo del sistema web permitió optimizar el proceso de gestión de muestras médicas en HOSPIMEDIKKA, dejando atrás el sistema tradicional (Kardex) logrando automatizar el registro, seguimiento y trazabilidad de las muestras. Esto redujo el margen de error aumentando la eficiencia en el manejo de datos. La digitalización del proceso mejoró la organización interna de la empresa, agilizando el acceso a la información por parte del personal encargado.

Se analizó detalladamente del proceso actual de entrega (Kardex), el registro y control de muestras médicas se identificó múltiples errores, como la falta de trazabilidad y la existencia de registros manuales propensos a errores. Esta etapa es importante para justificar la necesidad de una solución tecnológica que optimice el flujo de información.

La propuesta tecnológica se diseñó considerando aspectos de eficiencia, facilidad de uso y control de las muestras médicas. Se planteó una interfaz amigable que registra y rastrear muestras en tiempo real. Esto no solo aborda los problemas detectados en el análisis inicial, sino que permitirá una futura ampliación del sistema hacia otros dispositivos. El diseño fue validado mediante prototipos antes de su implementación final.

El sistema se desarrolló en base a la arquitectura MVC con Laravel y MySQL, facilitando una estructura organizada y mantenible del código. Esta tecnología permite una correcta separación entre la lógica del negocio, la presentación de datos y la gestión de la base de datos. Se automatizó procesos como el registro, seguimiento y control de muestras, obteniendo un margen de error mínimo. La implementación se realizó en base a buenas prácticas de desarrollo y documentación.

En base al desarrollo del sistema se obtuvieron pruebas funcionales y de usabilidad con usuarios finales, confirmando así que la herramienta cumple con los requerimientos necesarios. Se observó mejoras en la eficiencia del registro obteniendo un mayor control sobre la ubicación y estado de las muestras médicas.

La implementación del sistema automatizado representa un avance significativo para la gestión de muestras médicas en la empresa HOSPIMEDIKKA. Mejorando la trazabilidad de las muestras, la reducción de errores humanos asociados al manejo manual (kardex). Este proyecto demuestra cómo la tecnología puede ser aplicada eficazmente a contextos clínicos con impactos positivos inmediatos, facilitando el desempeño de los trabajadores.

RECOMENDACIONES

A futuro (expandir el diseño para un dispositivo móvil)

Se recomienda el desarrollo e implementación de una versión móvil del sistema web, facilitando el acceso y el registro de muestras médicas en tiempo real desde cualquier ubicación que se encuentren los visitantes. Esta mejora permitiría a los usuarios de bodega realizar seguimientos más ágiles y mantener la trazabilidad incluso fuera del entorno laboral.

Así mismo, se sugiere integrar notificaciones y alertas inteligentes que permitan mejorar el control de tiempos, de registro y prevenir errores de digitación. Esta versión móvil aumentaría significativamente la accesibilidad, eficiencia del sistema y la experiencia de usuario.

Práctica (capacitación al personal y levantamiento de un servidor)

Se sugiere realizar capacitaciones dirigidas al personal que utilizará el sistema, con el objetivo de garantizar un correcto uso de la herramienta. Esta formación debe enfocarse en el manejo del sistema, resolución de incidencias y buenas prácticas en la gestión de muestras. Esto ayudará a una rápida adaptación al nuevo sistema y minimizará errores por desconocimiento dando mayor confianza al personal con el uso de las tecnologías digitales.

Se recomienda desplegar el sistema en un servidor que asegure la alta disponibilidad y el respaldo de la información registrada. Existen 2 alternativas viables un servidor dedicado que es ideal un control total sobre la infraestructura, y una solución en la nube que es la opción más adoptada por su escalabilidad, facilidad de almacenamiento y menores costos iniciales.

Estas implementaciones deben tener políticas de respaldo automáticas y medidas de seguridad como cifrado y control de accesos, esto asegurará la continuidad del servicio y la integridad de la información médica.

BIBLIOGRAFÍA

- SafetyCulture. (2024). *Sistema de Inventario: Definición, tipos y métodos*. SafetyCulture. <https://safetyculture.com/es/temas/manejo-de-inventario/control-de-inventarios/>
- Vera, R. (31 de marzo de 2021). *Qué es Laravel: Características y Ventajas*. OpenWebinars.net <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/>
- MySQL. (s.f.). *Base de Datos MySQL*. MySQL.com <https://www.mysql.com/products/enterprise/database/>
- Erickson, J. (29 de agosto de 2024). *MySQL: Qué es y Cómo se usa*. <https://www.oracle.com/latam/mysql/what-is-mysql/>
- TailwindCSS. (s.f.). *Tailwind CSS – Framework de Utilidades para CSS*. TailwindCSS. [Tailwind CSS - Rapidly build modern websites without ever leaving your HTML.](https://tailwindcss.com/)
- Guerrero. N. (26 de julio de 2019). *¿Qué es Laragon?* Programa en Línea. <https://www.programaenlinea.net/que-es-es-laragon/>
- jQuery. (s.f.). *¿Qué es jQuery?* jQuery.com [jQuery](https://jquery.com/)
- Guevara, N., Valencia, A., Romero, E., Quiroz, C., Arenas, M., Salcedo, M. (abril de 2022). *La trazabilidad en las mediciones del laboratorio clínico: impacto en la calidad y seguridad del paciente*. [La trazabilidad en las mediciones del laboratorio clínico: impacto en la calidad y seguridad del paciente](#)
- Agencia Nacional de Regulación, Control y Vigilancia Sanitaria (ARCSA). (2017). Estatuto Orgánico de la Agencia de Regulación y Vigilancia Sanitaria (Resolución ARCSA-DE-016-2017-JCGO). Registro Oficial Edición Especial No. 18, 22 de junio de 2017.

Sampieri, R. H., Collado, C. F., & Lucio, P. B. (2014). *Metodología de la investigación*. Dialnet. <https://dialnet.unirioja.es/servlet/libro?codigo=775008>

Dr. Shinde, V. (28 de diciembre de 2023). *Análisis cuantitativo y cualitativo en productos farmacéuticos*. Veeprho. [Análisis cuantitativo y cualitativo en productos farmacéuticos. - Veeprho](#)

Sarmiento, L. (2 de agosto de 2024). *¿Qué es un Kardex y por qué es importante para tu empresa?* Acatha. [¿Qué es un kardex y por qué es importante para tu empresa? - Acatha](#)

ANEXOS

MANUAL DE USUARIO

Login de Usuario

Para poder ingresar dentro de la red local al sistema web nos debemos dirigir a cualquier navegador (Firefox, Mozilla) se debe tomar en cuenta que el sistema solicita navegadores con al menos las dos últimas actualizaciones, no soporta IE (Internet Explorer).

Una vez dentro de nuestro navegador vamos a dirigirnos a la siguiente dirección:
<http://muestrasmedicas-app.test/>

Inicio de Sesión

1. Una vez que estamos dentro del sitio web, vamos a visualizar una pantalla de inicio como se muestra a continuación:

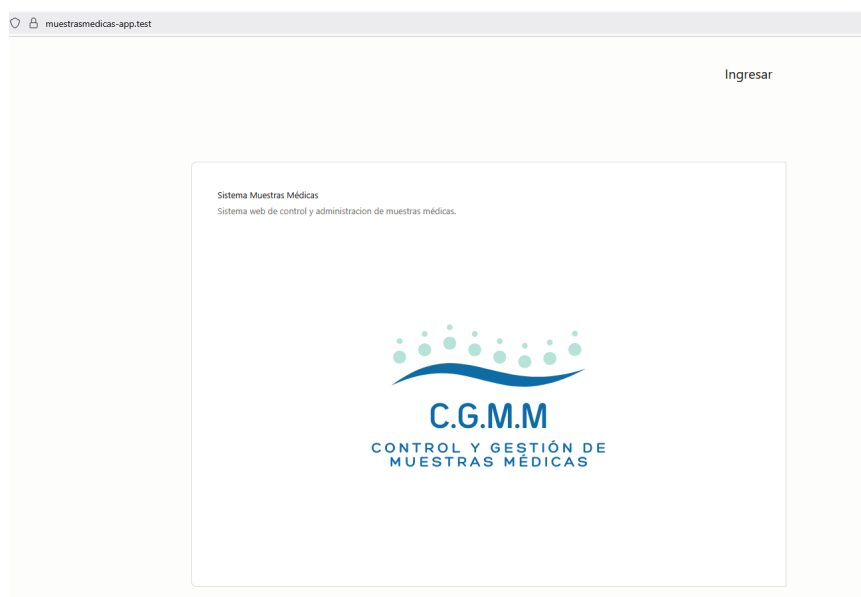


Ilustración 41. Vista Principal del Sistema de Control y Gestión de Muestras Médicas.

2. Para poder ingresar al sistema cada usuario deberá digitar su correo y contraseña proporcionados con anterioridad por el administrador del sistema, ya con estos datos nos dirigimos a la opción **“Ingresar”** ubicada en la parte superior derecha de nuestra pantalla de bienvenida, posteriormente damos clic en dicha opción.
3. En este paso el sistema nos mostrara otra vista en donde tendremos que digitar el correo electrónico y contraseña, damos clic en el botón de **“entrar”** para su respectiva validación.

Pantalla de ingreso credenciales usuario

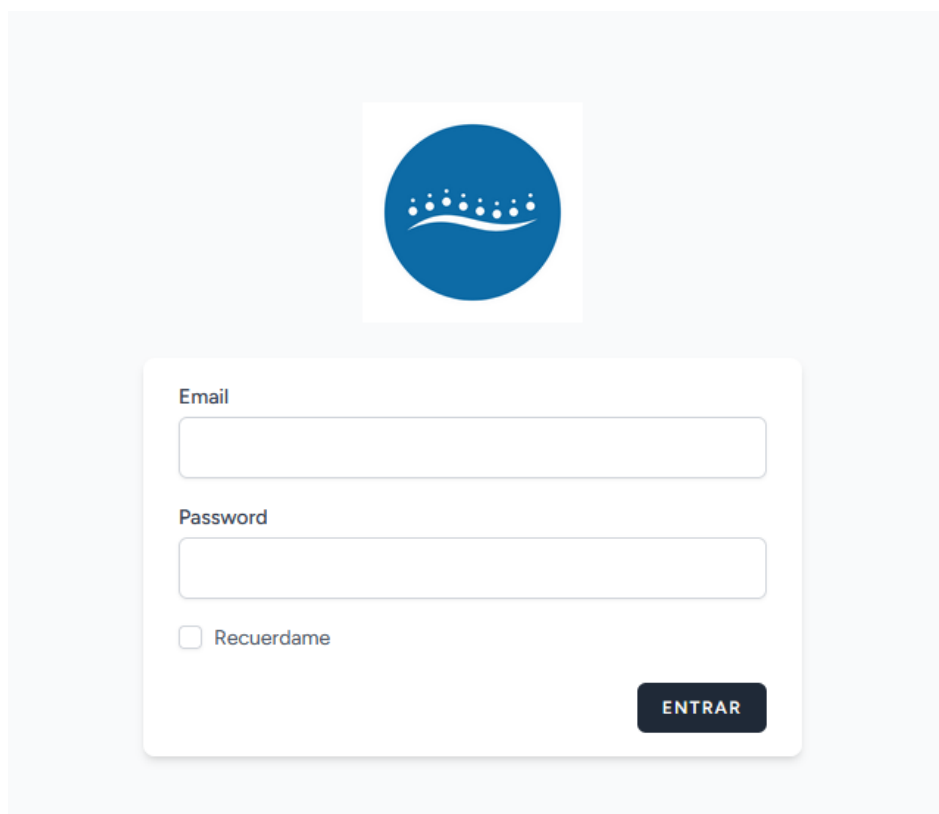


Ilustración 42. Pantalla Ingreso de Credenciales.

4. Una vez validadas las credenciales de usuario y al ser estas correctas, el sistema nos dirigirá a la pantalla de gestión, dentro de esta pantalla tendremos

los diferentes módulos de administración en los cuales se realizarán las diferentes transacciones u operaciones del sistema.

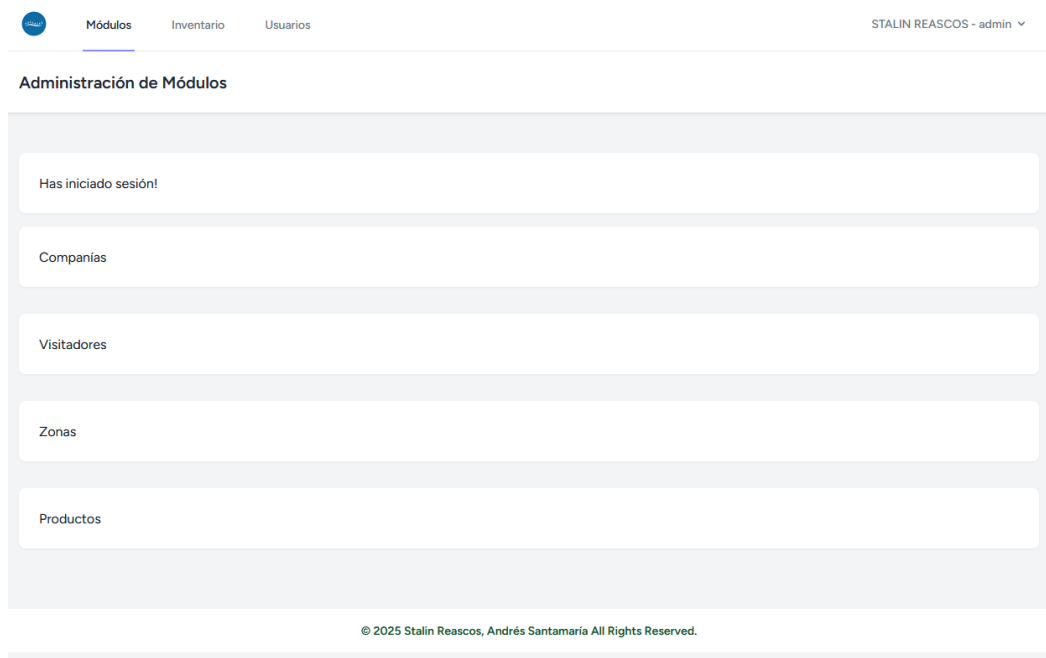


Ilustración 43. Vista Administración Módulo.

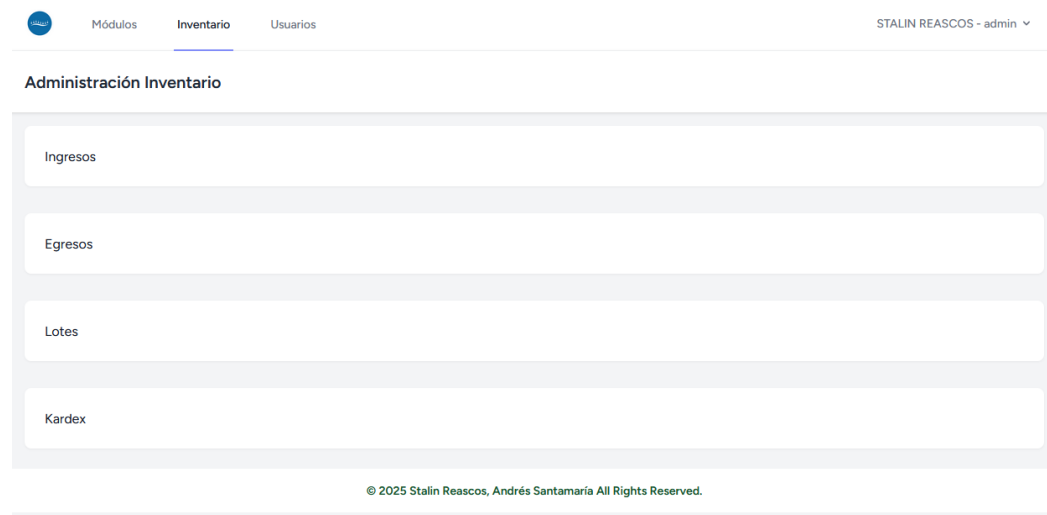


Ilustración 44. Vista Administración de Inventario

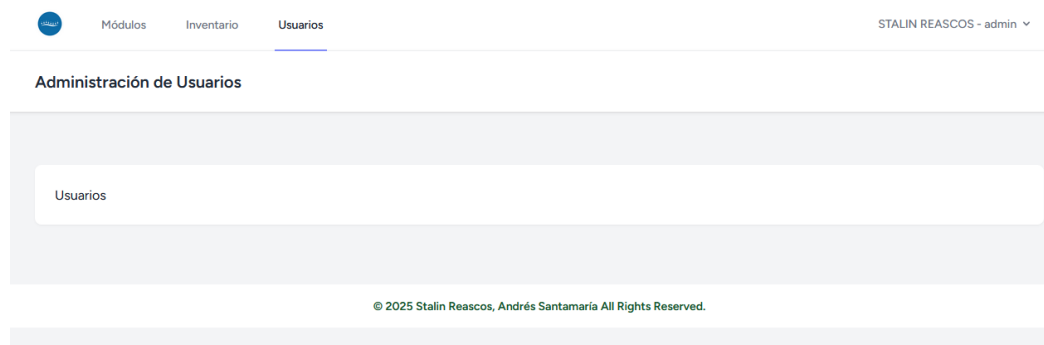


Ilustración 45. Vista Administración de Usuarios

Nota: esta vista únicamente estará visible para los usuarios del sistema que poseen el rol de “administrador”.

Perfil de Usuario

Cada usuario tiene un perfil creado con anterioridad por el administrador del sistema, este perfil se lo puede editar ya sea por el administrador, así como también, por el mismo usuario, dentro de este módulo podemos editar los siguientes datos (Nombre, Apellido y contraseña de ingreso) esta opción de perfil se encuentra ubicada en la parte superior derecha de nuestra barra de navegación donde se visualiza nuestro nombre de usuario.

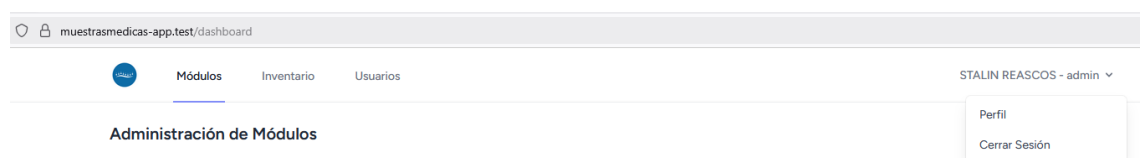


Ilustración 46. Módulo de edición de perfil.

Editar datos Usuario

Una vez dentro del perfil de usuario podremos editar los campos disponibles en la vista, estos son el Nombre, Apellido, una vez tengamos llenos estos campos únicamente damos clic en el botón “**Guardar**”, y automáticamente se actualizarán los cambios en el sistema.

Información del perfil
Actualiza tu información de perfil y correo electrónico

Nombre
STALIN

Apellido
REASCOS

Email
admin@admin.com

GUARDAR

Ilustración 47. Vista del Formulario de edición de Perfil.

Nota: el campo email no estará disponible para edición debido a que el correo es el usuario de inicio de sesión en el sistema.

Editar contraseña

Adicional a la información del perfil, también podemos editar o cambiar nuestra contraseña de inicio de sesión para lo cual debemos ingresar primero la contraseña actual con la que ingresamos al sistema, posterior a esto ingresamos la nueva contraseña y confirmamos la misma, una vez realizadas estas acciones damos clic en el botón de guardar para que se actualicen los datos.

Actualizar Password
Asegúrese de que su cuenta utilice una contraseña larga y aleatoria para mantener la seguridad.

Password Actual

Nuevo Password

Confirmar Password

GUARDAR

Ilustración 48. Formulario Actualización de Contraseña.

Eliminar cuenta de Usuario

El sistema también nos permite eliminar nuestra cuenta de usuario, se debe tomar en cuenta que esta opción únicamente la puede realizar los administradores del sistema, adicional el sistema valida que el usuario no haya realizado ninguna transacción en los diferentes módulos, en el caso de que si se realizaron transacciones el sistema no permitirá la eliminación de la cuenta.

Para eliminar la cuenta dentro de la edición de perfil únicamente damos clic en el botón “**eliminar cuenta**”, ubicado en la parte inferior de la vista de perfil.

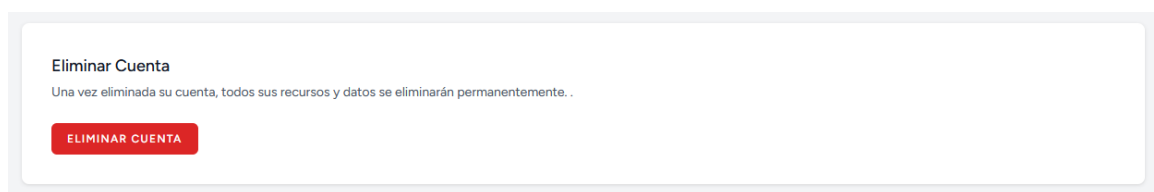


Ilustración 49. Eliminación de Cuenta de Usuario.

Nota: esta opción únicamente estará visible para los usuarios del sistema que poseen el rol de “administrador”.

LogOut de Usuario

Cerrar Sesión

Para cerrar sesión y salir del sistema únicamente nos dirigimos en la barra de navegación hacia el nombre de perfil, aquí nos mostrara la opción de cerrar sesión la cual al seleccionarla nos redirige hacia la vista inicial del sistema.

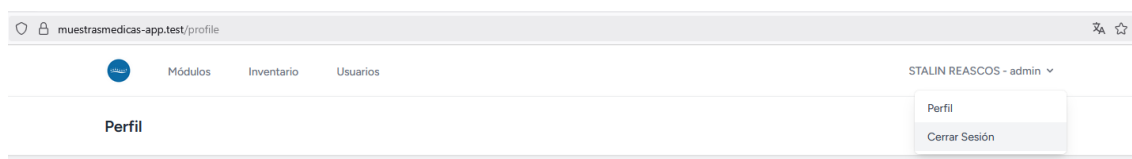


Ilustración 50. Cierre de Sesión Usuario

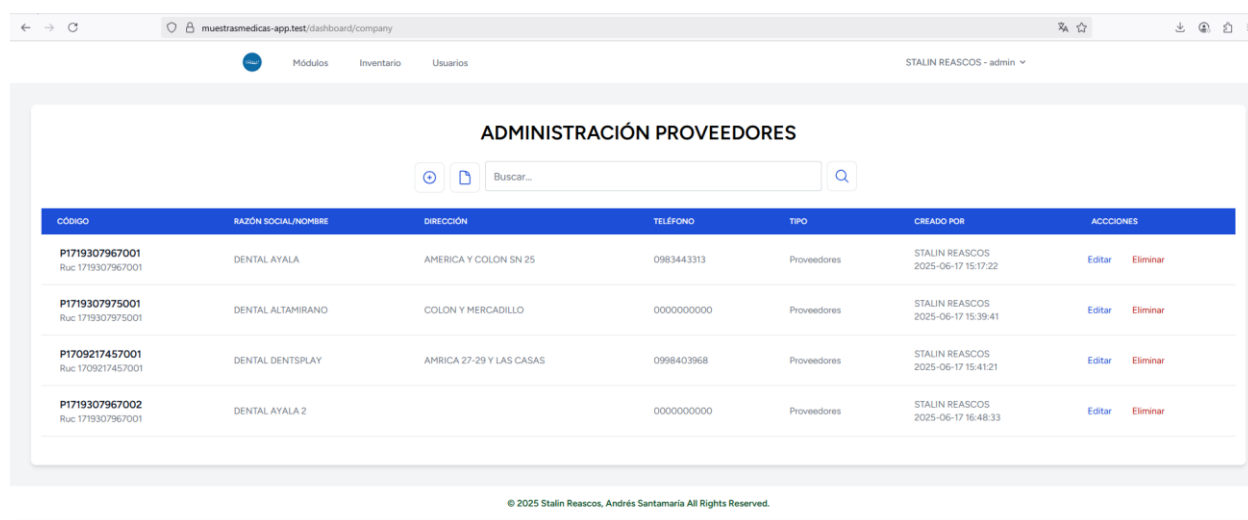
Nota: en los siguientes módulos se detallan pasos similares a los antes mencionados ya que todas las interfaces dentro de cada módulo poseen un mismo diseño.

Administración de Módulos

El sistema de Gestión de muestras médicas nos permite administrar los diferentes módulos que alimentan la base de datos, en este apartado revisaremos cada uno de estos con el fin de poder ingresar, modificar o eliminar de manera adecuada la información de cada módulo.

Administración de Compañías

Dentro de nuestro DASHBOARDS “*Módulos*”, nos dirigimos a la opción de “*Compañías*”, una vez seleccionada esta opción nos enviará a la vista principal que nos permitirá administrar los proveedores de nuestro sistema, a continuación, se presenta una vista general de este módulo.



CÓDIGO	RAZÓN SOCIAL/NOMBRE	DIRECCIÓN	TELÉFONO	TIPO	CREADO POR	ACCIONES
P1719307967001 Ruc 1719307967001	DENTAL AYALA	AMERICA Y COLON SN 25	0983443313	Proveedores	STALIN REASCOS 2025-06-17 15:17:22	Editar Eliminar
P1719307975001 Ruc 1719307975001	DENTAL ALTAMIRANO	COLON Y MERCADILLO	0000000000	Proveedores	STALIN REASCOS 2025-06-17 15:39:41	Editar Eliminar
P1709217457001 Ruc 1709217457001	DENTAL DENTSPLAY	AMERICA 27-29 Y LAS CASAS	0998403968	Proveedores	STALIN REASCOS 2025-06-17 15:41:21	Editar Eliminar
P1719307967002 Ruc 1719307967001	DENTAL AYALA 2		0000000000	Proveedores	STALIN REASCOS 2025-06-17 16:48:33	Editar Eliminar

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 51. Vista Principal Administración Proveedores.

Dentro de esta vista, podemos encontrar algunas opciones que nos permitirán realizar varias transacciones ya sean para crear, editar o eliminar los proveedores en el sistema, a continuación, se presenta una tabla que describe cada una de las funciones disponibles:

Opciones Administración Proveedores

OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo registro de Proveedor en la base de datos.
	Botón Listado: permite crear un listado con la información básica de los proveedores, se crea el reporte en formato PDF.
	Botón Filtro: permite filtrar los proveedores por algún criterio (Código, Ruc, Nombre, etc.).
Editar	Botón Editar: permite editar la información del Proveedor en la base de datos.
Eliminar	Botón Eliminar: permite eliminar el registro de Proveedor en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

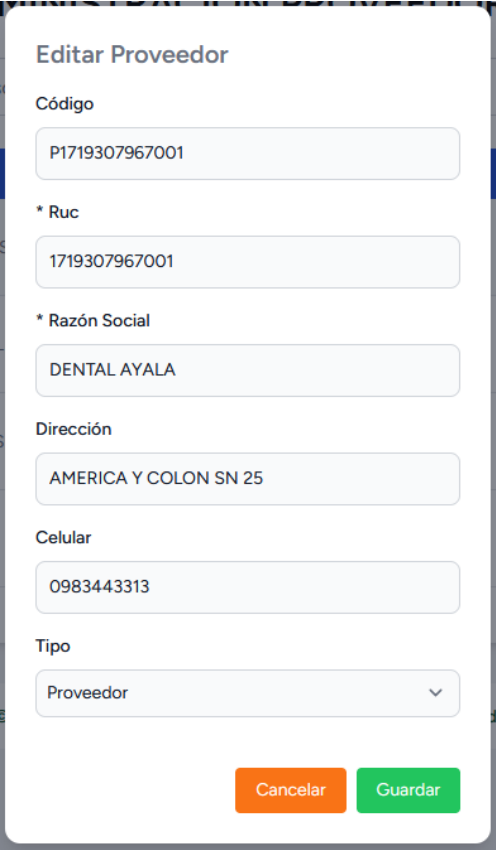
Tabla 5. Opciones Administración Proveedores.

