



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE SISTEMAS Y COMPUTACIÓN

TRABAJO DE TITULACIÓN
APLICACIÓN WEB PROGRESIVA PARA EL MANEJO DE INVENTARIO EN LA
FARMACIA DE LA COORDINACIÓN DE SALUD ZONA 3

Proyecto de Investigación previo a la obtención del título de Ingeniero en Sistemas y
Computación

Autor(es):

Flavio Wilson Tapuy Shiguango
Sairy Amauta Segovia Yamberla

Tutor:

Mgs. Milton Paúl López Ramos

Riobamba – Ecuador

2021

DERECHOS DE AUTORÍA

Nosotros, Flavio Wilson Tapuy