Crear nuevo Proveedor

Como se ha visto en la tabla anterior para crear un nuevo proveedor únicamente debemos dar clic en el botón **“Nuevo”**, este nos abrirá un formulario que nos permitirá llenar con los datos del proveedor que queremos registrar, hay que tomar en cuenta que los campos del formulario que se encuentren marcados con * serán obligatorios.

Editar Proveedor

Para poder editar la información de un Proveedor únicamente nos dirigimos al listado detallado de proveedores, ubicamos al proveedor que queremos editar (Podemos utilizar la opción de filtro para ubicarlo más rápido), y en sus acciones damos clic en **“Editar”**, esto nos abre una vista con el formulario de edición, el cual contendrá en los campos la información del proveedor, dentro de este formulario el campo de código no se podrá editar ya que es un código único con el que se le registro en el sistema al proveedor, de igual manera los campos con * serán obligatorios.

El formulario, titulado "Editar Proveedor", contiene los siguientes campos: "Código" con el valor "P1719307967001"; "* Ruc" con el valor "1719307967001"; "* Razón Social" con el valor "DENTAL AYALA"; "Dirección" con el valor "AMERICA Y COLON SN 25"; "Celular" con el valor "0983443313"; y "Tipo" con un menú desplegable que muestra "Proveedor". En la parte inferior hay dos botones: "Cancelar" (naranja) y "Guardar" (verde).

Editar Proveedor	
Código	P1719307967001
* Ruc	1719307967001
* Razón Social	DENTAL AYALA
Dirección	AMERICA Y COLON SN 25
Celular	0983443313
Tipo	Proveedor
Cancelar Guardar	

Ilustración 54. Formulario edición Proveedor.

Para guardar en la base de datos del sistema, únicamente damos clic en el botón **“Guardar”**.

Eliminar Proveedor

Antes de pasar a eliminar un Proveedor del sistema, debemos tener en cuenta que esta acción únicamente la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar los proveedores que no tengan ninguna transacción en el sistema (Ingreso de Inventario).

Para poder realizar esta eliminación nos ubicamos en el proveedor que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido al proveedor en el caso de que existan más de 20 registros.

Ya ubicados en el proveedor escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el Código y Razón Social del proveedor que vamos a eliminar, damos clic en el botón **“Confirmar”**, y automáticamente se eliminara el registro de la base de datos.

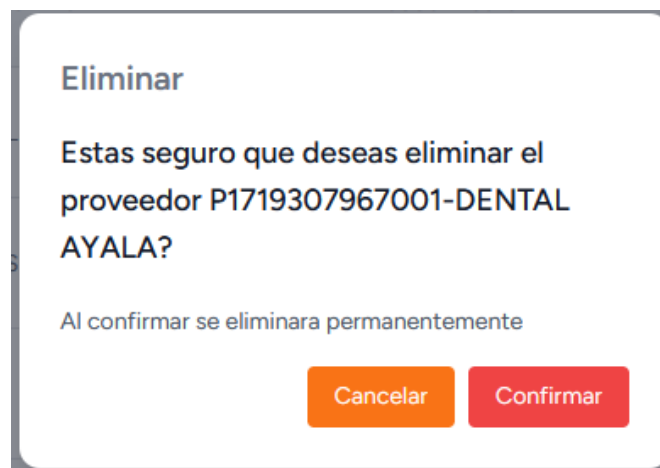


Ilustración 55. Eliminación de Proveedores.

Reporte básico Proveedores

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los Proveedores registrados.

Para poder generar este reporte damos clic en el icono de listado que se encuentra alado del botón **“Crear Proveedor”**, posterior a esta acción el sistema mostrará un modal en donde cargará el informe con el detalle de los Proveedores que se han registrado en la base de datos.



Vista Previa PDF

1 de 1

Tamaño automático

C.G.M.M.
CONTROL Y GESTIÓN DE
MUESTRAS MÉDICAS

FECHA IMPRESIÓN	HORA IMPRESIÓN	USUARIO
2025-07-08	16:27:49	STALIN REASCOS

LISTADO DE PROVEEDORES

Código	Ruc	Razon Soocial	Dirección	Teléfono	Tipo
P1709217457001	1709217457001	DENTAL DENTSPLAY	AMRICA 27-29 Y LAS CASAS	0998403968	Proveedor
P1719307967001	1719307967001	DENTAL AYALA	AMERICA Y COLON SN 25	0983443313	Proveedor
P1719307967002	1719307967001	DENTAL AYALA 2			Proveedor
P1719307975001	1719307975001	DENTAL ALTAMIRANO	COLON Y MERCADILLO		Proveedor

Ilustración 56. Reporte Listado Proveedores.

Administración de Visitadores

Dentro de nuestro DASHBOARDS **“Módulos”**, nos dirigimos a la opción de **“Visitadores”**, una vez seleccionada esta opción nos enviará a la vista principal que nos

permitirá administrar los visitantes o representantes de ventas de nuestro sistema, a continuación, se presenta una vista general de este módulo.

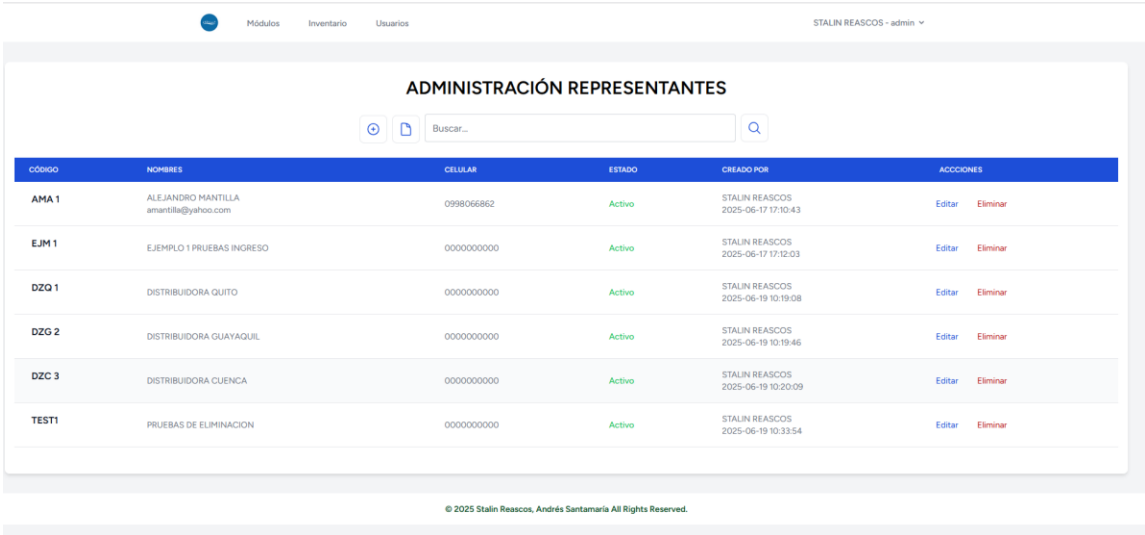


Ilustración 57. Vista Principal Administración Visitadores.

Dentro de esta vista, podemos encontrar algunas opciones que nos permitirán realizar varias transacciones ya sean para crear, editar o eliminar los Representantes en el sistema, a continuación, se presenta una tabla que describe cada una de las funciones disponibles:

Opciones Administración Representantes

OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo registro de visitador o representante de ventas en la base de datos.
	Botón Listado: permite crear un listado con la información básica de los representantes de venta, se crea el reporte en formato PDF.

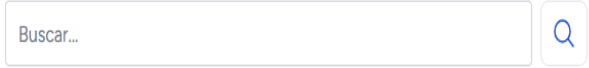
	Botón Filtro: permite filtrar los representantes por algún criterio (Código, Nombre, Estado, etc.).
	Botón Editar: permite editar la información del Representante en la base de datos.
	Botón Eliminar: permite eliminar el registro de un Representante en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 6. Opciones Administración Representantes.

Crear nuevo Representante

Como se ha visto en la tabla anterior para crear un nuevo representante únicamente debemos dar clic en el botón **“Nuevo”**, este nos abrirá un formulario que nos permitirá llenar con los datos del representante que queremos registrar, hay que tomar en cuenta que los campos del formulario que se encuentren marcados con * serán obligatorios.

Nuevo Representante

* Código

Example VEN 1

* Nombres y Apellidos

Nombres y apellidos del representante

Email

example@example.com

Celular

Numero celular

Cancelar

Guardar

Ilustración 58. Formulario Registro Representante.

Una vez con el formulario lleno únicamente damos clic en **“Guardar”**, y el sistema almacenara esta información en su base de datos, este paso de creación es muy sencillo y una vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de representantes en donde ya se encontrará detallado en el listado el representante creado.

CÓDIGO	NOMBRES	CELULAR	ESTADO	CREADO POR	ACCIONES	
AMA 1	ALEJANDRO MANTILLA amantilla@yahoo.com	0998066862	Activo	STALIN REASCOS 2025-06-17 17:10:43	Editar	Eliminar
EJM 1	EJEMPLO 1 PRUEBAS INGRESO	0000000000	Activo	STALIN REASCOS 2025-06-17 17:12:03	Editar	Eliminar
DZQ 1	DISTRIBUIDORA QUITO	0000000000	Activo	STALIN REASCOS 2025-06-19 10:19:08	Editar	Eliminar
DZG 2	DISTRIBUIDORA GUAYAQUIL	0000000000	Activo	STALIN REASCOS 2025-06-19 10:19:46	Editar	Eliminar
DZC 3	DISTRIBUIDORA CUENCA	0000000000	Activo	STALIN REASCOS 2025-06-19 10:20:09	Editar	Eliminar
TEST1	PRUEBAS DE ELIMINACION	0000000000	Activo	STALIN REASCOS 2025-06-19 10:33:54	Editar	Eliminar

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 59. Detalle de Representantes Registrados.

Editar Representante

Para poder editar la información de un Representante de ventas únicamente nos dirigimos al listado detallado de representantes, ubicamos el registro que queremos editar

(Podemos utilizar la opción de filtro para ubicarlo más rápido), y en sus acciones damos clic en **“Editar”**, esto nos abre una vista con el formulario de edición, el cual contendrá en los campos la información del representante, dentro de este formulario el campo de código no se podrá editar ya que es un código único con el que se le registro en el sistema al representante, de igual manera los campos con * serán obligatorios.

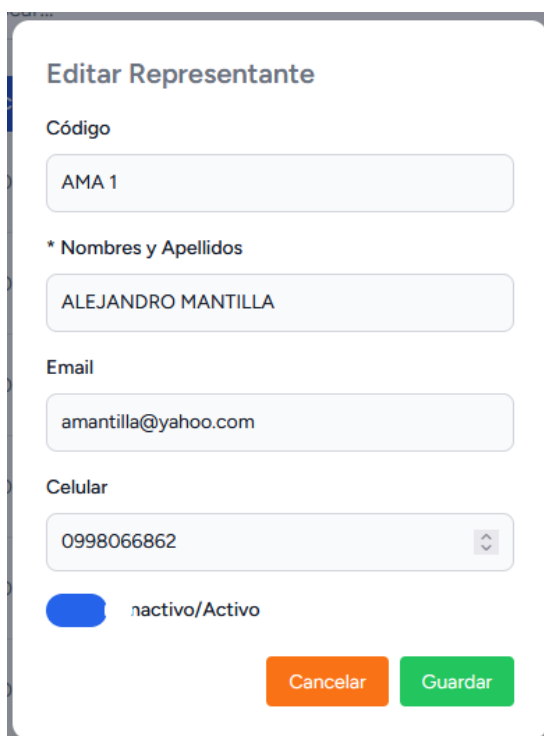
El formulario se titula "Editar Representante". Contiene los siguientes campos: "Código" con el valor "AMA 1"; "* Nombres y Apellidos" con el valor "ALEJANDRO MANTILLA"; "Email" con el valor "amantilla@yahoo.com"; "Celular" con el valor "0998066862" y un icono de teléfono a la derecha; un interruptor de estado actualmente en "Inactivo" (botón azul) con la etiqueta "Inactivo/Activo"; y dos botones de acción al final: "Cancelar" (naranja) y "Guardar" (verde).

Ilustración 60. Formulario edición Representante.

Para guardar en la base de datos del sistema, únicamente damos clic en el botón **“Guardar”**.

Eliminar Representante

Antes de pasar a eliminar un Representante del sistema, debemos tener en cuenta que esta acción únicamente la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar los Representantes que no tengan ninguna transacción en el sistema (Egresos de Inventario).

Para poder realizar esta eliminación nos ubicamos en el registro que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido al representante en el caso de que existan más de 20 registros.

Ya ubicados en el representante escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el Código y Nombres del Representante que vamos a eliminar, damos clic en el botón **“Confirmar”**, y automáticamente se eliminara el registro de la base de datos.

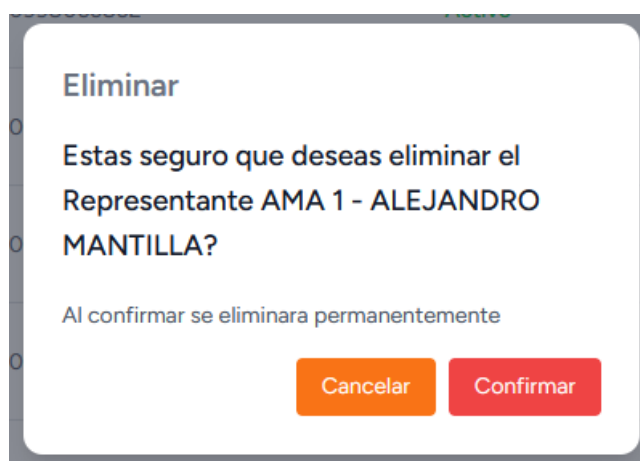


Ilustración 61. Eliminación de Representantes.

Reporte básico Representantes

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los Representantes registrados.

Para poder generar este reporte damos clic en el icono de listado que se encuentra alado del botón **“Crear Representante”**, posterior a esta acción el sistema mostrará un modal en donde cargará el informe con el detalle de los Representantes que se han registrado en la base de datos.



FECHA IMPRESIÓN	HORA IMPRESIÓN	USUARIO
2025-07-08	16:58:50	STALIN REASCOS

LISTADO DE VISITADORES CON SUS ZONAS

Codigo	Nombres	Email	Celular	Estado	Zonas
AMA 1	ALEJANDRO MANTILLA		0998066862	ACTIVO	1001 ZONA CENTRO QUITO 1002 ZONA SUR QUITO 1003 ZONA NORTE QUITO
DZC 3	DISTRIBUIDORA CUENCA			ACTIVO	
DZG 2	DISTRIBUIDORA GUAYAQUIL			ACTIVO	2101 ZONA CENTRO GUAYAS 2102 ZONA NORTE GUAYAS 2103 ZONA SUR GUAYAS
DZQ 1	DISTRIBUIDORA QUITO			ACTIVO	1004 ZONA VALLES
EJM 1	EJEMPLO 1 PRUEBAS INGRESO			ACTIVO	
TEST1	PRUEBAS DE ELIMINACION			ACTIVO	

Ilustración 62. Reporte Listado Representantes.

Administración de Zonas

Dentro de nuestro DASHBOARDS “**Módulos**”, nos dirigimos a la opción de “**Zonas**”, una vez seleccionada esta opción nos enviará a la vista principal que nos permitirá administrar las Zonas en nuestro sistema, a continuación, se presenta una vista general de este módulo.

ADMINISTRACIÓN ZONAS				
Buscar...				
CÓDIGO	DESCRIPCIÓN	REPRESENTANTE	CREADOR POR:	ACCIONES
1001	ZONA CENTRO QUITO	ALEJANDRO MANTILLA	STALIN REASCOS 2025-06-24 11:22:00	Editar Eliminar
1002	ZONA SUR QUITO	ALEJANDRO MANTILLA	STALIN REASCOS 2025-06-30 18:24:37	Editar Eliminar
1003	ZONA NORTE QUITO	ALEJANDRO MANTILLA	STALIN REASCOS 2025-06-30 18:24:46	Editar Eliminar
1004	ZONA VALLES	DISTRIBUIDORA QUITO	STALIN REASCOS 2025-06-30 18:25:01	Editar Eliminar
2101	ZONA CENTRO GUAYAS	DISTRIBUIDORA GUAYAQUIL	STALIN REASCOS 2025-06-30 18:25:24	Editar Eliminar
2102	ZONA NORTE GUAYAS	DISTRIBUIDORA GUAYAQUIL	STALIN REASCOS 2025-06-30 18:25:38	Editar Eliminar
2103	ZONA SUR GUAYAS	DISTRIBUIDORA GUAYAQUIL	STALIN REASCOS 2025-06-30 18:25:55	Editar Eliminar

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 63. Vista Principal Administración Zonas.

Dentro de esta vista, podemos encontrar algunas opciones que nos permitirán realizar varias transacciones ya sean para crear, editar o eliminar las Zonas en el sistema, a continuación, se presenta una tabla que describe cada una de las funciones disponibles:

Opciones Administración Zonas

OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo registro de Zonas en la base de datos.
	Botón Listado: permite crear un listado con la información básica de las Zonas con sus representantes de venta, reporte en formato PDF.
Buscar...	Botón Filtro: permite filtrar las Zonas por algún criterio (Código, Descripción, etc.).

Editar	Botón Editar: permite editar la información de la Zona en la base de datos.
Eliminar	Botón Eliminar: permite eliminar el registro de una Zona en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 7. Opciones Administración Zonas.

Crear nueva Zona

Como se ha visto en la tabla anterior para crear una nueva Zona únicamente debemos dar clic en el botón **“Nuevo”**, este nos abrirá un formulario que nos permitirá llenar con los datos de la Zona que queremos registrar, hay que tomar en cuenta que los campos del formulario que se encuentren marcados con * serán obligatorios.

Nueva Zona

* Código

Ejemplo 3004

* Descripción

Describe el nombre de la zona

* Seleccione Representante

AMA 1 - ALEJANDRO MANTILLA

Cancelar

Guardar

Ilustración 64. Formulario Registro Zona.

Una vez con el formulario lleno únicamente damos clic en **“Guardar”**, y el sistema almacenara esta información en su base de datos, este paso de creación es muy sencillo y una

vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de Zonas en donde ya se encontrará detallado en el listado el representante creado.

ADMINISTRACIÓN ZONAS				
BUSCAR...				
CÓDIGO	DESCRIPCIÓN	REPRESENTANTE	CREADOR POR:	ACCIONES
1001	ZONA CENTRO QUITO	ALEJANDRO MANTILLA	STALIN REASCOS 2025-06-24 11:22:00	Editar Eliminar
1002	ZONA SUR QUITO	ALEJANDRO MANTILLA	STALIN REASCOS 2025-06-30 18:24:37	Editar Eliminar
1003	ZONA NORTE QUITO	ALEJANDRO MANTILLA	STALIN REASCOS 2025-06-30 18:24:46	Editar Eliminar
1004	ZONA VALLES	DISTRIBUIDORA QUITO	STALIN REASCOS 2025-06-30 18:25:01	Editar Eliminar
2101	ZONA CENTRO GUAYAS	DISTRIBUIDORA GUAYAQUIL	STALIN REASCOS 2025-06-30 18:25:24	Editar Eliminar
2102	ZONA NORTE GUAYAS	DISTRIBUIDORA GUAYAQUIL	STALIN REASCOS 2025-06-30 18:25:38	Editar Eliminar
2103	ZONA SUR GUAYAS	DISTRIBUIDORA GUAYAQUIL	STALIN REASCOS 2025-06-30 18:25:55	Editar Eliminar

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 65. Detalle de Zonas Registradas.

Editar Zona

Para poder editar la información de una Zona únicamente nos dirigimos al listado detallado de zonas, ubicamos el registro que queremos editar (Podemos utilizar la opción de filtro para ubicarlo más rápido), y en sus acciones damos clic en **“Editar”**, esto nos abre una vista con el formulario de edición, el cual contendrá en los campos la información de la zona, dentro de este formulario el campo de código no se podrá editar ya que es un código único con el que se le registro en el sistema a la Zona, de igual manera los campos con * serán obligatorios.

El formulario 'Editar Zona' tiene un título 'Editar Zona' en la parte superior. Debajo, hay un campo 'Código' con el valor '1001' y un icono de flechas hacia arriba y hacia abajo. A continuación, hay un campo '* Descripción' con el valor 'ZONA CENTRO QUITO'. Luego, hay un campo 'Seleccione Representante' con el valor 'AMA 1 - ALEJANDRO MANTILLA' y un icono de flecha hacia abajo. En la parte inferior derecha, hay dos botones: 'Cancelar' (naranja) y 'Guardar' (verde).

Editar Zona

Código

1001

* Descripción

ZONA CENTRO QUITO

Seleccione Representante

AMA 1 - ALEJANDRO MANTILLA

Cancelar Guardar

Ilustración 66. Formulario edición Zona.

Para guardar en la base de datos del sistema, únicamente damos clic en el botón **“Guardar”**.

Eliminar Zona

Antes de pasar a eliminar una Zona del sistema, debemos tener en cuenta que esta acción únicamente la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar las Zonas que no tengan ninguna transacción en el sistema.

Para poder realizar esta eliminación nos ubicamos en el registro que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido a la Zona en el caso de que existan más de 20 registros.

Ya ubicados en la Zona escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el Código y Descripción de la zona que vamos a eliminar, damos clic en el botón **“Confirmar”**, y automáticamente se eliminara el registro de la base de datos.

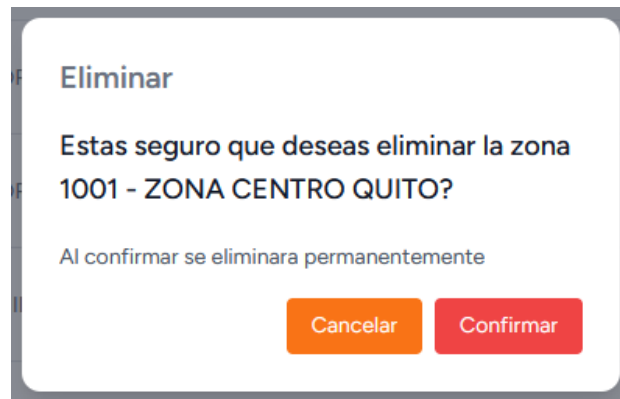


Ilustración 67. Eliminación Zona

Reporte básico Zonas

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de las Zonas registradas.

Para poder generar este reporte damos clic en el icono de listado que se encuentra alado del botón “*Crear Zona*”, posterior a esta acción el sistema mostrará un modal en donde cargará el informe con el detalle de las Zonas que se han registrado en la base de datos.



Ilustración 68. Reporte Listado Zonas.

Administración de Productos

Dentro de nuestro DASHBOARDS “*Módulos*”, nos dirigimos a la opción de “*Productos*”, una vez seleccionada esta opción nos enviará a la vista principal que nos permitirá administrar los Productos en nuestro sistema, a continuación, se presenta una vista general de este módulo.

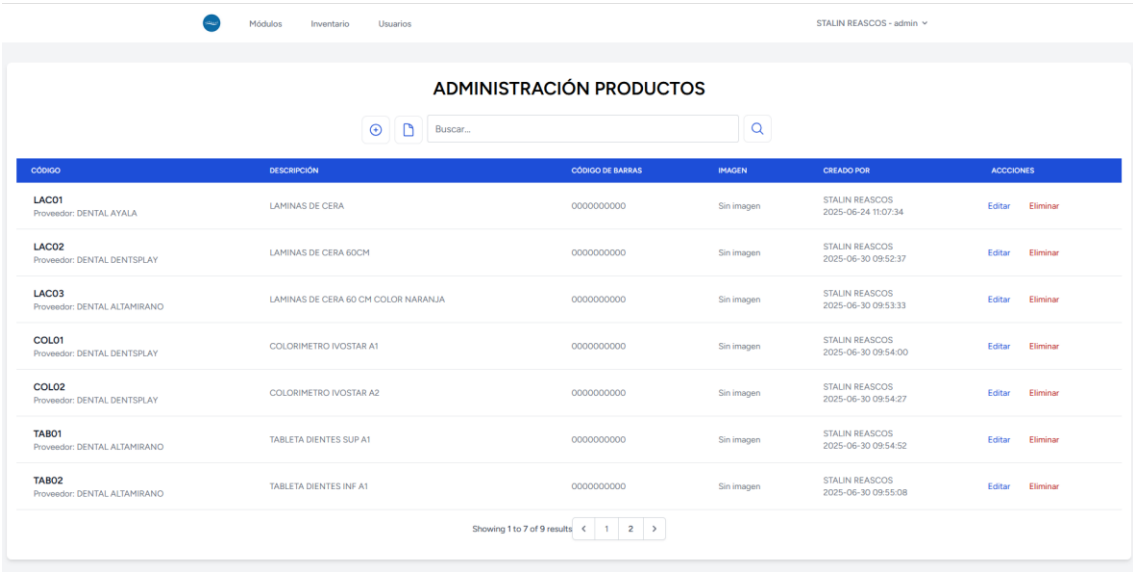


Ilustración 69. Vista Principal Administración Productos.

Dentro de esta vista, podemos encontrar algunas opciones que nos permitirán realizar varias transacciones ya sean para crear, editar o eliminar los Productos en el sistema, a continuación, se presenta una tabla que describe cada una de las funciones disponibles:

Opciones Administración Productos

OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo registro de Producto en la base de datos.

	Botón Listado: permite crear un listado con la información básica de los Productos con sus Proveedores, reporte en formato PDF.
Buscar... 	Botón Filtro: permite filtrar los Productos por algún criterio (Código, Descripción, Proveedor, etc.).
Editar	Botón Editar: permite editar la información del Producto en la base de datos.
Eliminar	Botón Eliminar: permite eliminar el registro de un Producto en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 8. Opciones Administración Productos

Crear nuevo Producto

Como se ha visto en la tabla anterior para crear un nuevo Producto únicamente debemos dar clic en el botón “**Nuevo**”, este nos abrirá un formulario que nos permitirá llenar con los datos del Producto que queremos registrar, hay que tomar en cuenta que los campos del formulario que se encuentren marcados con * serán obligatorios.

Nuevo Producto

* Código

Example COD01

* Descripción

Láminas de cera

Código de Barras

Example 010321654879

* Proveedor

P1709217457001 - DENTAL DENTSPRAY

Imagen

Examinar...

No se ha sele...ngún archivo.

Cancelar

Guardar

Ilustración 70. Formulario Registro Producto.

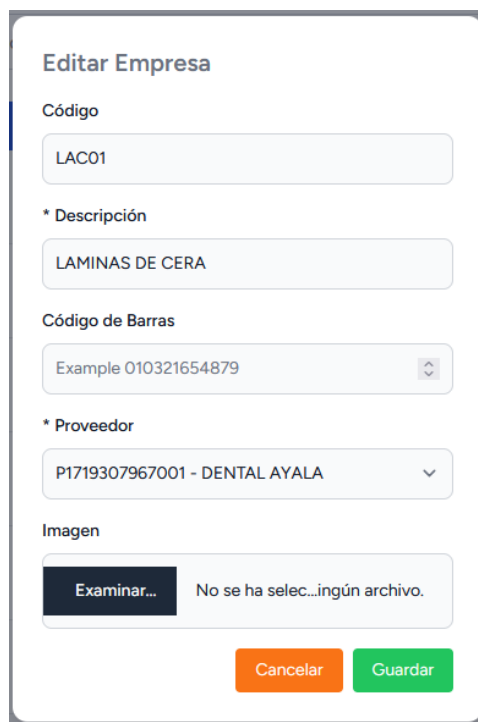
Una vez con el formulario lleno únicamente damos clic en **“Guardar”**, y el sistema almacenara esta información en su base de datos, este paso de creación es muy sencillo y una vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de Productos en donde ya se encontrará detallado en el listado el producto creado.

ADMINISTRACIÓN PRODUCTOS					
CÓDIGO	DESCRIPCIÓN	CÓDIGO DE BARRAS	IMAGEN	CREADO POR	ACCIONES
LAC01 Proveedor: DENTAL AYALA	LAMINAS DE CERA	0000000000	Sin imagen	STALIN REASCOS 2025-06-24 11:07:34	Editar Eliminar
LAC02 Proveedor: DENTAL DENTSPRAY	LAMINAS DE CERA 60CM	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:52:37	Editar Eliminar
LAC03 Proveedor: DENTAL ALTAMIRANO	LAMINAS DE CERA 60 CM COLOR NARANJA	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:53:33	Editar Eliminar
COL01 Proveedor: DENTAL DENTSPRAY	COLORIMETRO IVOSTAR A1	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:54:00	Editar Eliminar
COL02 Proveedor: DENTAL DENTSPRAY	COLORIMETRO IVOSTAR A2	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:54:27	Editar Eliminar
TAB01 Proveedor: DENTAL ALTAMIRANO	TABLETA DIENTES SUP A1	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:54:52	Editar Eliminar
TAB02 Proveedor: DENTAL ALTAMIRANO	TABLETA DIENTES INF A1	0000000000	Sin imagen	STALIN REASCOS 2025-06-30 09:55:08	Editar Eliminar

Ilustración 71. Detalle de Productos Registrados.

Editar Producto

Para poder editar la información de un Producto únicamente nos dirigimos al listado detallado de productos, ubicamos el registro que queremos editar (Podemos utilizar la opción de filtro para ubicarlo más rápido), y en sus acciones damos clic en **“Editar”**, esto nos abre una vista con el formulario de edición, el cual contendrá en los campos la información del producto, dentro de este formulario el campo de código no se podrá editar ya que es un código único con el que se le registro en el sistema al Producto, de igual manera los campos con * serán obligatorios.

El formulario, titulado "Editar Empresa", contiene los siguientes campos: "Código" con el valor "LAC01"; "* Descripción" con el valor "LAMINAS DE CERA"; "Código de Barras" con el valor "Example 010321654879"; "* Proveedor" con el valor "P1719307967001 - DENTAL AYALA"; y "Imagen" con un botón "Examinar..." y el texto "No se ha seleccionado ningún archivo.". En la parte inferior hay dos botones: "Cancelar" (naranja) y "Guardar" (verde).

Editar Empresa	
Código	LAC01
* Descripción	LAMINAS DE CERA
Código de Barras	Example 010321654879
* Proveedor	P1719307967001 - DENTAL AYALA
Imagen	Examinar... No se ha seleccionado ningún archivo.
Cancelar Guardar	

Ilustración 72. Formulario edición Producto.

Para guardar en la base de datos del sistema, únicamente damos clic en el botón **“Guardar”**.

Eliminar Producto

Antes de pasar a eliminar un Producto del sistema, debemos tener en cuenta que esta acción únicamente la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar los Productos que no tengan ninguna transacción en el sistema (Ingresos o Egresos de Inventario).

Para poder realizar esta eliminación nos ubicamos en el registro que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido al Producto en el caso de que existan más de 20 registros.

Ya ubicados en el Producto escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el Código y Descripción del producto que vamos a eliminar, damos clic en el botón **“Confirmar”**, y automáticamente se eliminara el registro de la base de datos.

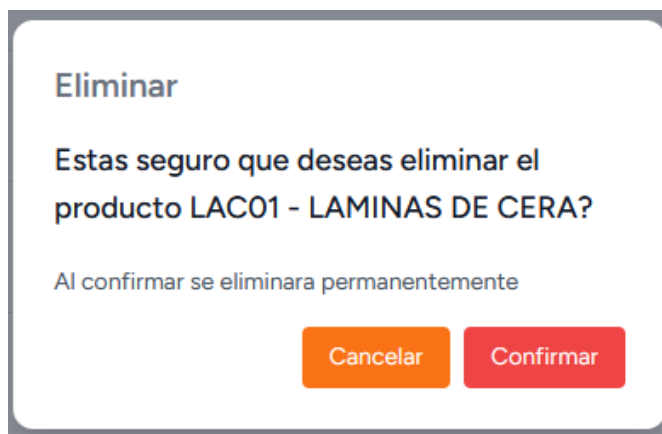


Ilustración 73. Eliminación Producto.

Reporte básico Productos

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los Productos registrados.

Para poder generar este reporte damos clic en el icono de listado que se encuentra alado del botón “*Crear Producto*”, posterior a esta acción el sistema mostrará un modal en donde cargará el informe con el detalle de los Productos que se han registrado en la base de datos.

Vista Previa PDF

Código	Proveedor	Descripción	Código de barras	Imagen
COL01	DENTAL DENTSPLAY	COLORIMETRO IVOSTAR A1	0000000000	Sin Imagen
COL02	DENTAL DENTSPLAY	COLORIMETRO IVOSTAR A2	0000000000	Sin Imagen
LAC01	DENTAL AYALA	LAMINAS DE CERA	0000000000	Sin Imagen
LAC02	DENTAL DENTSPLAY	LAMINAS DE CERA 60CM	0000000000	Sin Imagen
LAC03	DENTAL ALTAMIRANO	LAMINAS DE CERA 60 CM COLOR NARANJA	0000000000	Sin Imagen
TAB01	DENTAL ALTAMIRANO	TABLETA DIENTES SUP A1	0000000000	Sin Imagen
TAB02	DENTAL ALTAMIRANO	TABLETA DIENTES INF A1	0000000000	Sin Imagen
TAB03	DENTAL AYALA	TABLETA DIENTES SUP A2	0000000000	Sin Imagen
TAB04	DENTAL ALTAMIRANO	TABLETA DIENTES INF A2	0000000000	Sin Imagen

Ilustración 74. Reporte Listado Zonas.

Administración de Inventario

El sistema de Gestión de muestras médicas nos permite administrar los diferentes módulos que alimentan el inventario en el sistema, en este apartado revisaremos cada uno de estos con el fin de poder realizar ingresos, egresos, administración de lotes y verificación del Kardex de los productos registrados.

Administración de Lotes

Dentro de nuestro DASHBOARDS “*Inventario*”, nos dirigimos a la opción de “*Lotes*”, una vez seleccionada esta opción nos enviará a la vista principal que nos permitirá administrar los Lotes registrados al momento de realizar un ingreso de inventario en nuestro sistema, a continuación, se presenta una vista general de este módulo.

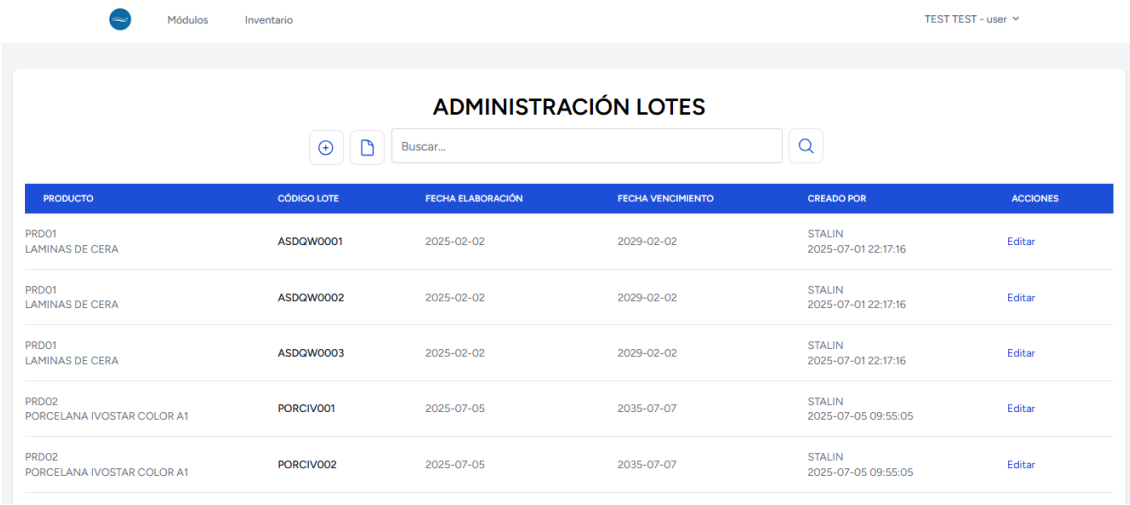


Ilustración 75. Vista Principal Administración Lotes.

Dentro de esta vista, podemos encontrar algunas opciones que nos permitirán realizar varias transacciones ya sean para crear, editar o eliminar los Lotes de los productos en el sistema, a continuación, se presenta una tabla que describe cada una de las funciones disponibles:

Opciones Administración Lotes

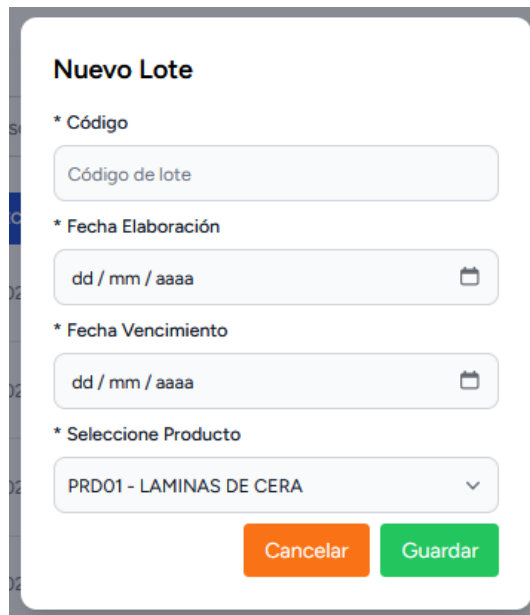
OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo Lote asignarlo a un producto en la base de datos.

	Botón Listado: permite crear un listado con la información básica de los Lotes registrados en el sistema, se crea el reporte en formato PDF.
Buscar... 	Botón Filtro: permite filtrar los lotes por algún criterio (Código, Producto, etc.).
Editar	Botón Editar: permite editar la información del Lote en la base de datos.
Eliminar	Botón Eliminar: permite eliminar el registro de Lote en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 9. Opciones Administración Lotes.

Crear nuevo Lote

Como se ha visto en la tabla anterior para crear un nuevo lote únicamente debemos dar clic en el botón “**Nuevo**”, este nos abrirá un formulario que nos permitirá llenar con los datos del lote que queremos registrar, hay que tomar en cuenta que los campos del formulario que se encuentren marcados con * serán obligatorios, así como también, deberá estar previamente creado el producto al cual le vamos asignar el lote que se está creando en el sistema.



Nuevo Lote

* Código

Código de lote

* Fecha Elaboración

dd / mm / aaaa

* Fecha Vencimiento

dd / mm / aaaa

* Seleccione Producto

PRD01 - LAMINAS DE CERA

Cancelar Guardar

Ilustración 76. Formulario Creación Lote.

Nota: No se podrá crear un lote si un producto al cual le vamos asignar, no está previamente creado.

Una vez con el formulario lleno únicamente damos clic en **“Guardar”**, y el sistema almacenara esta información en su base de datos, este paso de creación es muy sencillo y una vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de Lotes en donde ya se encontrará detallado en el listado el lote creado.

ADMINISTRACIÓN LOTES					
Buscar...					
PRODUCTO	CÓDIGO LOTE	FECHA ELABORACIÓN	FECHA VENCIMIENTO	CREADO POR	ACCIONES
PRD01 LAMINAS DE CERA	ASDQW0001	2025-02-02	2029-02-02	STALIN 2025-07-01 22:17:16	Editar Eliminar
PRD01 LAMINAS DE CERA	ASDQW0002	2025-02-02	2029-02-02	STALIN 2025-07-01 22:17:16	Editar Eliminar
PRD01 LAMINAS DE CERA	ASDQW0003	2025-02-02	2029-02-02	STALIN 2025-07-01 22:17:16	Editar Eliminar
PRD02 PORCELANA IVOSTAR COLOR A1	PORCIV001	2025-07-05	2035-07-07	STALIN 2025-07-05 09:55:05	Editar Eliminar
PRD02 PORCELANA IVOSTAR COLOR A1	PORCIV002	2025-07-05	2035-07-07	STALIN 2025-07-05 09:55:05	Editar Eliminar

Ilustración 77. Detalle de Lotes Registrados.

Editar Lote

Para poder editar la información de un Lote únicamente nos dirigimos al listado detallado de lotes, ubicamos al lote que queremos editar (Podemos utilizar la opción de filtro para ubicarlo más rápido), y en sus acciones damos clic en **“Editar”**, esto nos abre una vista con el formulario de edición, el cual contendrá en los campos la información del lote, dentro de este formulario el campo de código y producto no se podrán editar ya que el código es único con el que se le registró en el sistema al lote, con respecto al producto no se lo podrá cambiar por seguridad en las transacciones realizadas previamente (Ingresos o Egresos de Inventario), de igual manera los campos con * serán obligatorios.

Modificar Lote

Código

ASDQW0001

* Fecha Elaboración

02 / 02 / 2025

* Fecha Vencimiento

02 / 02 / 2029

Producto

PRD01 - LAMINAS DE CERA

Cancelar Guardar

Ilustración 78. Formulario edición Lote.

Para guardar en la base de datos del sistema, únicamente damos clic en el botón **“Guardar”**.

Eliminar Lote

Antes de pasar a eliminar un Lote del sistema, debemos tener en cuenta que esta acción solo la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar los Lotes que no tengan ninguna transacción en el sistema (Ingreso o Egresos de Inventario).

Para poder realizar esta eliminación nos ubicamos en el lote que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido al lote en el caso de que existan más de 20 registros.

Ya ubicados en el lote escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el Código del lote que vamos a eliminar, damos clic en el botón **“Confirmar”**, y automáticamente se eliminara el registro de la base de datos.

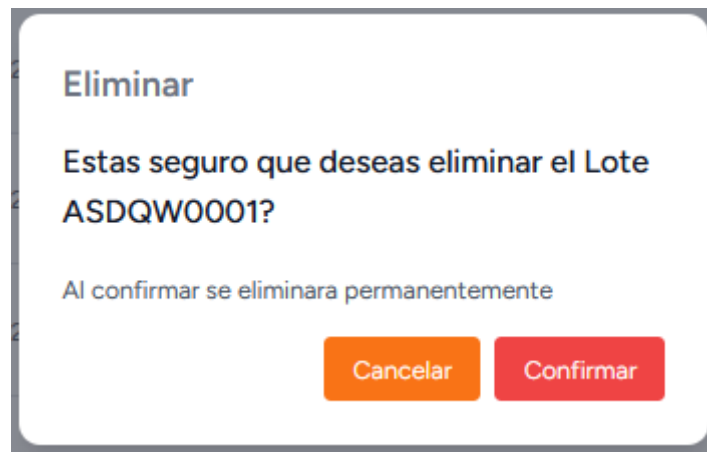


Ilustración 79. Eliminación de Lote.

Reporte básico de Lotes

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los lotes registrados.

Para poder generar este reporte damos clic en el icono de listado que se encuentra alado del botón **“Crear Lote”**, posterior a esta acción el sistema mostrará un modal en donde cargará el informe con el detalle de los Lotes que se han registrado en la base de datos.

Vista Previa PDF

1 de 1 100%

C.G.M.M
CONTROL Y GESTIÓN DE MUESTRAS MÉDICAS

FECHA IMPRESIÓN	HORA IMPRESIÓN	USUARIO
2025-07-08	20:55:08	STALIN REASCOS

LISTADO DE LOTES REGISTRADOS

CÓDIGO LOTE	PRODUCTO	PROVEEDOR	F. ELAB.	F. VENC.
ASDQW0001	LAMINAS DE CERA	VANESSA ERAZO	2025-02-02	2029-02-02
ASDQW0002	LAMINAS DE CERA	VANESSA ERAZO	2025-02-02	2029-02-02
ASDQW0003	LAMINAS DE CERA	VANESSA ERAZO	2025-02-02	2029-02-02
PORCIV001	PORCELANA IVOSTAR COLOR A1	DISTRIBUIDORA DENTAL AYALA	2025-07-05	2035-07-07
PORCIV002	PORCELANA IVOSTAR COLOR A1	DISTRIBUIDORA DENTAL AYALA	2025-07-05	2035-07-07

Ilustración 80. Reporte Listado Lotes.

Usuarios del sistema

El sistema de Gestión de muestras médicas nos permite administrar los diferentes módulos que alimentan la información de usuarios en el sistema, en este apartado revisaremos cada uno de estos con el fin de poder realizar registro, modificación y eliminación de usuarios, así como también, el cambio o actualización de la contraseña.

Administración de Usuarios

Dentro de nuestro DASHBOARDS **“Usuarios”**, nos dirigimos a la opción de **“Administrar”**, una vez seleccionada esta opción nos enviará a la vista principal que nos permitirá administrar los usuarios registrados en nuestro sistema, a continuación, se presenta una vista general de este módulo.

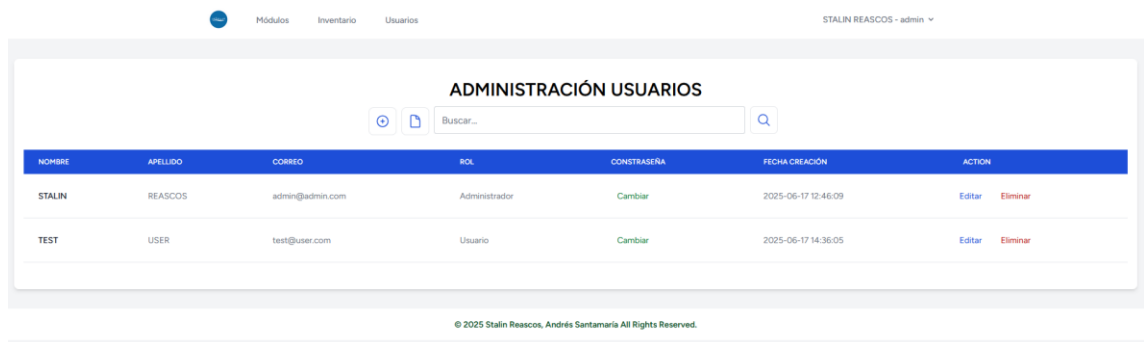


Ilustración 81. Vista Principal Administración Usuarios.

Dentro de esta vista, podemos encontrar algunas opciones que nos permitirán realizar varias transacciones ya sean para crear, editar, eliminar o actualizar la contraseña de los usuarios en el sistema, a continuación, se presenta una tabla que describe cada una de las funciones disponibles:

Opciones Administración Usuarios

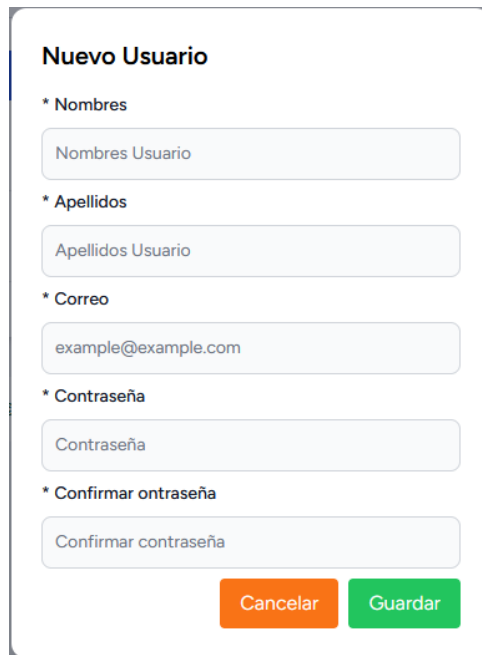
OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo Lote asignarlo a un producto en la base de datos.
	Botón Listado: permite crear un listado con la información básica de los Lotes registrados en el sistema, se crea el reporte en formato PDF.
Buscar...	Botón Filtro: permite filtrar los lotes por algún criterio (Código, Producto, etc.).
Editar	Botón Editar: permite editar la información del Lote en la base de datos.

Cambiar	Botón Cambiar: permite cambiar o actualizar la contraseña del usuario seleccionado. (Únicamente puede ser visualizado por el administrador del sistema)
Eliminar	Botón Eliminar: permite eliminar el registro de Lote en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 10. Opciones Administración Usuarios.

Crear nuevo Usuario

Como se ha visto en la tabla anterior para crear un nuevo usuario únicamente debemos dar clic en el botón “**Nuevo**”, este nos abrirá un formulario que nos permitirá llenar con los datos del usuario que queremos registrar, hay que tomar en cuenta que los campos del formulario que se encuentren marcados con * serán obligatorios.



Nuevo Usuario

* Nombres

Nombres Usuario

* Apellidos

Apellidos Usuario

* Correo

example@example.com

* Contraseña

Contraseña

* Confirmar contraseña

Confirmar contraseña

Cancelar Guardar

Ilustración 82. Formulario Creación Usuario.

Una vez con el formulario lleno únicamente damos clic en **“Guardar”**, y el sistema almacenara esta información en su base de datos, este paso de creación es muy sencillo y una vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de Usuarios en donde ya se encontrará detallado en el listado el usuario creado.



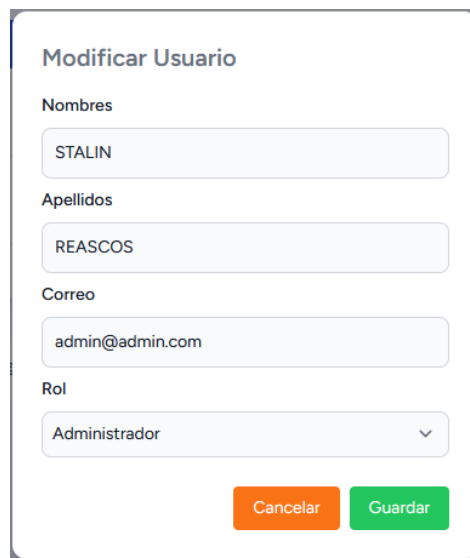
ADMINISTRACIÓN USUARIOS							
Buscar...							
NOMBRE	APELLIDO	CORREO	ROL	CONTRASEÑA	FECHA CREACIÓN	ACTION	
STALIN	REASCOS	admin@admin.com	Administrador	Cambiar	2025-06-17 12:46:09	Editar	Eliminar
TEST	USER	test@user.com	Usuario	Cambiar	2025-06-17 14:36:05	Editar	Eliminar

Ilustración 83. Detalle de Usuarios Registrados.

Editar Usuario

Para poder editar la información de un Usuario únicamente nos dirigimos al listado detallado de usuarios, ubicamos al usuario que queremos editar (Podemos utilizar la opción de filtro para ubicarlo más rápido), y en sus acciones damos clic en **“Editar”**, esto nos abrir una

vista con el formulario de edición, el cual contendrá en los campos la información del Usuario, dentro de este formulario el campo de código no se podrán editar ya que es un código único con el que se le registro en el sistema al usuario, con respecto a la contraseña su edición será realizada mediante otro formulario, adicional a esto nos permitirá escoger o cambiar el rol del usuario (Administrador o Usuario), de igual manera los campos con * serán obligatorios.

El formulario se titula "Modificar Usuario". Contiene cuatro campos de texto: "Nombres" con el valor "STALIN", "Apellidos" con el valor "REASCOS", "Correo" con el valor "admin@admin.com", y "Rol" que es un menú desplegable con "Administrador" seleccionado. En la parte inferior hay dos botones: "Cancelar" (naranja) y "Guardar" (verde).

Modificar Usuario	
Nombres	STALIN
Apellidos	REASCOS
Correo	admin@admin.com
Rol	Administrador
Cancelar Guardar	

Ilustración 84. Formulario edición Usuario.

Para guardar en la base de datos del sistema, únicamente damos clic en el botón **“Guardar”**.

Actualizar contraseña Usuario

Para poder actualizar la contraseña de un usuario únicamente nos ubicamos en el registro y damos clic en la opción **“Cambiar”**, nos aparecerá un formulario donde debemos ingresar la nueva contraseña y confirmarla, posterior a esto damos clic en la opción **“Guardar”**, para que se actualice en la base de datos.

Modificar Contraseña
STALIN REASCOS

Contraseña

.....

Confirmar contraseña

.....

Cancelar Guardar

Ilustración 85. Actualizar contraseña Usuario.

Eliminar Usuario

Antes de pasar a eliminar un usuario del sistema, debemos tener en cuenta que esta acción solo la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar los usuarios que no tengan ninguna transacción en el sistema.

Para poder realizar esta eliminación nos ubicamos en el Usuario que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido al usuario en el caso de que existan más de 20 registros.

Ya ubicados en el usuario escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el código y nombre del Usuario que vamos a eliminar, damos clic en el botón **“Confirmar”**, y automáticamente se eliminara el registro de la base de datos.

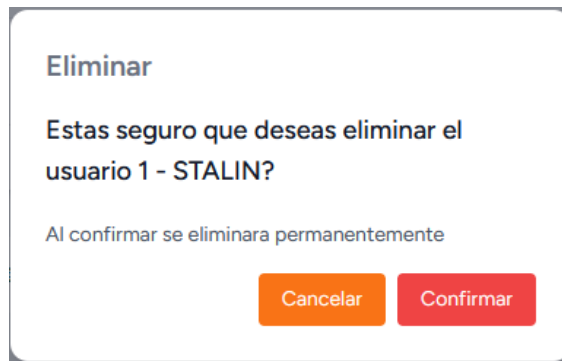


Ilustración 86. Eliminación de Usuario.

Reporte básico de Usuarios

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los usuarios registrados.

Para poder generar este reporte damos clic en el icono de listado que se encuentra alado del botón “*Crear Usuario*”, posterior a esta acción el sistema mostrará un modal en donde cargará el informe con el detalle de los Usuarios que se han registrado en la base de datos.



Ilustración 87. Reporte Listado Usuarios.

Administración Muestras Médicas

Ingreso muestras médicas

Dentro de nuestro DASHBOARDS **“Inventario”**, nos dirigimos a la opción de **“Ingresos”**, los ingresos son parte fundamental del sistema, es el proceso donde las muestras médicas son registradas en la base de datos con el fin de tener un control adecuado de las mismas, para esto se ha desarrollado un módulo de ingreso, así como también, una interfaz de administración con las siguientes opciones:

Opciones Ingreso muestras médicas

OPCIÓN	DESCRIPCIÓN
--------	-------------

	Botón Nuevo: permite crear un nuevo ingreso de muestras médicas al sistema.
	Botón PDF: crea el reporte en formato PDF del ingreso.
	Botón Filtro: permite filtrar los lotes por algún criterio (Número Ingreso, Fecha, etc.).
	Botón Visualizar: permite tener una vista previa del ingreso.
	Botón Eliminar: permite eliminar el registro del ingreso en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 11. Opciones Ingreso muestras médicas.

Crear nuevo ingreso

Como se ha visto en la tabla anterior para crear un nuevo ingreso únicamente debemos dar clic en el botón “**Nuevo**”, este nos redirigirá a una interfaz con el siguiente aspecto:

The screenshot shows a web application interface for inventory management. At the top, there is a navigation bar with 'Módulos', 'Inventario', and 'Usuarios' links, and a user profile 'STALIN REASCOS - admin'. The main content area is divided into two panels. The left panel, titled 'DATOS ENTRADA', contains form fields for 'FECHA ENTRADA' (21/07/2025), 'PROVEEDOR' (dropdown), 'PRODUCTO' (dropdown), 'Cantidad' (1), 'Lote' (NUMERO LOTE), 'Fabricación' (dd/mm/aaaa), and 'Vencimiento' (dd/mm/aaaa). There is an 'Agregar' button and an 'Observaciones' text area. The right panel, titled 'DETALLES', shows a table with columns: IDPROD, PRODUCTO, LOTE, CANTIDAD, F. FABRICACIÓN, F. VENCIMIENTO, and ACCIÓN. Below the table, it says 'TOTAL UNIDADES: 0'. There are 'Cancelar' and 'Crear Entrada' buttons. At the bottom, a copyright notice reads '© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.'

Ilustración 88. Interfaz Ingresos.

Nota: para poder realizar un ingreso se debe tener en cuenta que los proveedores y productos correspondientes a ese proveedor tienen que estar previamente registrados en el sistema, adicional cada ingreso pertenece a un solo proveedor, es decir, no se podrán mezclar productos de diferentes proveedores dentro del mismo ingreso.

Como se puede observar en la ilustración 88, la interfaz se divide en dos secciones, la sección de la izquierda representa los datos principales del ingreso, en donde todos los datos que se encuentran con el signo * son obligatorios.

This screenshot shows the 'DATOS ENTRADA' form with specific data entered. The fields are: '* FECHA ENTRADA' (21/07/2025), '* PROVEEDOR' (DENTAL DENTSPRAY), '* PRODUCTO' (COLORIMETRO IVOSTAR A1), '* Cantidad' (100), '* Lote' (COLIV00100), '* Fabricación' (21/07/2025), and '* Vencimiento' (31/07/2029). The 'Agregar' button is visible. The 'Observaciones' text area contains the text 'COMPRA COLORIMETROS IVOSTAR A1'.

Ilustración 89. Datos fundamentales del Ingreso.

Nota: el formulario valida que todos los campos que son obligatorios se encuentren correctamente llenos, adicional se debe tener en cuenta que la fecha de fabricación no puede ser mayor a la fecha actual, así como también, la fecha de vencimiento no puede ser menor a la fecha actual ni de fabricación, el campo Observaciones es opcional.

Una vez llena la información en la parte izquierda del ingreso damos clic en el botón **“Agregar”**, es realizara la acción de colocar a detalle este registro de producto como una visualización previa en la parte derecha del ingreso.

DETALLES						
IDPROD	PRODUCTO	LOTE	CANTIDAD	F. FABRICACIÓN	F. VENCIMIENTO	ACCIÓN
4	COLORIMETRO IVOSTAR A1	coliv00100	100	2025-07-21	2029-07-31	Eliminar
5	COLORIMETRO IVOSTAR A2	coliv00102	100	2025-07-21	2027-07-31	Eliminar
TOTAL UNIDADES: 200						
Cancelar Crear Entrada						

Ilustración 90. Visualización detalles del ingreso.

Nota: se puede ingresar desde 1 en detalles de productos en el ingreso, adicional permite eliminar detalles del mismo.

Finalmente revisado todos los detalles del ingreso únicamente damos clic en **“Guardar”**, y el sistema almacenara esta información en su base de datos, este paso de creación es muy sencillo y una vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de Ingresos en donde ya se encontrará detallado en el listado el ingreso creado.

ADMINISTRACIÓN INGRESOS					
Buscar...					
NUMERO ENTRADA	PROVEEDOR	FECHA	OBSERVACIONES	ACCIONES	
1	P1709217457001 DENTAL DENTSPRAY	2025-06-30	COMPRA MATERIALES DENTALES DENTAL DENTSPRAY	PDF	Visualizar Eliminar
2	P1719307967001 DENTAL AYALA	2025-06-30	COMPRA MUESTRAS DE LAMINAS DE CERA DENTAL AYALA	PDF	Visualizar Eliminar
3	P1719307975001 DENTAL ALTAMIRANO	2025-06-30	COMPRA TABLETAS DE DIENTES COLOR A2 INFERIORES A DENTAL ALTAMIRANO	PDF	Visualizar Eliminar
4	P1719307967001 DENTAL AYALA	2025-06-30		PDF	Visualizar Eliminar
5	P1709217457001 DENTAL DENTSPRAY	2025-06-30	COMPRA DE MUESTRAS MEDICAS DENTAL DENTSPRAY	PDF	Visualizar Eliminar
6	P1709217457001 DENTAL DENTSPRAY	2025-07-21	COMPRA COLORIMETROS IVOSTAR A1 Y A2	PDF	Visualizar Eliminar

Ilustración 91. Detalle de Ingresos registrados.

Eliminar Ingreso

Antes de pasar a eliminar un ingreso del sistema, debemos tener en cuenta que esta acción solo la podrá visualizar y realizar el administrador del sistema, adicional únicamente se podrán eliminar los ingresos que no tengan ninguna transacción en el sistema, es decir que los detalles de los productos que se registraron en el ingreso no estén comprometidos en un egreso de muestras médicas.

Para poder realizar esta eliminación nos ubicamos en el ingreso que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido el ingreso en el caso de que existan más de 20 registros.

Ya ubicados en el ingreso escogemos la opción de **“Eliminar”**, la cual nos presentara un modal de confirmación con el número y nombre del proveedor registrados en el ingreso, damos clic en el botón **“Confirmar”**, y automáticamente se eliminará el registro de la base de datos.

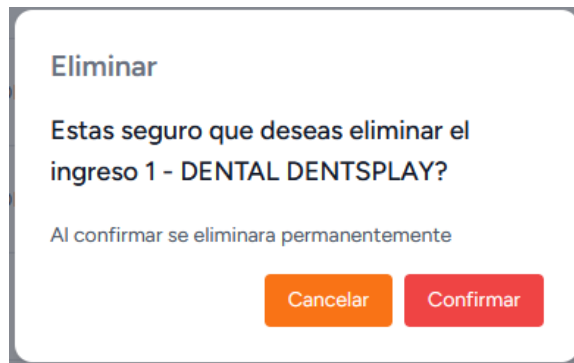


Ilustración 92. Eliminación de Ingresos.

Reporte básico de Ingresos

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los productos registrados en un ingreso, adicional podremos revisar el ingreso creado únicamente como una vista previa.

Para poder revisar el PDF de un ingreso, nos colocamos en el ingreso y damos clic en el botón PDF, este nos presentara el documento del ingreso el cual podremos imprimir para control, en el caso de que solo quisiéramos visualizar el ingreso damos clic en el botón de visualizar.

1 de 1 90%

C.G.M.M.
CONTROL Y GESTIÓN DE
MUESTRAS MÉDICAS

FECHA IMPRESIÓN	HORA IMPRESIÓN	USUARIO
2025-07-21	13:58:25	STALIN REASCOS

Impresión del Ingreso

Ingreso Num.	Fecha.	Proveedor
1	2025-06-30	DENTAL DENTSPLAY

Detalle de Productos

Cantidad	Producto	Lote	Elaboración	Vencimiento
15	LAMINAS DE CERA 60CM	LAC0001	2025-06-30	2029-06-30
10	COLORIMETRO IVOSTAR A1	COLIVA1001	2025-06-30	2029-06-30
10	COLORIMETRO IVOSTAR A2	COLIVA2001	2025-06-30	2029-06-30
Total Unidades 35				

Elaborado por:	Aprobado Por:
STALIN REASCOS	GERENCIA

Ilustración 93. PDF Ingreso.

Visualización del Ingreso

INGRESO NUM.	FECHA.	PROVEEDOR
1	2025-06-30	DENTAL DENTSPLAY

Detalle de Productos

CANTIDAD	PRODUCTO	LOTE	ELABORACIÓN	VENCIMIENTO
15	LAMINAS DE CERA 60CM	LAC0001	2025-06-30	2029-06-30
10	COLORIMETRO IVOSTAR A1	COLIVA1001	2025-06-30	2029-06-30
10	COLORIMETRO IVOSTAR A2	COLIVA2001	2025-06-30	2029-06-30
Total Unidades 35				

Cancelar

Ilustración 94. Vista Previa Ingreso.

Egreso muestras médicas

Dentro de nuestro DASHBOARDS *“Inventario”*, nos dirigimos a la opción de *“Egresos”*, los egresos son parte fundamental del sistema, es el proceso donde las muestras médicas son registradas como una salida en la base de datos con el fin de tener un control adecuado de las mismas, para esto se ha desarrollado un módulo de egreso, así como también, una interfaz de administración con las siguientes opciones:

Opciones Egreso muestras médicas

OPCIÓN	DESCRIPCIÓN
	Botón Nuevo: permite crear un nuevo egreso de muestras médicas al sistema.
	Botón PDF: crea el reporte en formato PDF del ingreso.
	Botón Filtro: permite filtrar los lotes por algún criterio (Número Egreso, Fecha, etc.).
	Botón Visualizar: permite tener una vista previa del egreso.
	Botón Eliminar: permite eliminar el registro del egreso en la base de datos. (Únicamente puede ser visualizado por el administrador del sistema)

Tabla 12. Opciones Egreso muestras médicas.

Crear nuevo egreso

Como se ha visto en la tabla anterior para crear un nuevo egreso únicamente debemos dar clic en el botón *“Nuevo”*, este nos redirigirá a una interfaz con el siguiente aspecto:

ModulosInventarioUsuarios

STALIN REASCOS - admin

DATOS SALIDA

* REPRESENTANTE DE VENTA

Seleccione representante

* FECHA SALIDA

21 / 07 / 2025

* STOCK DISPONIBLE

Buscar en la tabla

#	CODIGO	PRODUCTO	LOTE	VENCIMIENTO	STOCK	CANTIDAD	ACCIÓN
1	COL01	COLORIMETRO IVOSTAR A1	COLIVA1001	2029-06-30	8	Digite cantidad	Asignar
2	COL02	COLORIMETRO IVOSTAR A2	COLIVA2001	2029-06-30	2	Digite cantidad	Asignar
3	LAC01	LAMINAS DE CERA	LADA0001	2030-06-30	5	Digite cantidad	Asignar
4	TAB04	TABLETA DIENTES INF A2	TABDINF A2	2028-12-30	70	Digite cantidad	Asignar
5	LAC01	LAMINAS DE CERA	LAMC0102	2029-02-02	150	Digite cantidad	Asignar
6	LAC02	LAMINAS DE CERA 60CM	LAMCDD001	2050-10-10	300	Digite cantidad	Asignar
7	LAC02	LAMINAS DE CERA 60CM	LAMCDD002	2050-10-10	300	Digite cantidad	Asignar

Observaciones

Escribe tus observaciones...

DETALLES

CODE	PRODUCTO	LOTE	VTO	CANTIDAD	ACCIÓN
Total Unidades					
0					
Cancelar Crear Salida					

© 2025 Stalin Reascos, Andrés Santamaría All Rights Reserved.

Ilustración 95. Interfaz Egresos.

Nota: para poder realizar un egreso se debe tener en cuenta que los representantes tienen que estar previamente registrados en el sistema, así como también, los productos que vamos a seleccionar deben estar previamente registrados dentro de un ingreso para poder disponer de stock, adicional cada egreso pertenece a un solo representante, es decir, no se podrán mezclar representantes dentro del mismo egreso.

Como se puede observar en la ilustración 95, la interfaz se divide en dos secciones, la sección de la izquierda representa los datos principales del egreso, en donde todos los datos que se encuentran con el signo * son obligatorios, adicional esta interfaz proporciona de un buscador con el que podamos localizar el producto deseado de manera más rápida.

DATOS SALIDA

* REPRESENTANTE DE VENTA

Seleccione representante

* FECHA SALIDA

21 / 07 / 2025

Ilustración 96. Datos principales Egreso.

* STOCK DISPONIBLE

Buscar en la tabla

#	CODIGO	PRODUCTO	LOTE	VENCIMIENTO	STOCK	CANTIDAD	ACCIÓN
1	COL01	COLORIMETRO IVOSTAR A1	COLIVA1001	2029-06-30	8	Digite cantidad	<button>Asignar</button>
2	COL02	COLORIMETRO IVOSTAR A2	COLIVA2001	2029-06-30	2	Digite cantidad	<button>Asignar</button>
3	LAC01	LAMINAS DE CERA	LADA0001	2030-06-30	5	Digite cantidad	<button>Asignar</button>
4	TAB04	TABLETA DIENTES INF A2	TABDINFA2	2028-12-30	70	Digite cantidad	<button>Asignar</button>
5	LAC01	LAMINAS DE CERA	LAMC0102	2029-02-02	150	Digite cantidad	<button>Asignar</button>
6	LAC02	LAMINAS DE CERA 60CM	LAMCDD001	2050-10-10	300	Digite cantidad	<button>Asignar</button>
7	LAC02	LAMINAS DE CERA 60CM	LAMCDD002	2050-10-10	300	Digite cantidad	<button>Asignar</button>

Ilustración 97. Buscador de stock.

Nota: el formulario valida que todos los campos que son obligatorios se encuentren correctamente llenos, adicional se debe tener en cuenta que todo el stock disponible se encuentra detallado en esta sección, y que para poder asignar un producto al detalle se debe colocar la cantidad, dicha cantidad no puede exceder al stock disponible, el campo Observaciones es opcional.

Una vez llena la información en la parte izquierda del ingreso damos clic en el botón “**Asignar**”, el cual realizará la acción de colocar en la sección detalles esta asignación de producto como una visualización previa en la parte derecha del egreso.

DETALLES					
CODE	PRODUCTO	LOTE	VTO	CANTIDAD	ACCIÓN
COL01	COLORIMETRO IVOSTAR A1	COLIVA1001	2029-06-30	8	Eliminar
COL02	COLORIMETRO IVOSTAR A2	COLIVA2001	2029-06-30	2	Eliminar
LAC01	LAMINAS DE CERA	LADA0001	2030-06-30	5	Eliminar
Total Unidades					
15					
Cancelar Crear Salida					

Ilustración 98. Visualización detalles del ingreso.

Nota: se puede ingresar desde 1 en detalles de productos en el egreso, adicional permite eliminar detalles del mismo.

Finalmente revisado todos los detalles del egreso únicamente damos clic en **“Guardar Salida”**, y el sistema almacenará esta información en su base de datos, este paso de creación es muy sencillo y una vez que se haya almacenado los datos el sistema nos redirige a la vista de administración de Egresos en donde ya se encontrará detallado en el listado el ingreso creado.

ADMINISTRACIÓN EGRESOS				
Buscar...				
NÚMERO SALIDA	REPRESENTANTE VENTA	FECHA	OBSERVACIONES	ACCIONES
3	AMA 1 ALEJANDRO MANTILLA	2025-06-30		PDF Visualizar Eliminar
4	DZO 1 DISTRIBUIDORA QUITO	2025-07-21		PDF Visualizar Eliminar

Ilustración 99. Detalle de Egresos registrados.

Eliminar Egreso

Antes de pasar a eliminar un Egreso del sistema, debemos tener en cuenta que esta acción solo la podrá visualizar y realizar el administrador del sistema.

Para poder realizar esta eliminación nos ubicamos en el egreso que deseamos eliminar, como se había explicado anteriormente podemos realizar una búsqueda por filtro para localizar más rápido el egreso en el caso de que existan más de 20 registros.

Ya ubicados en el egreso escogemos la opción de **“Eliminar”**, la cual nos presentará un modal de confirmación con el número y nombre del representante registrados en el egreso, damos clic en el botón **“Confirmar”**, y automáticamente se eliminará el registro de la base de datos.

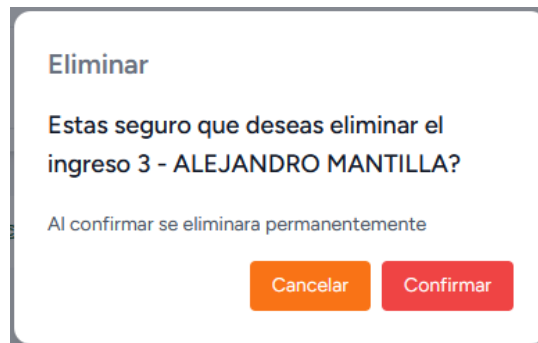


Ilustración 100. Eliminación de Egresos.

Reporte básico de Egresos

Como en todo sistema, la reportería es importante para un buen control de la información para esto el sistema nos permite visualizar un reporte en formato PDF con el listado de los productos registrados en un egreso, adicional podremos revisar el egreso creado únicamente como una vista previa.

Para poder revisar el PDF de un egreso, nos colocamos en el egreso y damos clic en el botón PDF, este nos presentara el documento del egreso el cual podremos imprimir para control, en el caso de que solo quisiéramos visualizar el egreso damos clic en el botón de visualizar.

1 de 190%

CONTROL Y GESTIÓN DE MUESTRAS MÉDICAS

FECHA IMPRESIÓN	HORA IMPRESIÓN	USUARIO
2025-07-21	14:24:40	STALIN REASCOS

Impresión del Egreso

Egreso Num.	Fecha.	Representante
3	2025-06-30	ALEJANDRO MANTILLA

Detalle de Productos

Cantidad	Producto	Lote	Elaboración	Vencimiento
15	LAMINAS DE CERA 60CM	LAC0001	2025-06-30	2029-06-30
2	COLORIMETRO IVOSTAR A1	COLIVA1001	2025-06-30	2029-06-30
8	COLORIMETRO IVOSTAR A2	COLIVA2001	2025-06-30	2029-06-30
5	LAMINAS DE CERA	LADA0001	2022-05-05	2030-06-30
30	TABLETA DIENTES INF A2	TABDINFA2	2024-01-05	2028-12-30
Total Unidades 60				

Elaborado por:
STALIN REASCOS

Aprobado Por:
GERENCIA

Ilustración 101. PDF Egreso.

Visualización del Egreso

INGRESO NUM.	FECHA.	REPRESENTANTE VENTA
3		ALEJANDRO MANTILLA

Detalle de Productos

CANTIDAD	PRODUCTO	LOTE	ELABORACIÓN	VENCIMIENTO
15	LAMINAS DE CERA 60CM	LAC0001	2025-06-30	2029-06-30
2	COLORIMETRO IVOSTAR A1	COLIVA1001	2025-06-30	2029-06-30
8	COLORIMETRO IVOSTAR A2	COLIVA2001	2025-06-30	2029-06-30
5	LAMINAS DE CERA	LADA0001	2022-05-05	2030-06-30
30	TABLETA DIENTES INF A2	TABDINFA2	2024-01-05	2028-12-30
Total Unidades 60				

Cancelar

Ilustración 102. Vista Previa Egreso.

Reporte general stock

El reporte general de stock nos permite revisar las cantidades existentes que disponemos para poder realizar nuestros egresos en el sistema, para poder revisar el reporte debemos dirigirnos al Dashboards **“Inventario”**, seleccionamos la opción **“Stock General”**, y el sistema nos redirige a una interfaz donde muestra el stock disponible con el cual contamos en tiempo real.

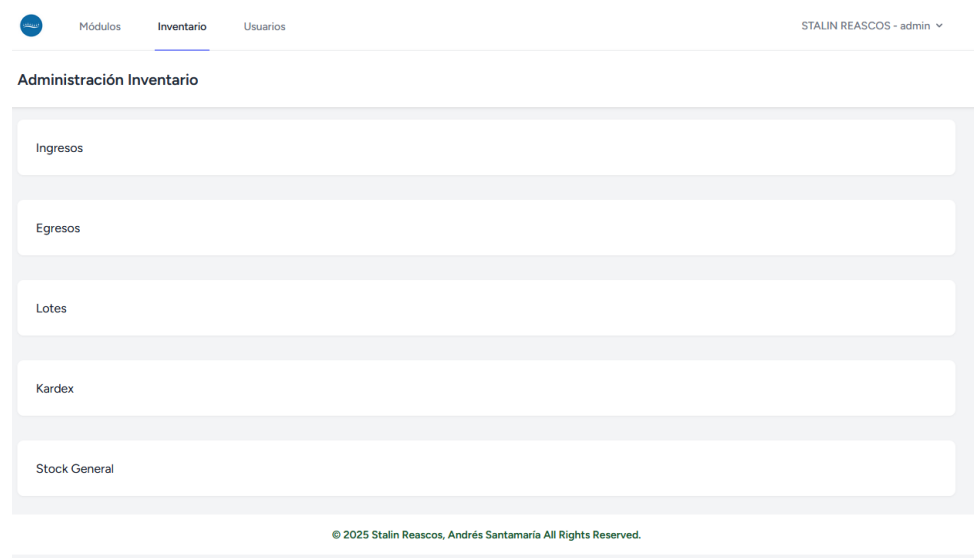


Ilustración 103. DASHBOARDS Inventario.

STOCK GENERAL						
Buscar en la tabla						
#	CODIGO	PRODUCTO	PROVEEDOR	LOTE	VENCIMIENTO	STOCK
1	TAB04	TABLETA DIENTES INF A2	DENTAL ALTAMIRANO	TABDINF02	2028-12-30	70
2	LAC01	LAMINAS DE CERA	DENTAL AYALA	LAMC0102	2029-02-02	150
3	LAC02	LAMINAS DE CERA 60CM	DENTAL DENTSPLAY	LAMCDD001	2050-10-10	300
4	LAC02	LAMINAS DE CERA 60CM	DENTAL DENTSPLAY	LAMCDD002	2050-10-10	300
5	LAC02	LAMINAS DE CERA 60CM	DENTAL DENTSPLAY	LAMCDD003	2050-10-10	300
6	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV0010	2050-10-10	300
7	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV0011	2050-10-10	300
8	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV0013	2050-10-10	300
9	COL01	COLORIMETRO IVOSTAR A1	DENTAL DENTSPLAY	COLIV00100	2025-05-31	100
10	COL02	COLORIMETRO IVOSTAR A2	DENTAL DENTSPLAY	COLIV00102	2027-07-31	100

Ilustración 104. Stock Disponible.

A handwritten signature in black ink, appearing to read "Ricardo", with several horizontal strokes extending from the end.

Ricardo Andrés Santamaría Rosario

C.I.: 1723820872

A handwritten signature in blue ink, appearing to read "Stalin", with a large, stylized initial and a long horizontal stroke.

Stalin Vinicio Reascos Mantilla

C.I.: 1719307967

MANUAL DEL PROGRAMADOR

Tecnologías utilizadas

- Backend: Laravel 12 (PHP)
- Frontend: Blade Templates
- Base de Datos: MySQL v8.4.3
- Apache v2.4.62
- Servidor Web Local: Apache/Nginx integrado en Laragon
- Control de versiones: Git
- Node.js

Requisitos del sistema

Hardware necesario

- Sistema Operativo: Windows 10 o superior
- Procesador Intel Core I3 o superior.
- RAM mínima: 4 GB
- HDD 50GB.

Software necesario

- Laragon 2025 v8
- Visual Studio Code v1.87.2
- Navegador web moderno (Chrome, Firefox, etc.)

- Git
- Node.js

Instalación de software requerido

Como se había mencionado en las especificaciones de software y hardware, debemos tener instalado un sistema operativo Windows 10 o superior, adicional memoria RAM de 4GB, en donde podamos usar mínimo 4 procesadores del sistema como se muestra a continuación:

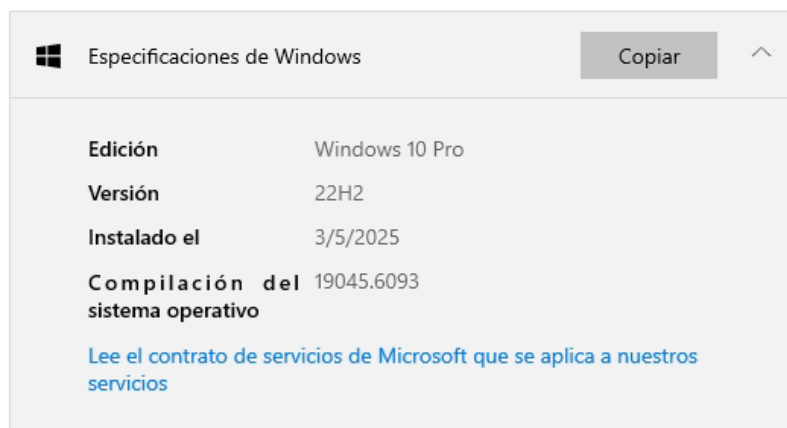


Ilustración 105. Sistema Operativo.



Ilustración 106. Especificaciones Hardware.

Instalación de Laragon

Laragon es un entorno de desarrollo local gratuito para Windows que facilita la creación y gestión de aplicaciones web, es especialmente útil para desarrolladores que trabajan con PHP, Node.js, etc.

Para el desarrollo de este proyecto instalamos la versión 2025 v8, que para el tiempo en la que estamos desarrollando esta aplicación, la podemos encontrar en el siguiente enlace <https://laragon.org/download> descargamos la versión que necesitamos y nos dirigimos a la carpeta de descargas para empezar con la instalación.

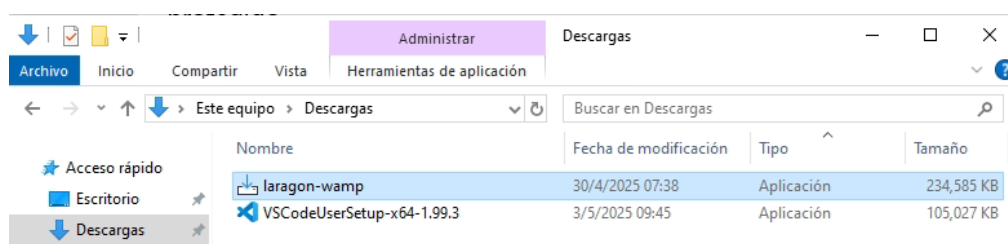


Ilustración 107. Directorio Descargas.

Una vez nos encontramos ubicados en el ejecutable llamado **Laragon-wamp.exe**, lo ejecutamos con permisos de administrador para que pueda realizar cambios en el sistema y realizamos los pasos que se muestran en las siguientes imágenes:

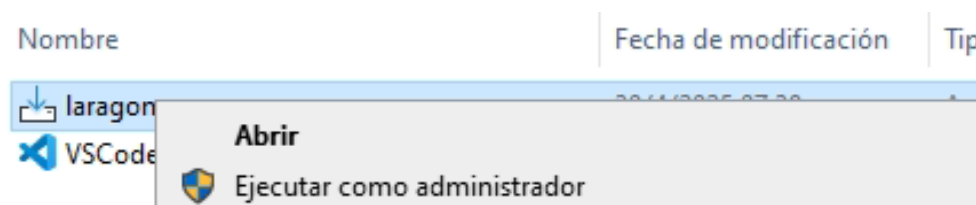


Ilustración 108. Ejecución como administrador.

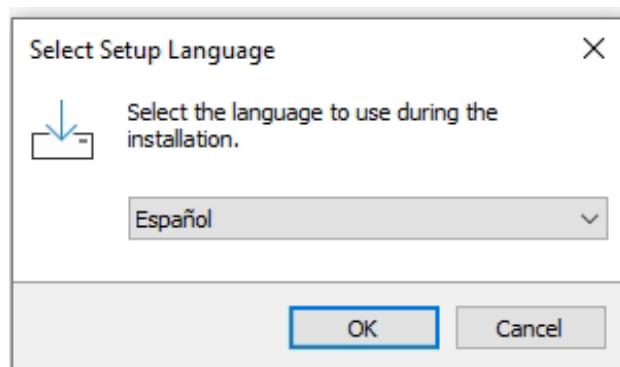


Ilustración 109. Selección Idioma.

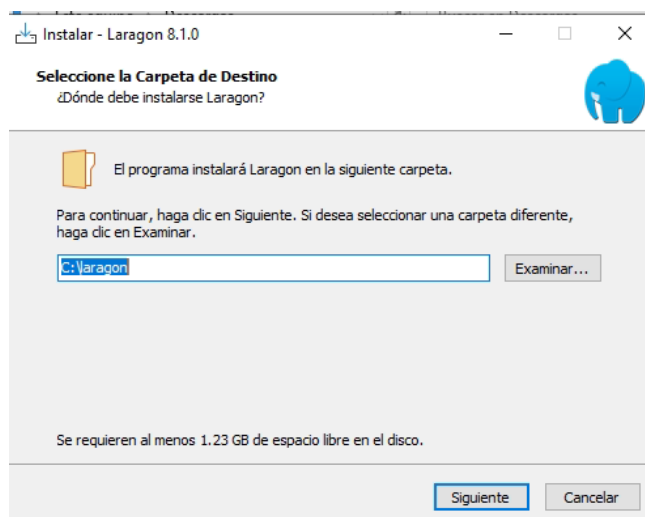


Ilustración 110. Selección directorio instalación.

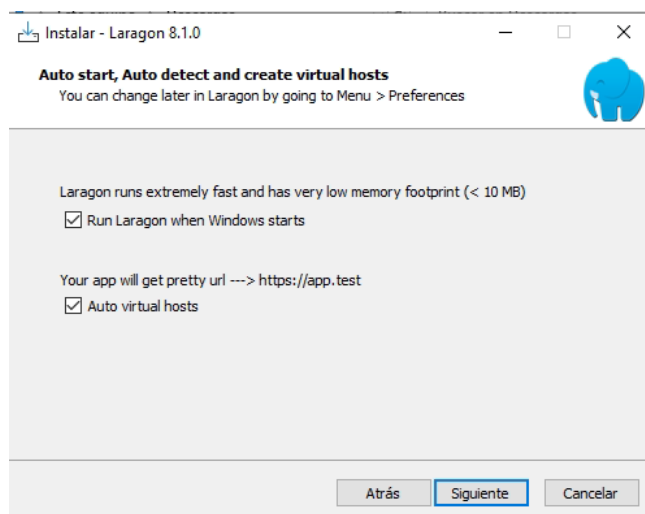


Ilustración 111. Opciones por defecto.

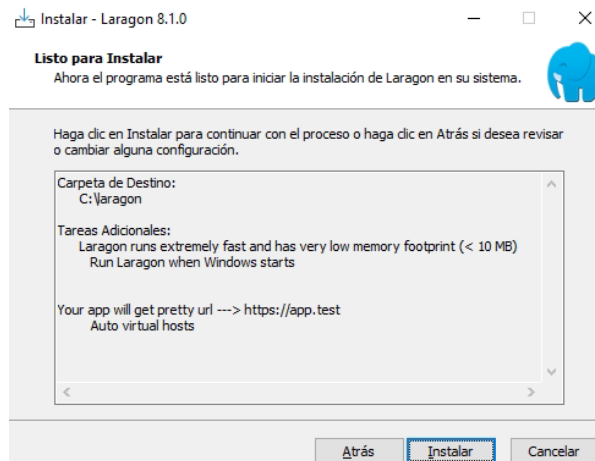


Ilustración 112. Inicio instalación Laragon.

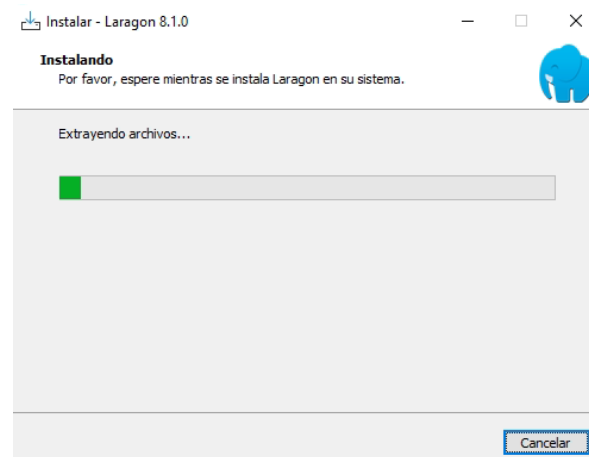


Ilustración 113. Proceso instalación Laragon.

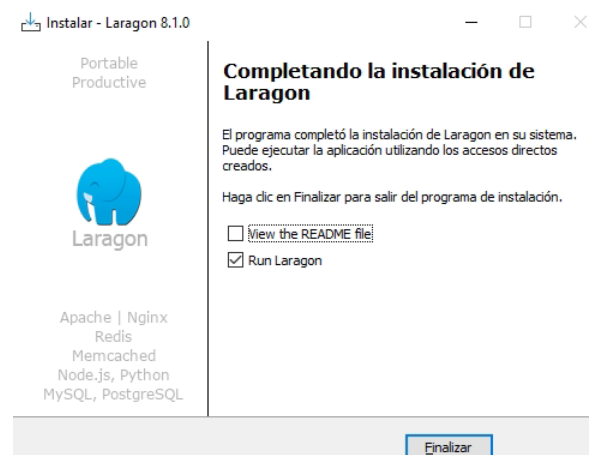


Ilustración 114. Finalización instalación.

Instalación de Node.js

Node.js es una plataforma de ejecución para JavaScript en el servidor, en lugar de usar JavaScript solo en el navegador (como se hacía originalmente), por lo tanto esta tecnología también se necesita instalar en el sistema operativo, para hacerlo descargamos el programa directamente del link <https://nodejs.org/es/download>, y seguimos estas instrucciones:

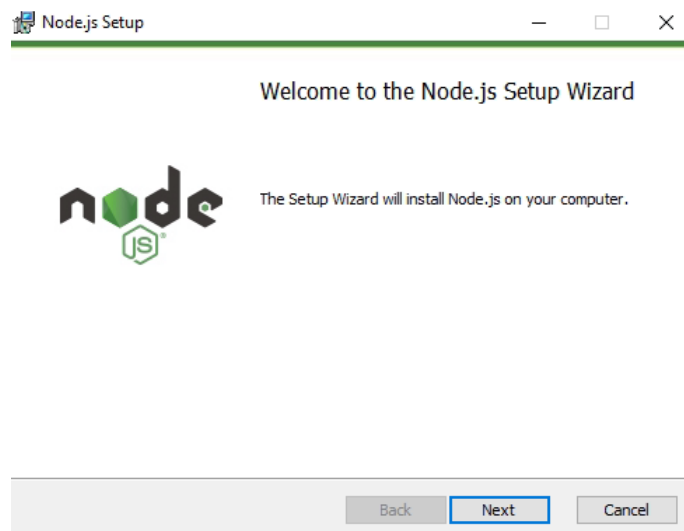


Ilustración 115. Ejecutar instalador node.

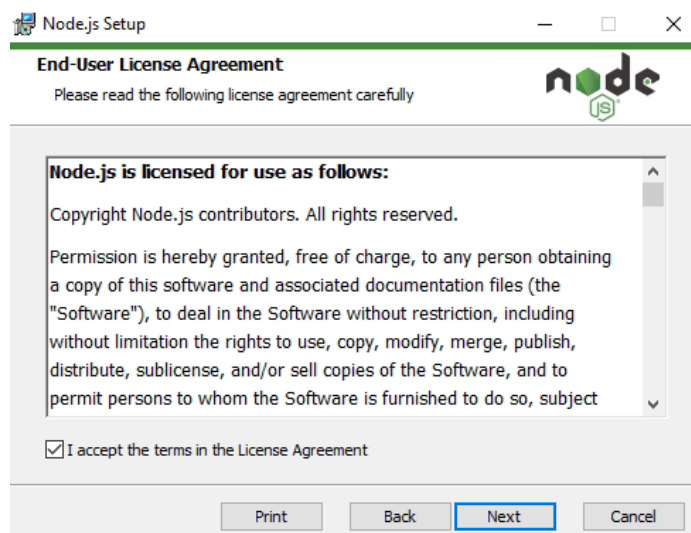


Ilustración 116. Aceptar Términos.

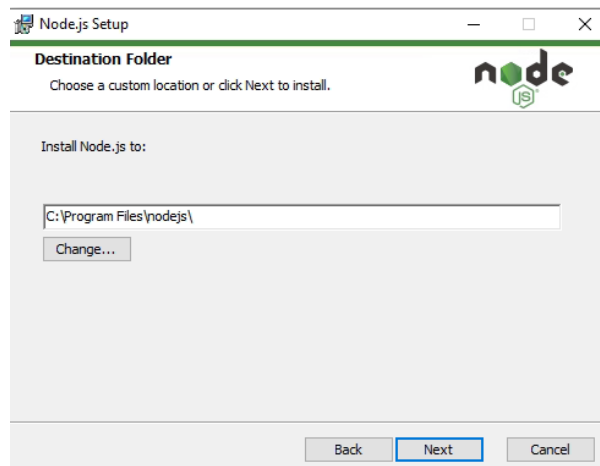


Ilustración 117. Directorio de instalación.

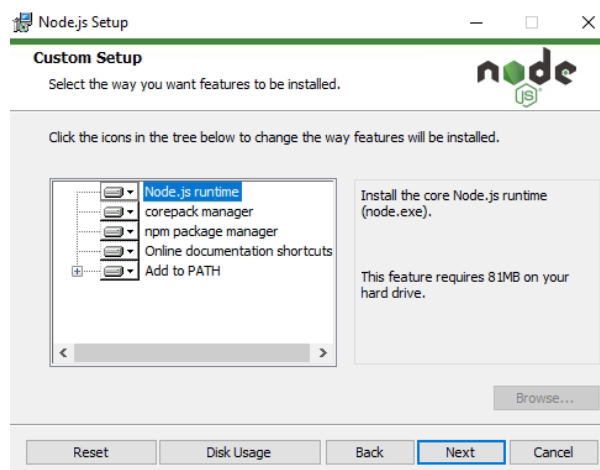


Ilustración 118. Opciones de instalación.

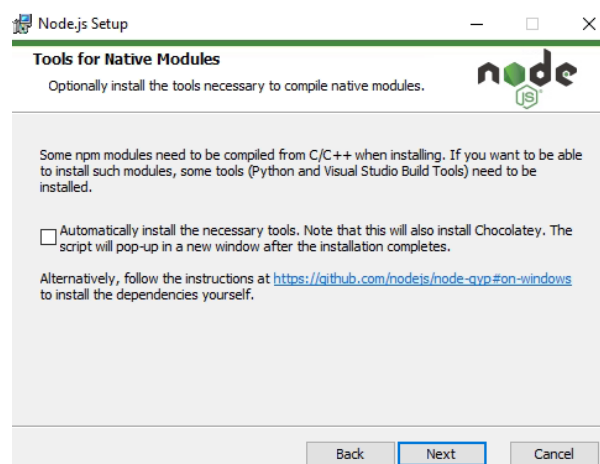


Ilustración 119. Opciones de instalación.

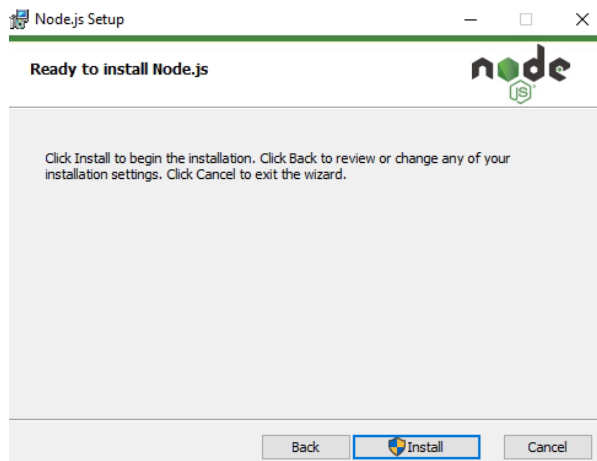


Ilustración 120. Inicio de instalación.

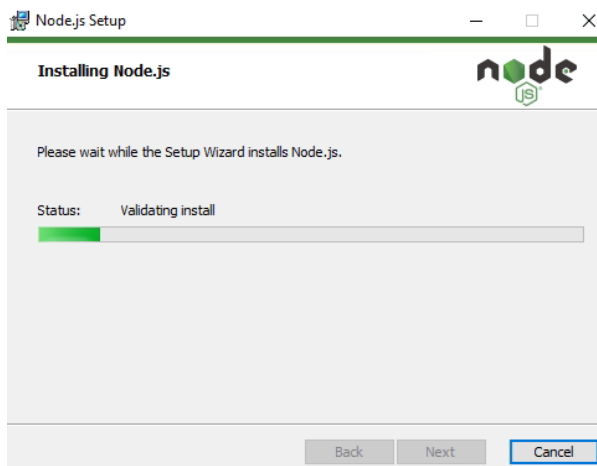


Ilustración 121. Progreso de instalación.

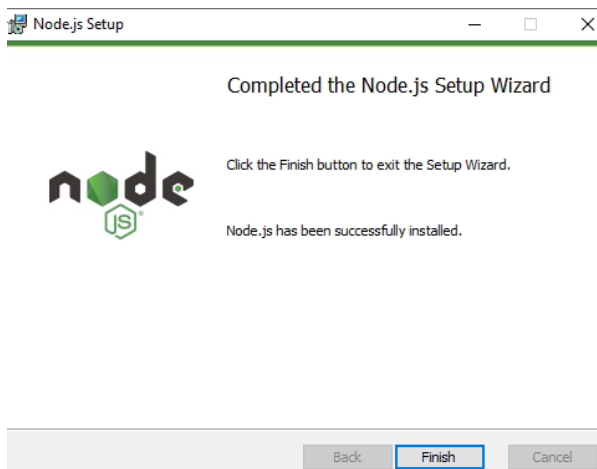


Ilustración 122. Instalación finalizada.

Instalación Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente ligero y potente, desarrollado por Microsoft, que funciona en múltiples plataformas (para el desarrollo de este sistema web usamos Windows), por lo cual realizamos los mismos pasos detallados en el apartado anterior, entonces procedemos a descargar **VS Code** desde el siguiente link <https://code.visualstudio.com/>, de la misma manera nos ubicamos en el directorio donde se descargó el programa y lo ejecutamos como administrador para que pueda realizar cambios en el sistema, adicional procedemos a seguir estos pasos que se detallan en las próximas ilustraciones:

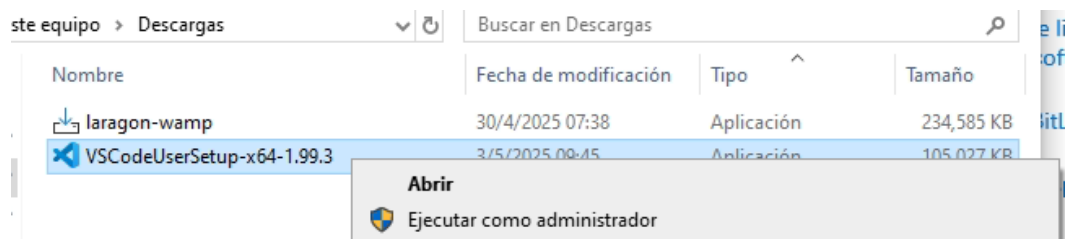


Ilustración 123. Directorio descargas.

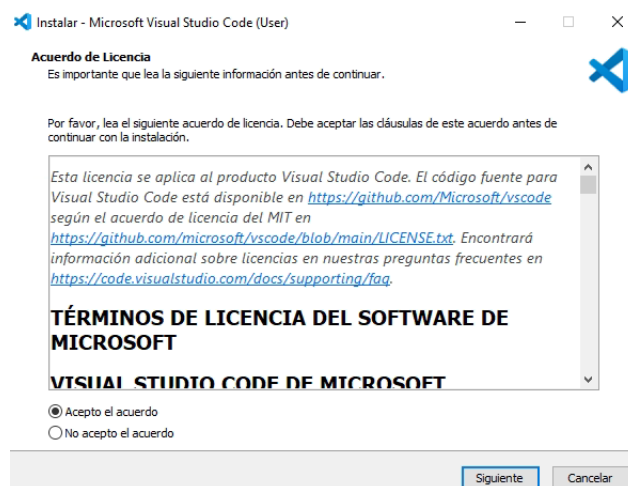


Ilustración 124. Aceptación de términos

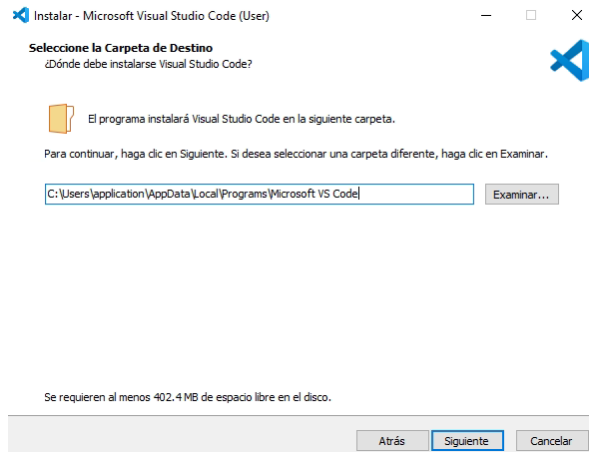


Ilustración 125. Directorio de instalación VSC.

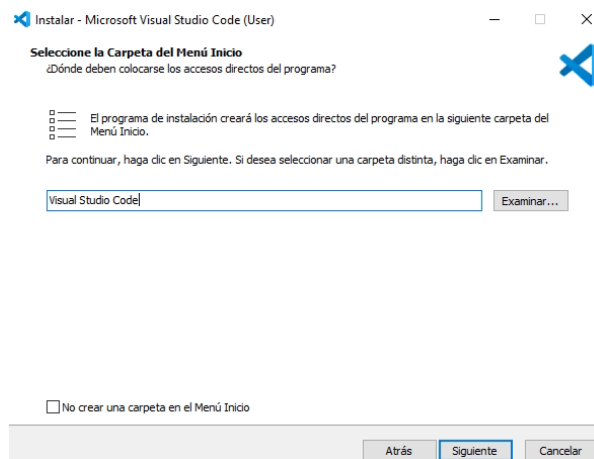


Ilustración 126. Opciones predeterminadas.

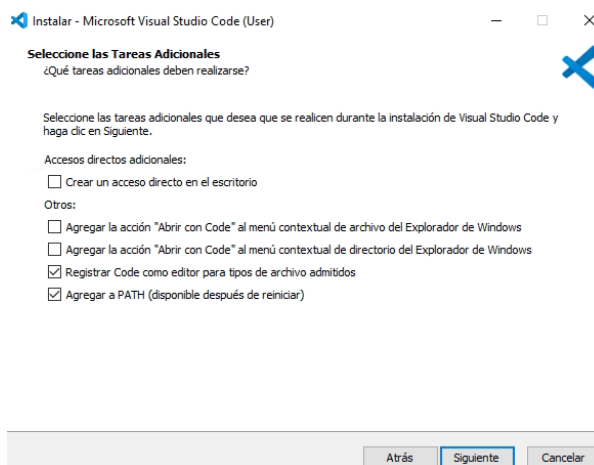


Ilustración 127. Opciones predeterminadas.

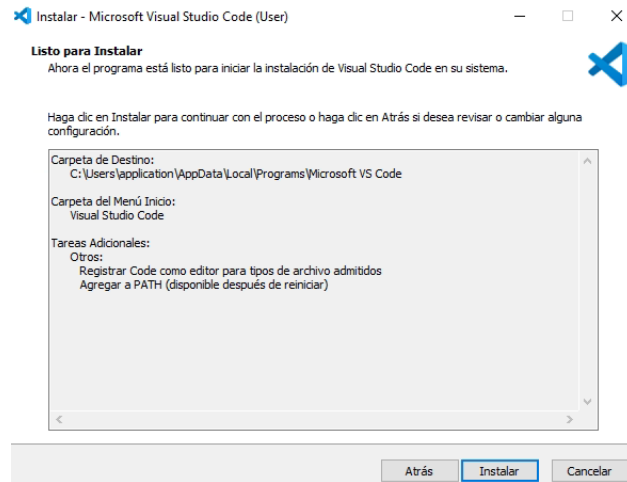


Ilustración 128. Inicio instalación VSC.

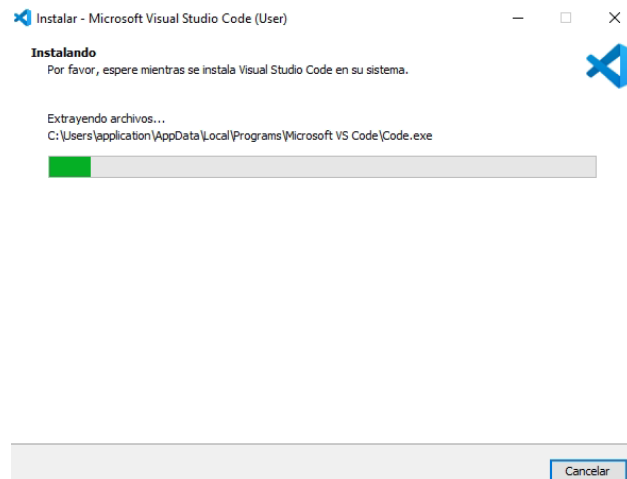


Ilustración 129. Progreso instalación VSC.

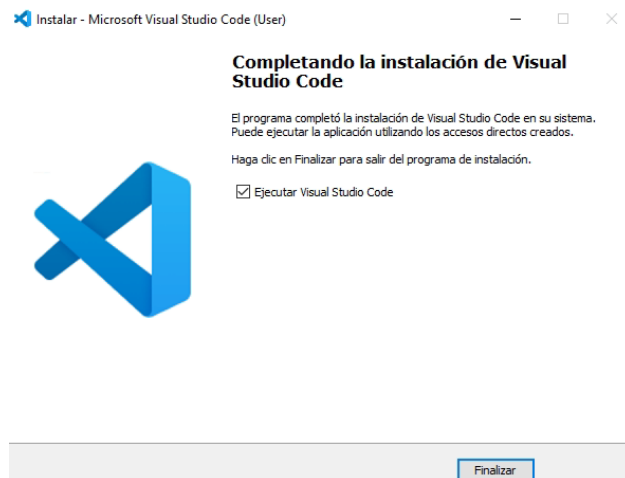


Ilustración 130. Instalación finalizada.

Instalación Git

Git es un sistema de control de versiones distribuido, gratuito y de código abierto. Se utiliza para rastrear los cambios en los archivos de un proyecto y permite a los desarrolladores trabajar de forma colaborativa en el mismo proyecto de manera eficiente, la utilización de **Git** dentro del desarrollo de este sistema web se basó en el seguimiento y control de todas las versiones del mismo, en este apartado se instala **Git**, para poder descargar el proyecto que contiene todo el código fuente del sistema web para lo cual realizaremos la descarga del programa Git desde el siguiente link <https://git-scm.com/downloads/win> y posterior realizamos los siguientes pasos:

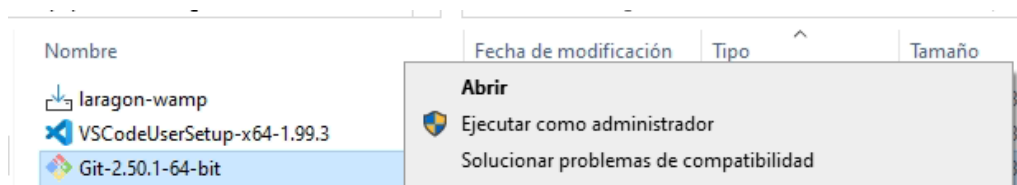


Ilustración 131. Directorio descargas.

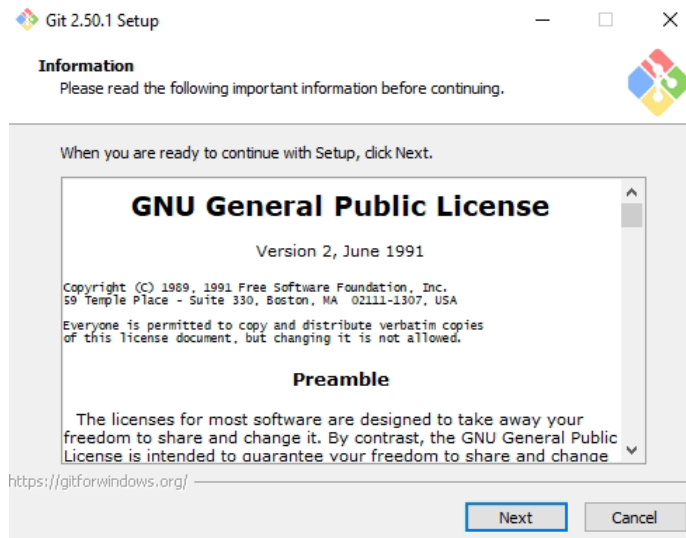


Ilustración 132. Opciones de instalación.

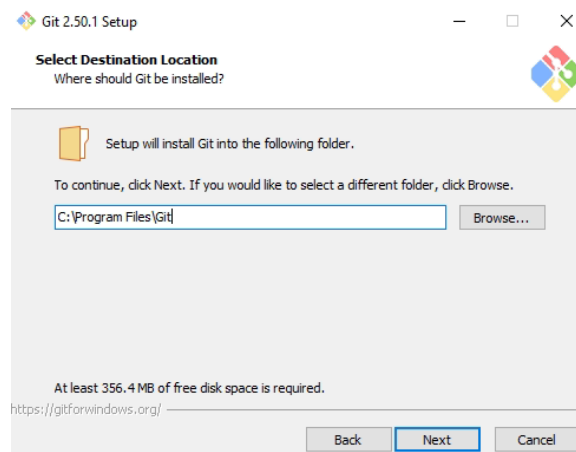


Ilustración 133. Directorio de instalación.

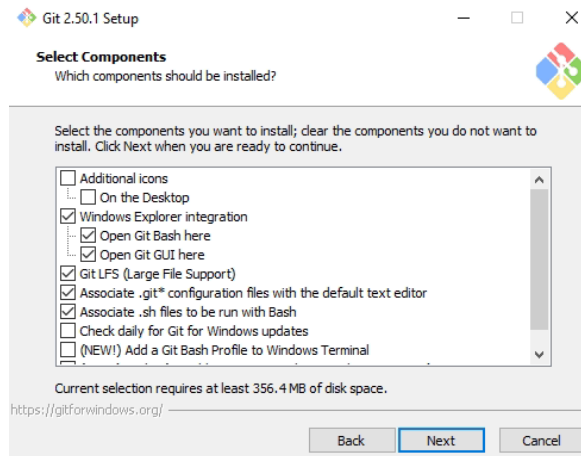


Ilustración 134. Opciones de instalación.

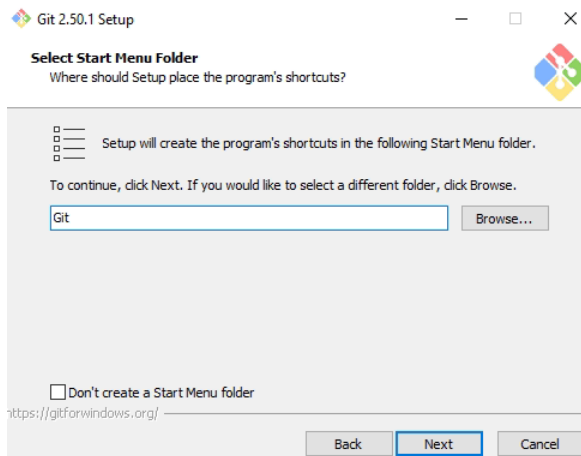


Ilustración 135. Opciones de instalación.

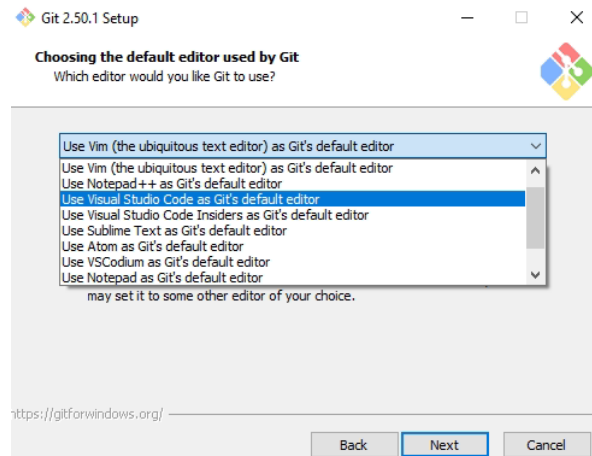


Ilustración 136. Selección de VSC como editor por defecto para Git.

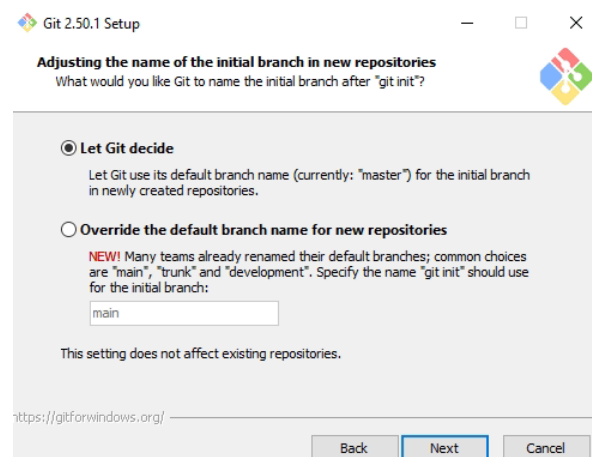


Ilustración 137. Opciones de instalación.

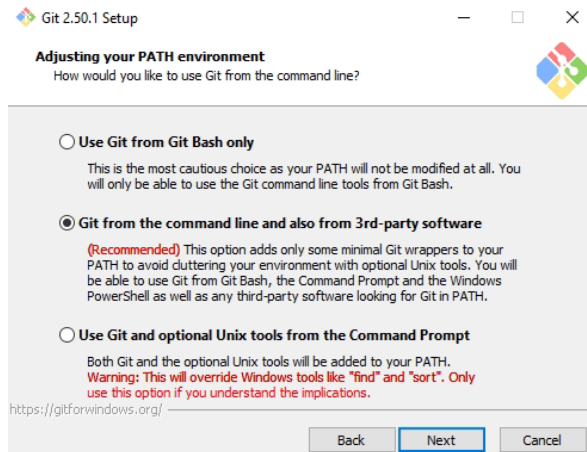


Ilustración 138. Opciones de instalación.

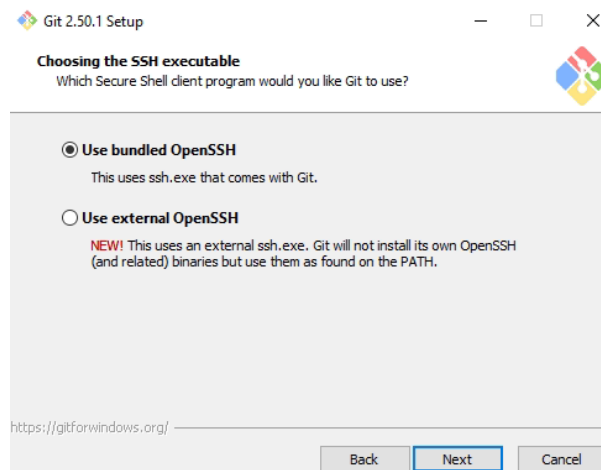


Ilustración 139. Opciones de instalación.

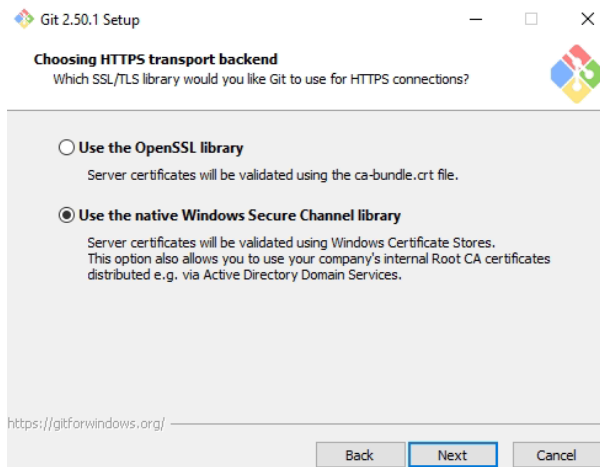


Ilustración 140. Opciones de instalación.

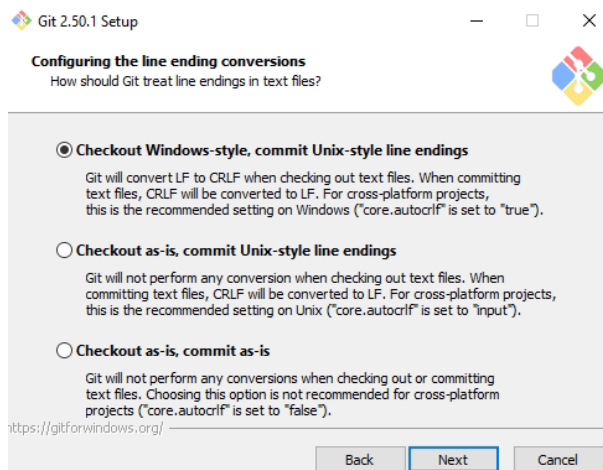


Ilustración 141. Opciones de instalación.

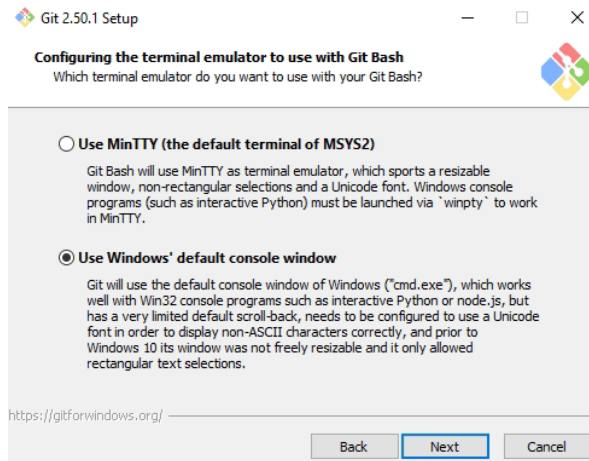


Ilustración 142. Símbolo del sistema como terminal por defecto.

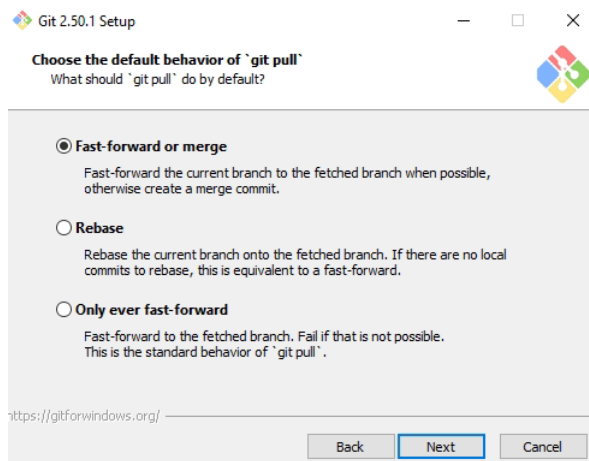


Ilustración 143. Opciones de instalación.

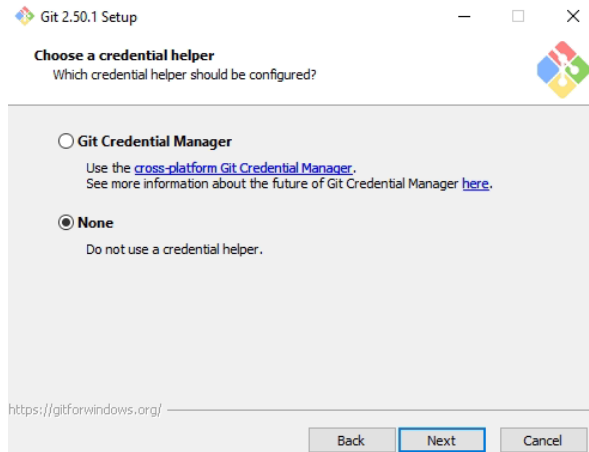


Ilustración 144. Opciones de instalación.

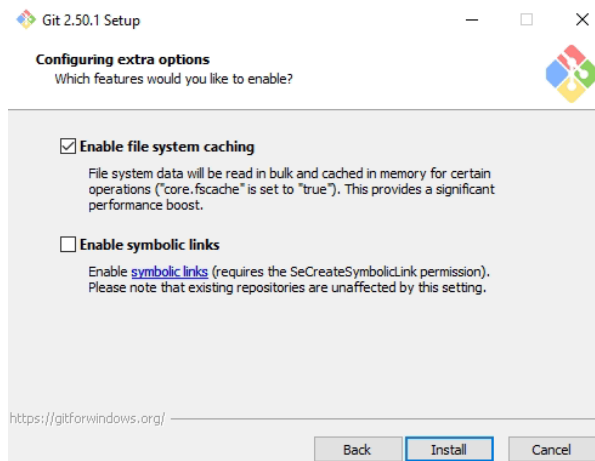


Ilustración 145. Inicio instalación Git.

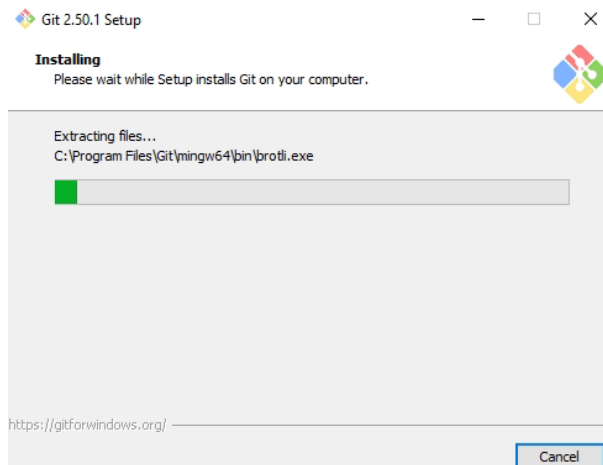


Ilustración 146. Progreso instalación Git.

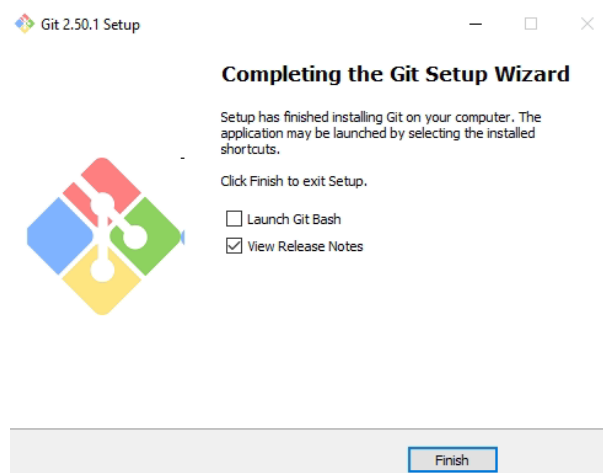


Ilustración 147. Finalización instalación Git.

Instalación del proyecto

Una vez realizadas todas las instalaciones necesarias de software en el sistema operativo, tal como se explica en los apartados anteriores (Laragon, VSC y Git), se procede a realizar la descarga del proyecto.

Clonación del repositorio

Como se mencionó en el apartado anterior, el proyecto y sus versiones fueron controladas en un repositorio de GitHub ubicado en <https://github.com/stalinc2k/proyecto-muestrasmedicasapp>, para lo cual únicamente debemos descargar o clonar el código fuente del proyecto.

Procedemos a crear un directorio en donde almacenaremos la información del proyecto que acabamos de clonar, para esto creamos dentro del siguiente directorio **C:\laragon\www**, una carpeta llamada **muestrasmedicas-app**, como se muestra en la siguiente ilustración:



Ilustración 148. Directorio del proyecto.

Nota 1 - Se debe crear obligatoriamente dentro de este directorio, para que Laragon pueda ejecutar de manera correcta todos los archivos del sistema web.

Dentro de un terminal ya sea de Windows o el propio de Git realizamos lo siguiente:

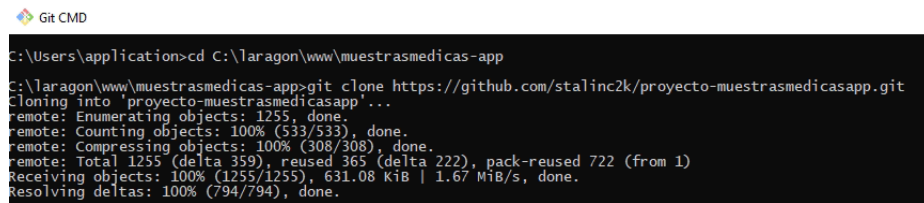
1. Nos ubicamos dentro de la carpeta **muestrasmedicas-app**
(C:\laragon\www\muestrasmedicas-app)



Ilustración 149. Directorio donde se clonará el repositorio.

2. Clonamos el repositorio utilizando el comando

```
git clone https://github.com/stalinc2k/proyecto-muestrasmedicasapp.git
```



```
Git CMD
C:\Users\application>cd C:\laragon\www\muestrasmedicas-app
C:\laragon\www\muestrasmedicas-app>git clone https://github.com/stalinc2k/proyecto-muestrasmedicasapp.git
Cloning into 'proyecto-muestrasmedicasapp'...
remote: Enumerating objects: 1255, done.
remote: Counting objects: 100% (533/533), done.
remote: Compressing objects: 100% (308/308), done.
remote: Total 1255 (delta 359), reused 365 (delta 222), pack-reused 722 (from 1)
Receiving objects: 100% (1255/1255), 631.08 KiB | 1.67 MiB/s, done.
Resolving deltas: 100% (794/794), done.
```

Ilustración 150. Proceso de clonación del repositorio.

3. Una vez finalizada la conación del repositorio podemos visualizar que todos los archivos del proyecto se encuentran descargados.

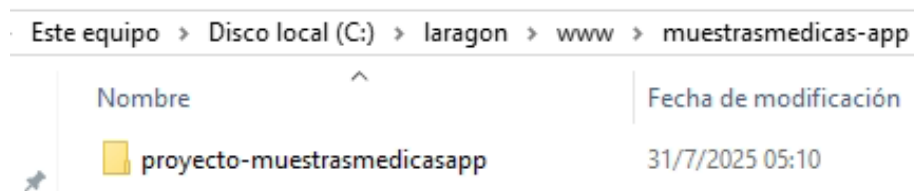


Ilustración 151. Directorio con el proyecto clonado.

Abrir el proyecto con VS Code

Ya clonado el proyecto en el computador dentro del directorio antes mencionado únicamente nos queda abrirlo en el editor VS Code para poder visualizar, editar o manipular el código fuente de cada uno de los módulos del sistema web **“Control y Gestión Muestras Médicas”**. Para esto realizaremos los siguientes pasos:

1. Abrimos VS Code
2. Nos dirigimos al menú File y seleccionamos Open Folder

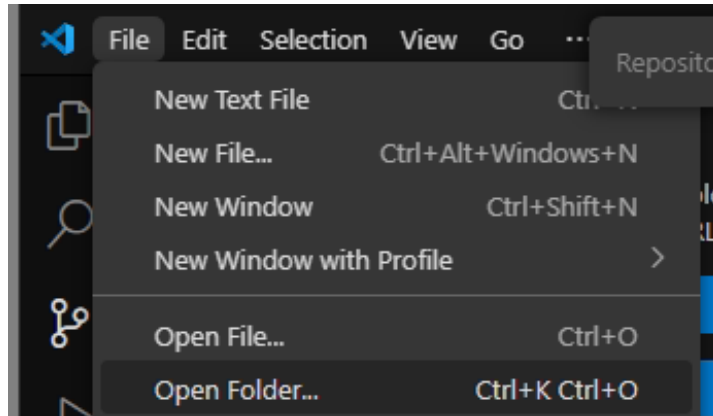


Ilustración 152. Abrir Proyecto.

3. Seleccionamos la carpeta donde se encuentra descargado el proyecto.

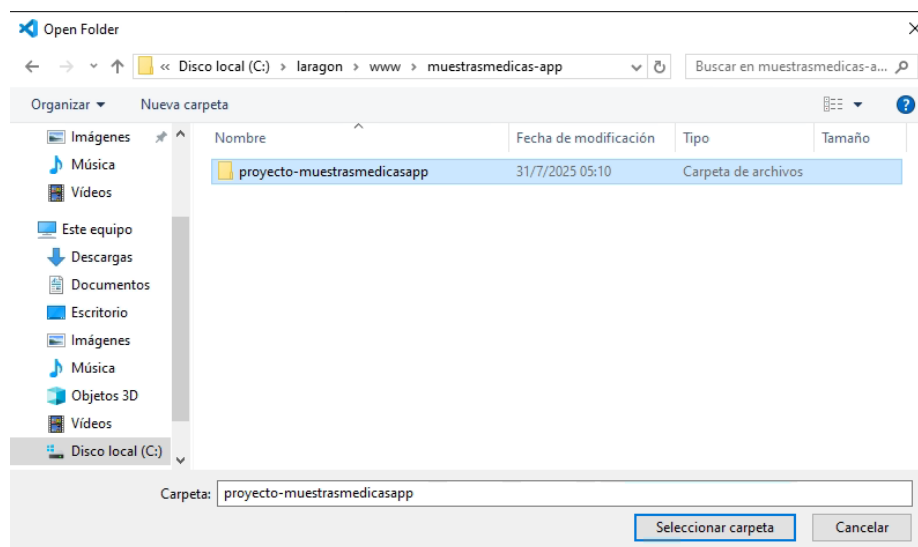


Ilustración 153. Seleccionar ubicación del proyecto.

4. Para saber que todo el proyecto se cargó correctamente en VSC debemos poder visualizar la estructura de archivos tal como se muestra en la siguiente ilustración:

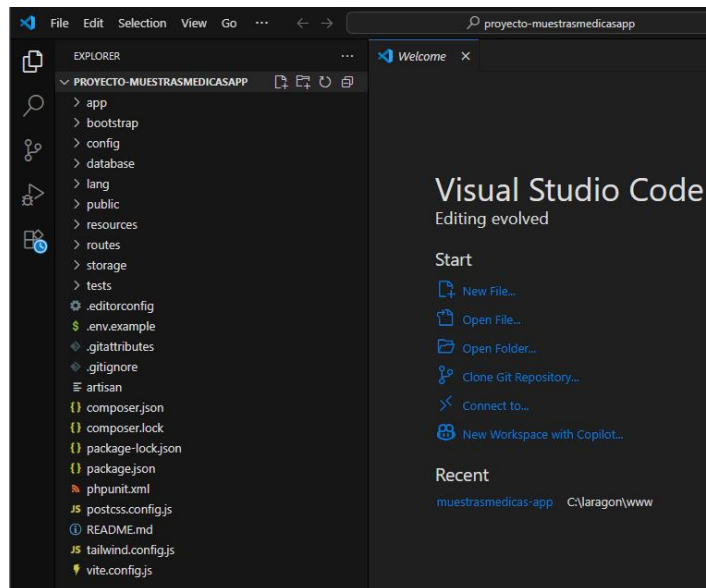


Ilustración 154. Estructura de archivos del proyecto.

Creación del archivo de configuración

El archivo de configuración del proyecto es el que contiene todas las variables de entorno, este archivo es fundamental para que el sistema funcione correctamente, a continuación, se detallan los pasos para crear dicho archivo:

Crea el archivo.env en la raíz del proyecto

- Ubícate en la raíz (nivel principal de tu carpeta de proyecto *muestrasmedicas-app*)
- Haz clic derecho → Nuevo archivo.

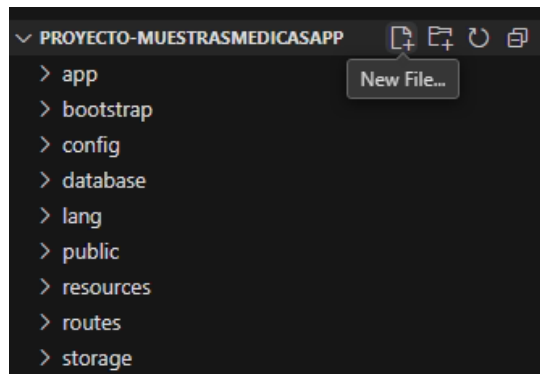


Ilustración 155. Crear archivo .env

- Nómbralo exactamente así **.env** (sin nombre, solo extensión)



Ilustración 156. Archivo .env

Agregar variables de entorno

Dentro el archivo de configuración ingresaremos el siguiente código con el cual se establecerá toda la configuración global del sistema web.

```
APP_NAME="Sistema Web C.G.M.M"
APP_ENV=local
APP_KEY=base64:k27KO60PFQmkG63In4RnfK3gwUrTh26ewU6q32HV3X4=
APP_DEBUG=true
APP_URL=http://localhost

APP_LOCALE=en
APP_FALLBACK_LOCALE=en
APP_FAKER_LOCALE=en_US

APP_MAINTENANCE_DRIVER=file
# APP_MAINTENANCE_STORE=database

PHP_CLI_SERVER_WORKERS=4

BCRYPT_ROUNDS=12

LOG_CHANNEL=stack
```

```
LOG_STACK=single
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

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

SESSION_DRIVER=database
SESSION_LIFETIME=120
SESSION_ENCRYPT=false
SESSION_PATH=/
SESSION_DOMAIN=null

BROADCAST_CONNECTION=log
FILESYSTEM_DISK=local
QUEUE_CONNECTION=database

CACHE_STORE=database
# CACHE_PREFIX=

MEMCACHED_HOST=127.0.0.1

REDIS_CLIENT=phpredis
REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=log
MAIL_SCHEME=null
MAIL_HOST=127.0.0.1
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=
AWS_USE_PATH_STYLE_ENDPOINT=false
```

```
VITE_APP_NAME="${APP_NAME}"
```

Instalación de dependencias

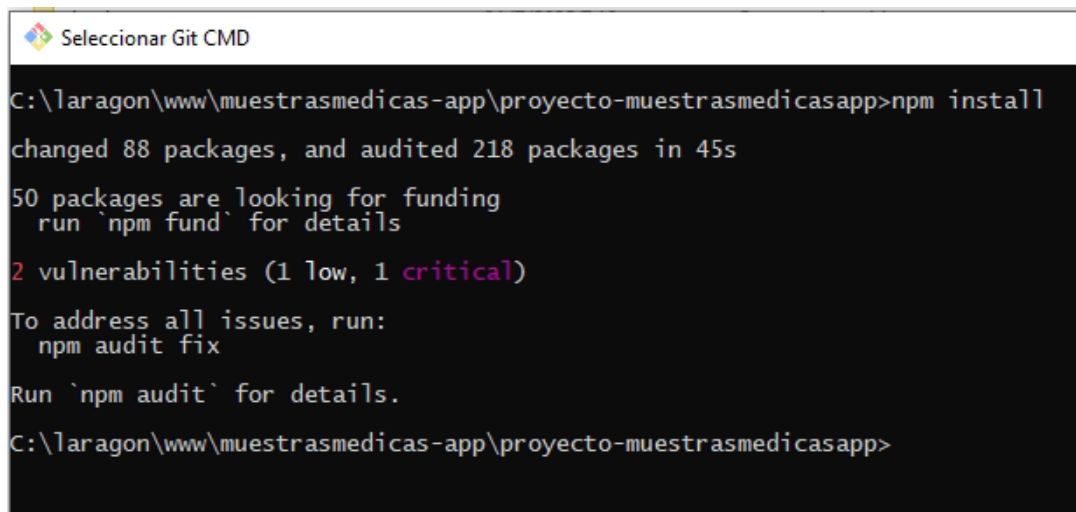
Ya se ha procedido a configurar el archivo .env del sistema, ahora se debe proceder a instalar todas las dependencias que usa el proyecto, para el momento en que se está realizando este manual, el sistema está desarrollado con dependencias modernas como son:

- Vite como bundler
- TailwindCSS para estilos
- AlpineJS y Flowbite para componentes JS/UI

Cada una de las dependencias antes mencionadas deben ser instaladas por comandos mediante el uso del terminal, con el terminal inicializado nos ubicamos en la raíz del proyecto *C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp*, ya dentro de la raíz ejecutamos los siguientes comandos npm install (instala las dependencias usan Node.js):

```
PS C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp> npm install
```

Ilustración 157. Instalación Node.js



```
C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp>npm install
changed 88 packages, and audited 218 packages in 45s

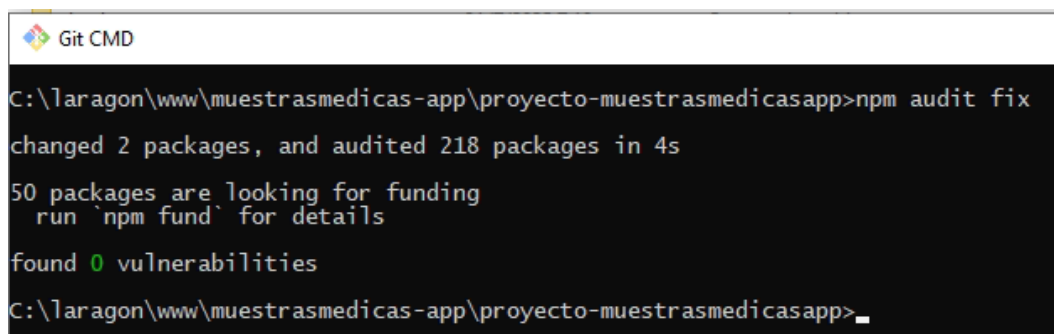
50 packages are looking for funding
  run `npm fund` for details

2 vulnerabilities (1 low, 1 critical)
To address all issues, run:
  npm audit fix

Run `npm audit` for details.
C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp>
```

Ilustración 158. Instalación terminada.

La ilustración 158 nos indica que se instalaron 88 paquetes y se auditaron 218, adicional nos presenta 2 vulnerabilidades (paquetes desactualizados), para poder corregir estas advertencias de vulnerabilidad realizamos lo siguiente (usamos el comando npm audit fix para poder actualizar los paquetes):



```
C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp>npm audit fix
changed 2 packages, and audited 218 packages in 4s

50 packages are looking for funding
  run `npm fund` for details

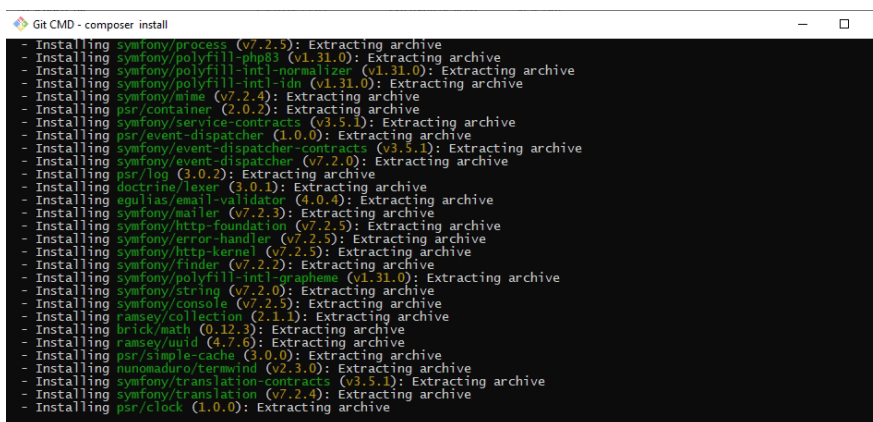
found 0 vulnerabilities
C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp>_
```

Ilustración 159. Corrección de vulnerabilidades en dependencias.

Nota 2 - Si se corrigieron correctamente las vulnerabilidades debemos visualizar en la consola el mensaje de 0 vulnerabilidades encontradas, este mensaje aparece al final de la ejecución del comando.

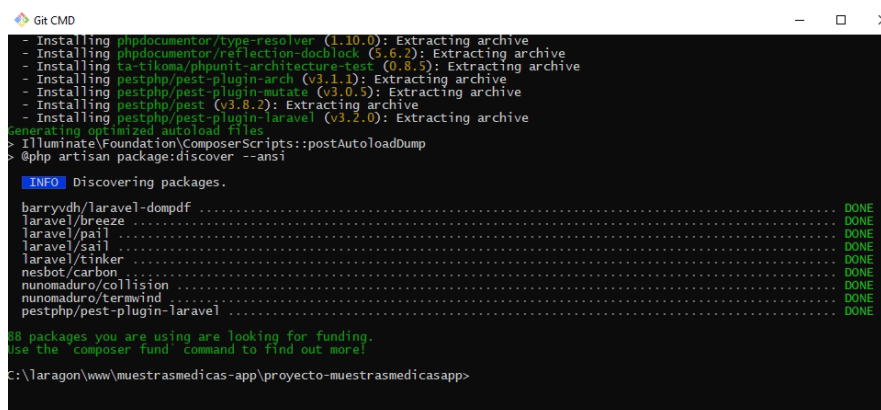
Instalación de composer

Composer es un gestor de dependencias para proyectos PHP, hay que tomar en cuenta que este proyecto es desarrollado con el Framework Laravel, este usa lenguaje de programación PHP, por lo tanto, se necesita instalar composer dentro del proyecto (el instalador se descarga automáticamente cuando se instaló Laragon en el inicio de este manual), entonces dentro del terminal nos ubicado en la raíz del proyecto y digitamos el siguiente comando ***composer install***.



```
Git CMD - composer install
- Installing symfony/process (v7.2.5): Extracting archive
- Installing symfony/polyfill-php83 (v1.31.0): Extracting archive
- Installing symfony/polyfill-intl-normalizer (v1.31.0): Extracting archive
- Installing symfony/polyfill-intl-idn (v1.31.0): Extracting archive
- Installing symfony/mime (v7.2.4): Extracting archive
- Installing psr/container (2.0.2): Extracting archive
- Installing symfony/service-contracts (v3.5.1): Extracting archive
- Installing psr/event-dispatcher (1.0.0): Extracting archive
- Installing symfony/event-dispatcher-contracts (v3.5.1): Extracting archive
- Installing symfony/event-dispatcher (v7.2.0): Extracting archive
- Installing psr/log (3.0.2): Extracting archive
- Installing doctrine/lexer (3.0.1): Extracting archive
- Installing egulias/email-validator (4.0.4): Extracting archive
- Installing symfony/mailer (v7.2.3): Extracting archive
- Installing symfony/http-foundation (v7.2.5): Extracting archive
- Installing symfony/error-handler (v7.2.5): Extracting archive
- Installing symfony/http-kernel (v7.2.5): Extracting archive
- Installing symfony/finder (v7.2.2): Extracting archive
- Installing symfony/polyfill-intl-grapheme (v1.31.0): Extracting archive
- Installing symfony/string (v7.2.0): Extracting archive
- Installing symfony/console (v7.2.5): Extracting archive
- Installing ramsey/collection (2.1.1): Extracting archive
- Installing brick/math (0.12.3): Extracting archive
- Installing ramsey/uuid (4.7.6): Extracting archive
- Installing psr/simple-cache (3.0.0): Extracting archive
- Installing nunomaduro/termwind (v2.3.0): Extracting archive
- Installing symfony/translation-contracts (v3.5.1): Extracting archive
- Installing symfony/translation (v7.2.4): Extracting archive
- Installing psr/clock (1.0.0): Extracting archive
```

Ilustración 160. Instalación de composer.



```
Git CMD
- Installing phpdocumentor/type-resolver (1.10.0): Extracting archive
- Installing phpdocumentor/reflection-docblock (5.6.2): Extracting archive
- Installing ektikoma/phpunit-architecture-test (0.8.3): Extracting archive
- Installing pestphp/pest-plugin-arch (v3.1.1): Extracting archive
- Installing pestphp/pest-plugin-mutate (v3.0.5): Extracting archive
- Installing pestphp/pest (v3.8.2): Extracting archive
- Installing pestphp/pest-plugin-laravel (v3.2.0): Extracting archive
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

barryvdh/laravel-dumpdf ..... DONE
laravel/breeze ..... DONE
laravel/pail ..... DONE
laravel/sail ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
pestphp/pest-plugin-laravel ..... DONE

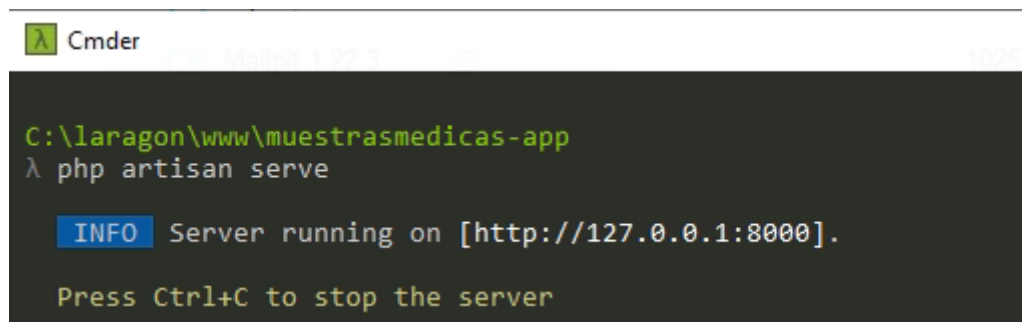
88 packages you are using are looking for funding.
Use the "composer fund" command to find out more!

C:\laragon\www\muestrasmedicas-app\proyecto-muestrasmedicasapp>
```

Ilustración 161. Instalación terminada composer.

Una vez terminada la instalación de ***Composer***, vamos a comprobar si nuestro servidor en laravel está funcionando correctamente y para esto dentro del terminal ubicados en la carpeta

raíz del proyecto digitamos los siguientes comandos *php artisan serve*, como se muestra en la siguiente ilustración:



```
λ Cmder
C:\laragon\www\muestrasmedicas-app
λ php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

Ilustración 162. Ejecutar Servidor.

Como se puede observar en la ilustración 162, nuestro proyecto se está ejecutando dentro del servidor ***http://127.0.0.1:8000***, adicional se observa que no presenta ningún mensaje de error en el terminar por lo tanto procedemos a inicializar los servicios de laragon para que el servidor pueda ejecutar todo el código fuente del proyecto de manera correcta.

Inicializar servicios de Laragon

Para un correcto funcionamiento del sistema web, se necesita ejecutar estos dos servicios fundamentales que se instalaron en conjunto con Laragon los cuales se detalla a continuación:

- Apache 2.4.62
- MySQL 8.4.3

Cada uno cumple un papel fundamental dentro del proyecto, es decir, Apache ejecuta todo el código fuente realizado en php y MySQL se encarga del manejo de la base de datos del sistema de muestras médicas.

Entonces una vez explicado los servicios necesarios procedemos a seguir los pasos detallados en las ilustraciones que se muestran a continuación:

En el buscador de Windows digitamos laragon y ejecutamos el aplicativo que ya se encuentra instalado.

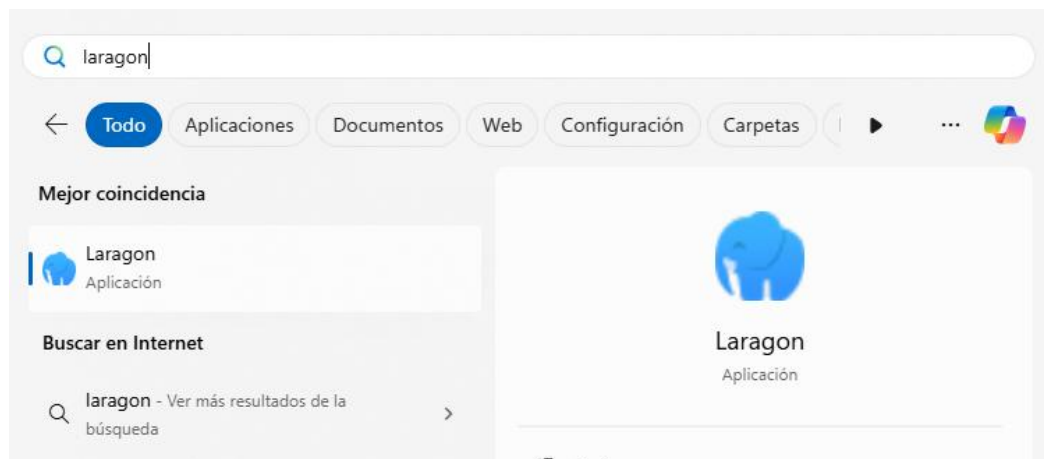


Ilustración 163. Búsqueda de Laragon.

Una vez dentro de la interfaz de Laragon seleccionamos o damos clic en el botón iniciar todo, esto habilitara los servicios antes mencionados.

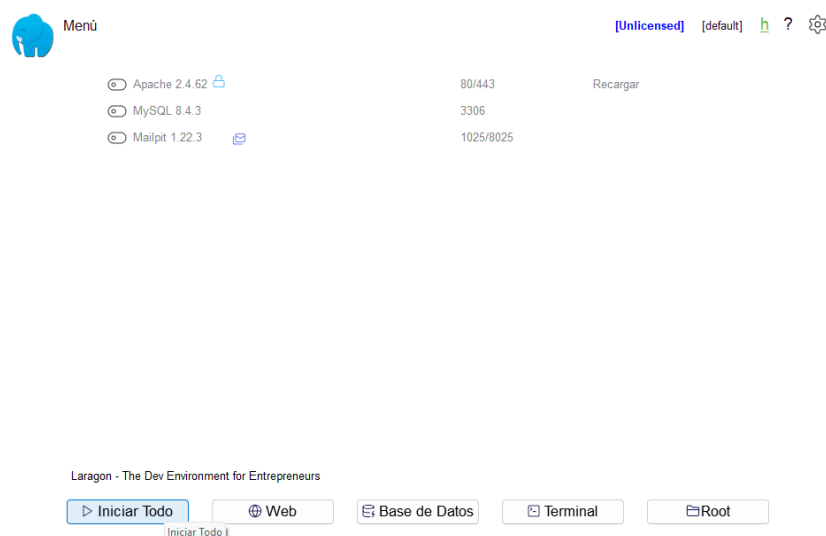


Ilustración 164. Activar servicios.

Por último, comprobamos que los servicios necesarios se estén ejecutando, y para esto únicamente visualizaremos si los toolbars que se encuentran alado del nombre del servicio cambiaron de color gris a verde.

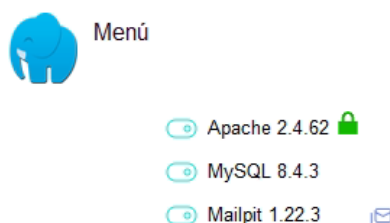


Ilustración 165. Verificación de servicios.

Ya con los servicios y el servidor inicializados, se debe comprobar que nuestro sistema web se esté ejecutando correctamente, es decir ya nos presente la interfaz gráfica de los módulos de inicio de sesión para los usuario o administradores del sistema, para esto ingresamos a uno de los navegadores que soporta el sistema (Mozilla, Firefox, navegadores actuales a la fecha de este manual), y en la barra de direcciones accederemos al sistema con cualquiera de las direcciones (<http://127.0.0.1:8000/> o <http://muestrasmedicas-app.test/>), y nos debe presentar la interfaz inicial del sistema como se muestra en las siguientes ilustraciones:

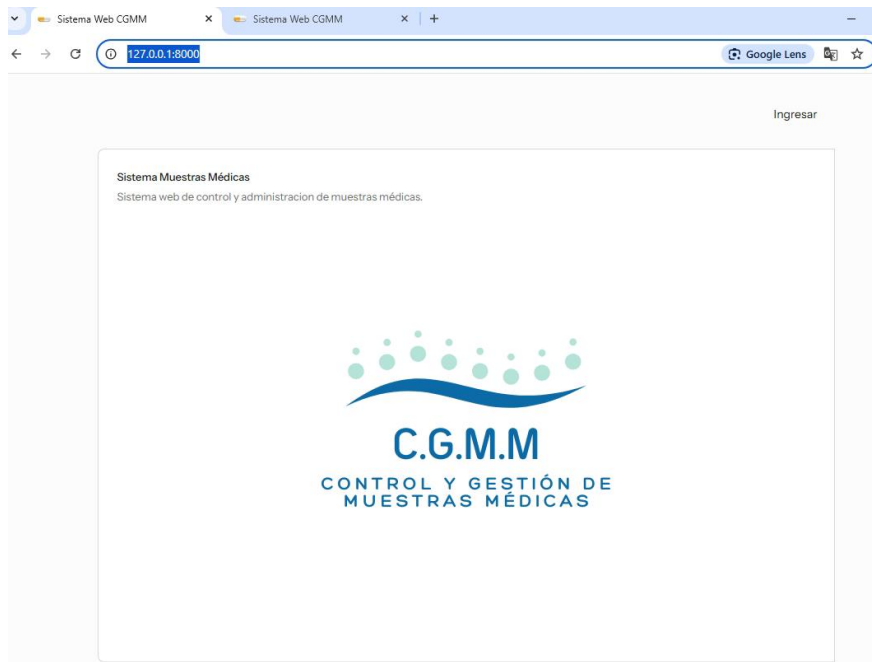


Ilustración 166. Dirección 127.0.0.1:8000

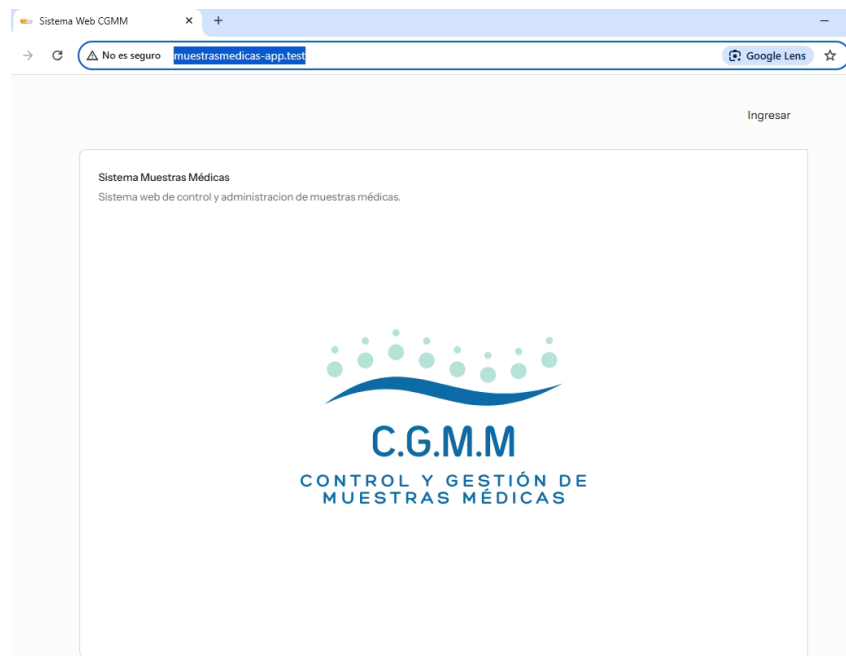


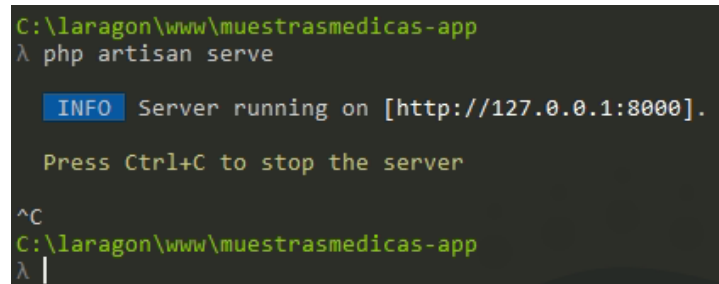
Ilustración 167. Dirección muestrasmedicas-app.test.

Si llegamos hasta este punto podemos deducir que la instalación del sistema web de control muestras médicas ha sido levantado correctamente.

Creación de la base de datos

En este punto hemos configurado e instalado el proyecto correctamente, se puede ingresar a la interfaz de inicio de sesión y ver correctamente el diseño de la misma, ahora para poder realizar todas las transacciones dentro del sistema debemos crear la base de datos que se encargara de manejar y almacenar toda la información que se genere y para esto debemos ejecutar las migraciones y seeders que se encuentran codificadas dentro del proyecto, para esto ingresamos al terminal y realizamos estos pasos:

Cancelamos la ejecución del servidor utilizando **Ctrl + C**



```
C:\laragon\www\muestrasmedicas-app
λ php artisan serve

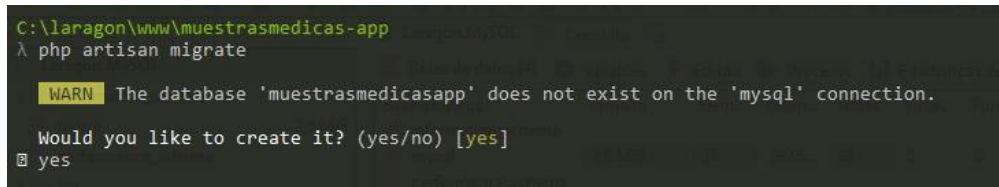
INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

^C
C:\laragon\www\muestrasmedicas-app
λ |
```

Ilustración 168. Apagar servidor.

Verificamos si seguimos en la carpeta raíz del proyecto y digitamos el siguiente comando **php artisan migrate**. Esto creara la base de datos y sus tablas, incluidas las relaciones entre cada una, pero antes de esto preguntara si deseamos crear la BD colocamos la palabra yes para confirmar.



```
C:\laragon\www\muestrasmedicas-app
λ php artisan migrate

WARN The database 'muestrasmedicasapp' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [yes]
yes
```

Ilustración 169. Creación base de datos.

Si se crearon correctamente todas las tablas de la base de datos incluidas las relaciones, se podrá visualizar en el terminal el proceso finalizado tal como se indica en la ilustración.

```
C:\laragon\www\muestrasmedicas-app
php artisan migrate

[WARN] The database 'muestrasmedicasapp' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [yes]
yes

[INFO] Preparing database.

Creating migration table ..... 136.83ms DONE

[INFO] Running migrations.

0001_01_01_000000_create_users_table ..... 783.12ms DONE
0001_01_01_000001_create_cache_table ..... 248.55ms DONE
0001_01_01_000002_create_jobs_table ..... 556.69ms DONE
2025_04_30_165843_create_companies_table ..... 554.77ms DONE
2025_04_30_174302_create_zones_table ..... 1s DONE
2025_05_01_000003_create_products_table ..... 1s DONE
2025_05_01_000004_create_batches_table ..... 867.67ms DONE
2025_05_01_163637_create_incomes_table ..... 947.98ms DONE
2025_05_01_164631_create_expenses_table ..... 981.13ms DONE
2025_05_01_164841_create_inventories_table ..... 3s DONE

C:\laragon\www\muestrasmedicas-app
```

Ilustración 170. Base de datos creada.

Creación del administrador del sistema

Con la base de datos creada como último paso debemos crear el usuario que va administrar el sistema este usuario constara de los siguientes datos:

- email: admin@admin.com
- name: admin
- lastname: administrador
- password: admin1234

no habrá necesidad de codificar nada porque dentro del proyecto del sistema disponemos de un seeder que contiene el código el cual creara directamente en la base de datos el usuario administrador y para esto en el mismo terminar procedemos a digitar los siguientes comandos **php artisan db:seed**, a continuación, se muestra el resultado de la ejecución de la línea de comando.

```
C:\laragon\www\muestrasmedicas-app
^ php artisan db:seed

INFO Seeding database.

Database\Seeders\AdminUserSeeder ..... RUNNING
Database\Seeders\AdminUserSeeder ..... 449 ms DONE

C:\laragon\www\muestrasmedicas-app
```

Ilustración 171. Creación administrador del sistema.

Si en el terminal podemos observar que se creó sin ninguna complicación, debemos revisar la base de datos para poder confirmar que el usuario administrador se encuentra correctamente creado y para esto nos dirigimos a *Laragon*, y seleccionamos la opción base de datos.

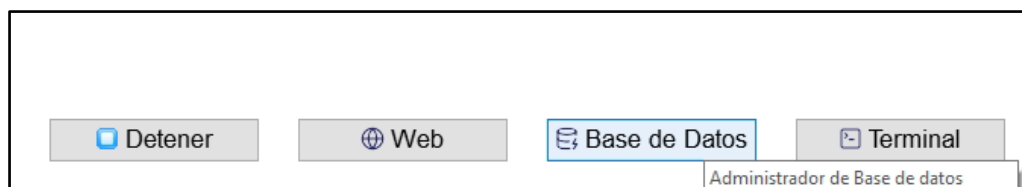


Ilustración 172. Opción base de datos.

Esto nos abrirá el administrador de la base de datos para este caso MySQL, y debemos seleccionar la DB creada (muestrasmedicasapp).

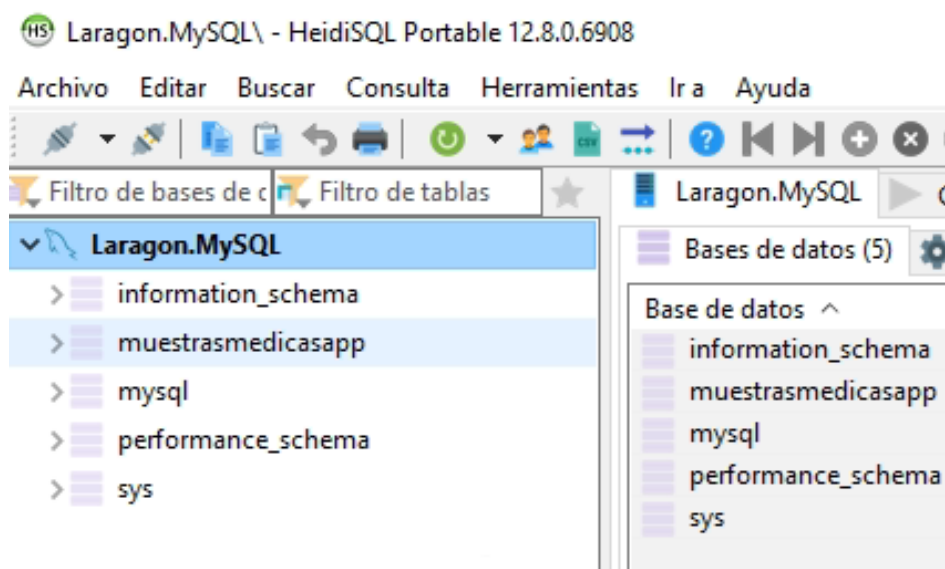
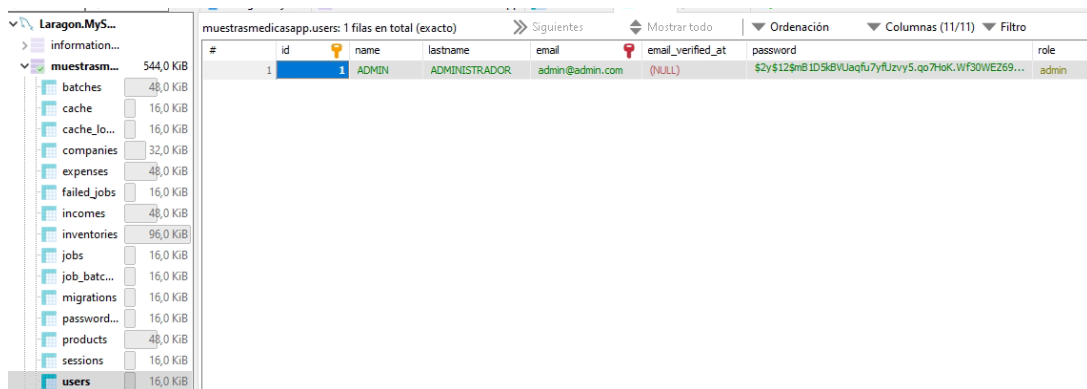


Ilustración 173. Administrador BD.

Ingresamos a la BD y revisamos la tabla users.



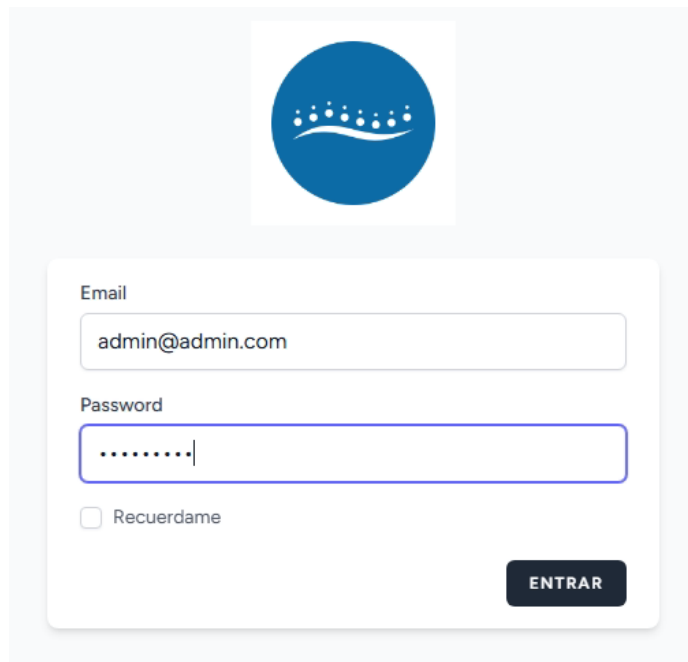
#	id	name	lastname	email	email_verified_at	password	role
1	1	ADMIN	ADMINISTRADOR	admin@admin.com	(NULL)	\$2y\$12\$m8ID5k8VUaqfu7yflZvry5.qo7HoK.Wf30WEZ69...	admin

Como podemos observar, se encuentra creado el usuario admin con los datos que describimos en el apartado anterior, ahora volvemos a ejecutar nuestro servidor con los comandos *php artisan serve* y nos dirigimos al navegador, escribimos la dirección del sistema web (muestrasmedicas-app.test) y damos clic en la opción ingresar.



Ilustración 174. Ingreso al sistema como administrador.

Luego en la interfaz de Login ingresamos los datos del administrador (email y password).



Email

admin@admin.com

Password

.....

☐ Recuerdame

ENTRAR

Ilustración 175. Login.

Por último, si todo nuestro levantamiento del sistema fue correcto, el sistema nos dirigirá al DASHBOARD principal del sistema.

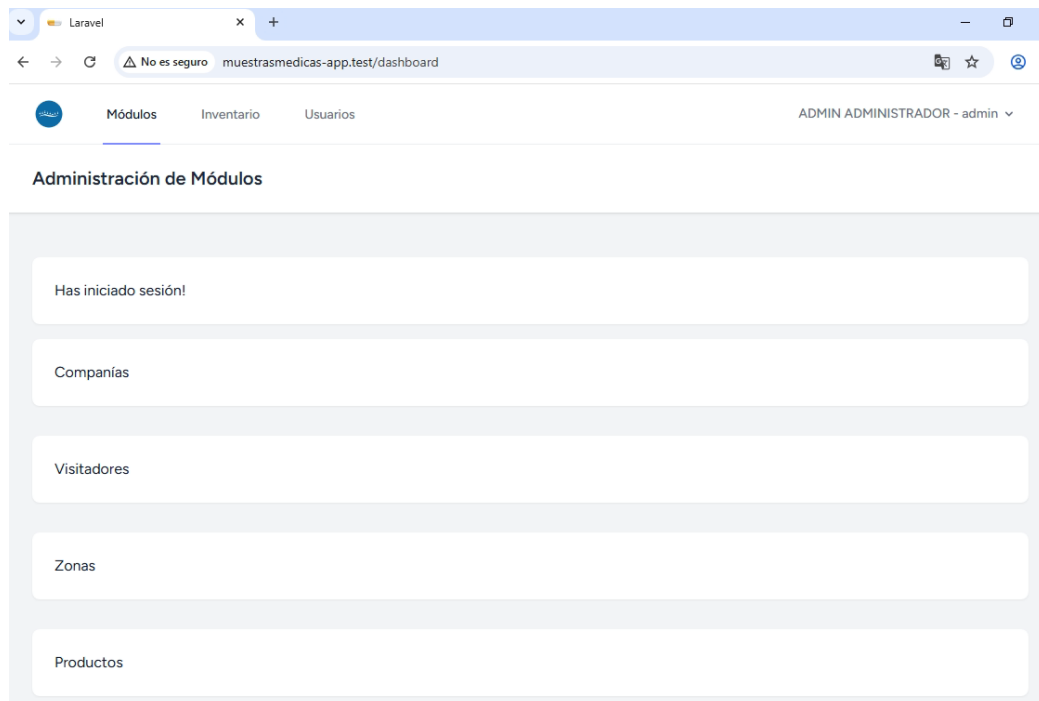


Ilustración 176. Interfaz principal del sistema.

Código fuente Módulo Proveedores

A continuación, se explicará el funcionamiento interno del módulo de Proveedores del sistema con el fin de que se pueda tener conocimiento acerca del código fuente del mismo.

Interfaz principal Proveedores

ADMINISTRACIÓN PROVEEDORES						
Buscar...						
CÓDIGO	RAZÓN SOCIAL/NOMBRE	DIRECCIÓN	TELÉFONO	TIPO	CREADO POR	ACCIONES
P1719307967001 Ruc 1719307967001	DENTAL AYALA	AMERICA Y COLON SN 25	0983443313	Proveedores	STALIN REASCOS 2025-06-17 15:17:22	Editar Eliminar
P1719307975001 Ruc 1719307975001	DENTAL ALTAMIRANO	COLON Y MERCADILLO	0000000000	Proveedores	STALIN REASCOS 2025-06-17 15:39:41	Editar Eliminar
P1709217457001 Ruc 1709217457001	DENTAL DENTSPLAY	AMERICA 27-29 Y LAS CASAS	0998403968	Proveedores	STALIN REASCOS 2025-06-17 15:41:21	Editar Eliminar
P1719307967002 Ruc 1719307967001	DENTAL AYALA 2		0000000000	Proveedores	STALIN REASCOS 2025-06-17 16:48:33	Editar Eliminar

Ilustración 177. Interfaz principal Proveedores.

Código fuente

Para visualizar y manipular el código fuente de la interfaz principal de proveedores (index.blade.php), nos ubicamos en el siguiente directorio dentro de VSC.

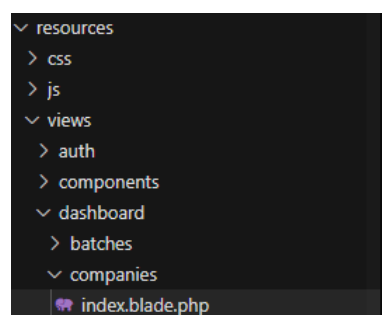


Ilustración 178. Views Companies.


```

<tr
  class="bg-white border-b dark:bg-gray-800 dark:border-gray-700 border-gray-
200 hover:bg-gray-50 dark:hover:bg-gray-600">
  <th scope="row"
    class="flex items-center px-6 py-4 text-gray-900 whitespace-nowrap dark:text-
white">
    <div class="ps-1">
      <div class="text-base font-semibold">{{ $company->code }}</div>
      <div class="font-normal text-gray-500">Ruc {{ $company->ruc }}</div>
    </div>
  </th>
  <td class="px-6 py-4">
    {{ $company->name }}
  </td>
  <td class="px-6 py-4">
    {{ $company->address }}
  </td>
  <td class="px-6 py-4">
    {{ $company->phone ?? '0000000000' }}
  </td>
  <td class="px-6 py-4">
    @if($company->type == 'major')
      {{ 'Empresa Principal' }}
    @else
      {{ 'Proveedores' }}
    @endif
  </td>
  <td class="px-6 py-4">
    {{ $company->user->name }} {{ $company->user->lastname }}
    <div class="font-normal text-gray-500">{{ $company->created_at }}</div>
  </td>
  <td class="px-2 py-2 justify-between">
    @can('update', $company)
      <x-companycomponents.modal-edit-company :companyId="$company-
>id" :company="$company" />
    @endcan
    @can('delete', $company)
      <x-companycomponents.modal-delete-company :companyId="$company-
>id" :company="$company" />
    @endcan
  </td>
</tr>
</tbody>
</table>
</div>

```

```

        class="flex items-center justify-center flex-column flex-wrap md:flex-row space-y-4
md:space-y-0 pb-4 bg-white dark:bg-gray-900">
        <div class="mt-3">
            {{ $companies->links() }}
        </div>
    </div>
</div>
<x-companycomponents.errormodal-open-company />
@endsection

```

Interfaz crear proveedor

Interfaz donde el usuario ingresa los datos del proveedor que desea registrar en el sistema (modal-new-company.blade.php).

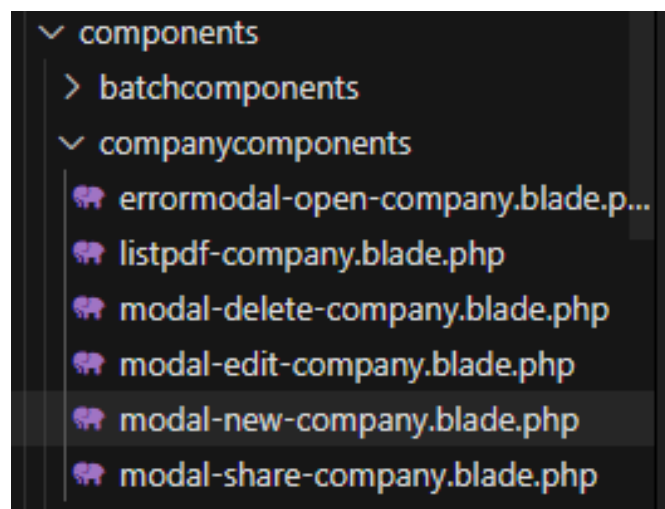


Ilustración 179. Directorio de componentes Company.

Nota 3 - Todos los componentes que veremos a continuación para el módulo de proveedores estarán en este directorio.

Ilustración 180. Interfaz crear proveedor.

Código fuente

```
<div>
  <button onclick="mostrarModal('companyModal')" type="button"
    class="text-blue-700 border mt-1 hover:bg-blue-700 hover:text-white focus:ring-4
focus:outline-none focus:ring-blue-300 font-sm rounded-lg text-sm p-2.5 text-center inline-flex
items-center me-2 dark:border-blue-500 dark:text-blue-500 dark:hover:text-white
dark:focus:ring-blue-800 dark:hover:bg-blue-500 cursor-pointer">
    <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-
width="1.5" stroke="currentColor"
      class="size-6">
      <path stroke-linecap="round" stroke-linejoin="round" d="M12 9v6m3-3H9m12 0a9 9
0 1 1-18 0 9 9 0 1 1 18 0Z" />
    </svg>
  </button>
  <div id="companyModal" class="fixed inset-0 bg-gray-900 bg-opacity-50 items-center
justify-center hidden z-50 overflow-y-auto">
    <div class="bg-white p-6 rounded-xl shadow-lg w-96">
      <h2 class="text-xl font-bold mb-4">Nuevo Proveedor</h2>
      @include('fragment._errors-form')
      <form id="formcompanyNew" action="{{ route('company.store') }}"
method="POST">
        @csrf
        <div class="grid gap-4 mb-4 grid-cols-2">
          <div class="col-span-2">
            <label for="code"
              class="block mb-2 text-sm font-medium text-gray-900 dark:text-white">*
Código</label>
            <input type="text" name="code" value="{{ old('code') }}" id="code"
              class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
```

```

        dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
        placeholder="Example P(RUC) - P000000000000" maxlength="14"
title="Debe empezar con P seguido del RUC o CI" required>
    </div>

    <div class="col-span-2">
        <label for="ruc"
            class="block mb-2 text-sm font-medium text-gray-900 dark:text-white">*
Ruc</label>
        <input type="text" name="ruc" value="{{ old('ruc') }}" id="ruc"
            class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
            dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
            placeholder="Example 151713681931001" title="Unicamente dígitos
numéricos" minlength="10" maxlength="13" required>
    </div>

    <div class="col-span-2">
        <label for="name" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">* Razón
Social</label>
        <input type="text" id="name" name="name" value="{{ old('name') }}"
            class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
            dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
            placeholder="Nombre o Razon Social" minlength="5" maxlength="150"
required>
    </div>

    <div class="col-span-2">
        <label for="address"
            class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Dirección</label>
        <input type="text" id="address" name="address" value="{{ old('address') }}"
            class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
            dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
            placeholder="Dirección compania" minlength="5" maxlength="150">
    </div>

    <div class="col-span-2">
        <label for="phone"

```

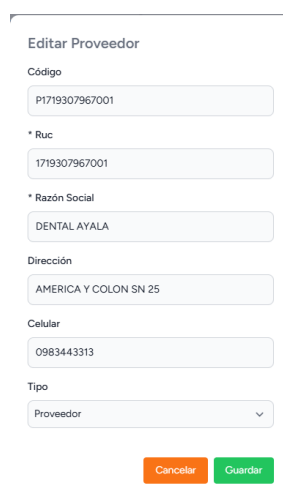
```

        class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Celular</label>
        <input type="number" id="phone" name="phone" value="{{ old('phone') }}"
        class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
        dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
        placeholder="Número celular" minlength="10" maxlength="13">
    </div>
</div>
<div class="flex justify-end gap-2">
    <a onclick="cerrarModal('companyModal')"
        class="bg-orange-500 text-white px-4 py-2 rounded cursor-
pointer">Cancelar</a>
        <button type="submit" class="bg-green-500 text-white px-4 py-2
rounded">Guardar</button>
    </div>
</form>
</div>
</div>
</div>

```

Interfaz editar proveedor

Esta interfaz se encarga de actualizar los datos del proveedor (modal-edit-company.blade.php).



Editar Proveedor

Código
P1719307967001

* Ruc
1719307967001

* Razón Social
DENTAL AYALA

Dirección
AMERICA Y COLON SN 25

Celular
0983443313

Tipo
Proveedor

Cancelar Guardar

Ilustración 181. Interfaz editar proveedor.

Código fuente

```
@props(['companyId','company'])

<a onclick="mostrarModal('companyModalEdit-{{$companyId}})"
  class="text-blue-700 hover:bg-blue-700 hover:text-white focus:ring-4 focus:outline-none
focus:ring-blue-300 font-sm rounded-lg text-sm p-2.5 text-center inline-flex items-center me-2
dark:border-blue-500 dark:text-blue-500 dark:hover:text-white dark:focus:ring-blue-800
dark:hover:bg-blue-500 cursor-pointer">Editar</a>
  <div id="companyModalEdit-{{$companyId}}" class="fixed inset-0 min-h-full bg-gray-900
bg-opacity-50 items-center justify-center hidden z-50 overflow-y-auto">
    <div class="bg-white p-6 rounded-xl shadow-lg w-96">
      <h2 class="text-xl font-bold mb-4">Editar Proveedor</h2>
      @include('fragment._errors-form')
      <form id="formCompanyEdit-{{$companyId}}"
action="{{route('company.update',$company)}}" method="POST" >
        @csrf
        @method('PATCH')
        <div class="grid gap-4 mb-4 grid-cols-2">
          <div class="col-span-2">
            <label for="code" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">Código</label>
            <input type="text" name="code" value="{{old('code',$company->code)}}"
id="code"
              class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
              dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
              placeholder="Example P(RUC) - P000000000000" maxlength="14"
readonly title="Debe empezar con P seguido del RUC o CI">
            </div>

            <div class="col-span-2">
              <label for="ruc" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">* Ruc</label>
              <input type="text" name="ruc" value="{{old('ruc',$company->ruc)}}" id="ruc"
              class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
              dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
              placeholder="Example 151713681931001" title="Unicamente dígitos
numéricos" minlength="10" maxlength="13" required>
              </div>

          <div class="col-span-2">
```

```

        <label for="name" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">* Razón Social</label>
        <input type="text" id="name" name="name" value="{{old('name',$company-
>name)}}}"
            class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
            dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
            placeholder="Nombre o Razon Social" minlength="5" maxlength="150"
required>
    </div>

    <div class="col-span-2">
        <label for="address" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">Dirección</label>
        <input type="text" id="address" name="address"
value="{{old('address',$company->address)}}}"
            class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
            dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
            placeholder="Dirección compania" minlength="5" maxlength="150">
    </div>

    <div class="col-span-2">
        <label for="phone" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">Celular</label>
        <input type="tel" id="phone" name="phone" value="{{old('phone',$company-
>phone)}}}"
            class="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg
focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5
            dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-
white dark:focus:ring-blue-500 dark:focus:border-blue-500"
            placeholder="Número celular" minlength="10" maxlength="13">
    </div>

    <div class="col-span-2 sm:col-span-2">
        <label for="small" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">Tipo</label>
        <select id="small" name='type' class="block w-full p-2 mb-6 text-sm text-gray-
900 border border-gray-300 rounded-lg bg-gray-50 focus:ring-blue-500 focus:border-blue-500
dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white
dark:focus:ring-blue-500 dark:focus:border-blue-500">
            <option value="major" {{ $company->type == 'major' ? 'selected':'' }}>Empresa
Principal</option>
            <option value="supplier" {{ $company->type == 'supplier' ?
'selected':'' }}>Proveedor</option>

```

```

        </select>
      </div>
    </div>
    <div class="flex justify-end gap-2">
      <input type="hidden" name="page" value="{{ request('page', 1) }}">
      <a onclick="cerrarModal('companyModalEdit-{{$companyId}}')" class="bg-
orange-500 text-white px-4 py-2 rounded cursor-pointer">Cancelar</a>
    </button>
    <button type="submit" class="bg-green-500 text-white px-4 py-2
rounded">Guardar</button>
  </div>
</form>
</div>
</div>

```

Interfaz eliminar proveedor

Interfaz encargada de eliminar un proveedor, únicamente visible para usuarios administradores del sistema (modal-delete-company.blade.php).

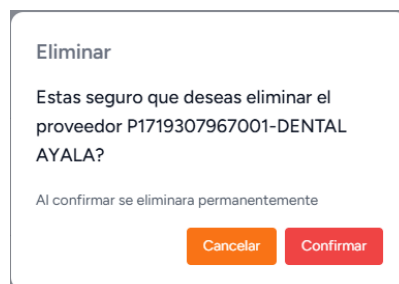


Ilustración 182. Interfaz eliminar proveedor.

Código fuente

```

@props(['companyId','company'])

<a onclick="mostrarModal('companyModalDelete-{{$companyId}}')"
  class=" cursor-pointer text-red-700 hover:bg-red-700 hover:text-white focus:ring-4
focus:outline-none focus:ring-red-300 font-sm rounded-lg text-sm p-2.5 text-center inline-flex
items-center me-2 dark:border-red-500 dark:text-red-500 dark:hover:text-white
dark:focus:ring-red-800 dark:hover:bg-red-500">Eliminar</a>
<div id="companyModalDelete-{{$companyId}}" class="fixed inset-0 bg-gray-900 bg-
opacity-50 items-center justify-center hidden z-50">
  <div class="bg-white p-6 rounded-xl shadow-lg w-96">
    <h2 class="text-xl font-bold mb-4">Eliminar</h2>

```

```

<form id="formcompany-{{$companyId}}"
action="{{route('company.destroy',$company)}}" method="POST" >
    @csrf
    @method('DELETE')
    <div class="grid gap-4 mb-4">
        <h2 class="text-lg font-medium text-gray-900 dark:text-gray-100">
            {{ __('Estas seguro que deseas eliminar el proveedor ') }} {{$company->code.'-'.
            '$company->name.'?}} <br>
        </h2>
        <p class="mt-1 text-sm text-gray-600 dark:text-gray-400">
            {{ __('Al confirmar se eliminara permanentemente') }}
        </p>
    </div>
    <div class="flex justify-end gap-2">
        <input type="hidden" name="page" value="{{ request('page', 1) }}">
        <a onclick="cerrarModal('companyModalDelete-{{$companyId}')" class="bg-
orange-500 text-white px-4 py-2 rounded cursor-pointer">
            Cancelar
        </a>
        <button type="submit" class="bg-red-500 text-white px-4 py-2
rounded">Confirmar</button>
    </div>
</form>
</div>
</div>

```

Interfaz buscar proveedor

Interfaz encargada de buscar un proveedor mediante un filtro ingresado por el usuario (modal-share-company.blade.php).



Ilustración 183. Interfaz búsqueda proveedor.

Código fuente

```

@props(['companies'])
<div class="w-full bg-white flex items-center justify-center p-2">
    @can('create', App\Models\Company::class)
        <x-companycomponents.modal-new-company />
    @endcan

```

```

<x-companycomponents.listpdf-company />
<form method="GET" action="{{ route('company.index') }}" class="flex w-full max-w-xl
gap-2 ">
  <input
    type="text"
    name="buscar"
    placeholder="Buscar..."
    value="{{ request('buscar') }}"
    class="flex-1 rounded border border-gray-300 text-slate-500"
  >
  <button
    type="submit"
    class="text-blue-700 border hover:bg-blue-700 hover:text-white focus:ring-4
focus:outline-none focus:ring-blue-300 font-sm rounded-lg text-sm p-2.5 text-center inline-flex
items-center me-2 dark:border-blue-500 dark:text-blue-500
    dark:hover:text-white dark:focus:ring-blue-800 dark:hover:bg-blue-500"
  >
    <svg xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 24 24" stroke-
width="1.5" stroke="currentColor" class="size-6">
      <path stroke-linecap="round" stroke-linejoin="round" d="m21 21-5.197-5.197m0
0A7.5 7.5 0 1 0 5.196 5.196a7.5 7.5 0 0 0 10.607 10.607Z" />
    </svg>
  </button>
</form>
</div>

```

Interfaz reporte de proveedores

Interfaz encargada de crear un reporte de todos los proveedores en formato PDF (listpdf.blade.php) esta interfaz se encuentra en el directorio View/Dashboard/Companies.

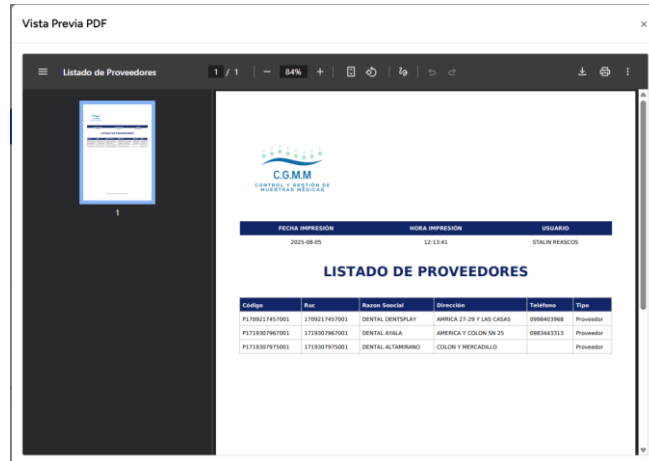


Ilustración 184. Reporte proveedores.

Código fuente

```
<!DOCTYPE html>
<html>
<head>
  <title>Listado de Proveedores</title>
  <x-style-report/>
</head>
<body>
  <x-header-report/>
  <h2>
    LISTADO DE PROVEEDORES
  </h2>
  @foreach ($companies->chunk(10) as $chunk)
  <table>
    <thead>
      <tr>
        <th>Código</th>
        <th>Ruc</th>
        <th>Razon Soocial</th>
        <th>Dirección</th>
        <th>Teléfono</th>
        <th>Tipo</th>
      </tr>
    </thead>
    <tbody>
      @foreach ($chunk as $company)
      <tr>
        <td>
          {{ $company->code }}
        </td>
      </tr>
    </tbody>
  </table>
  @endforeach
</body>
</html>
```

```

        <td>
            {{ $company->ruc }}
        </td>
        <td>{{ $company->name }}</td>
        <td>{{ $company->address }}</td>
        <td>{{ $company->phone }}</td>
        <td>
            @if ($company->type == 'supplier')
                Proveedor
            @else
                Empresa Principal
            @endif
        </td>
    </tr>
</foreach>
</tbody>
</table>
@if (!$loop->last)
    <div class="page-break"></div>
@endif
</foreach>
<x-footer-report/>
</body>
</html>

```

Componente JavaScript

Este componente se encarga de validar y realizar operaciones sin necesidad de recargar la página dentro del navegador con el fin de que el usuario tenga una experiencia visual y de operación de calidad.

Código fuente

```

@if ($errors->any() && session('editing_company_id'))
    <script>
        window.addEventListener('DOMContentLoaded', () => {
            const modalId = 'companyModalEdit-' + {{ session('editing_company_id') }};
            const modal = document.getElementById(modalId);
            if (modal) {
                modal.classList.remove('hidden');
                modal.classList.add('flex');
            }
        });
    </script>
@endif

```

```

@if ($errors->any() && session('create_company_id'))
<script>
    window.addEventListener('DOMContentLoaded', () => {
        const modalId = 'companyModal';
        const modal = document.getElementById(modalId);
        if (modal) {
            modal.classList.remove('hidden');
            modal.classList.add('flex');
        }
    });
</script>
@endif

<script>
    setTimeout(() => {
        const alert = document.getElementById('alert');
        if (alert) {
            alert.style.transition = "opacity 0.5s ease-out";
            alert.style.opacity = 0;

            // Opcional: quitar el elemento del DOM después de la animación
            setTimeout(() => alert.remove(), 500);
        }
    }, 3000);

    function cerrarModalPDF() {
        const modal = document.getElementById('modalPDF'); // <-- esta línea faltaba
        const iframe = document.getElementById('iframePDF');
        iframe.src = "";
        modal.classList.add('hidden');
        modal.classList.remove('flex');
    }

    function mostrarModal(id) {
        const modal = document.getElementById(id);
        modal.classList.remove('hidden');
        modal.classList.add('flex');
    }

    function cerrarModal(id) {
        const modal = document.getElementById(id);
        modal.classList.remove('flex');
        modal.classList.add('hidden');
        location.reload();
    }

```

```
function abrirModalPDF() {  
  const iframe = document.getElementById('iframePDF');  
  const modal = document.getElementById('modalPDF');  
  iframe.src = '{{ route('listado.empresas') }}';  
  modal.classList.remove('hidden');  
  modal.classList.add('flex');  
}  
</script>
```

Nota interesante

Como se pudo visualizar en el apartado anterior, este manual tiene el código fuente del módulo Proveedores, no se incluye el resto de módulos debido a que tienen la misma estructura por lo cual colocarlos en este manual sería un proceso repetitivo



Ricardo Andrés Santamaría Rosario

C.I.: 1723820872



Stalin Vinicio Reascos Mantilla

C.I.: 1719307967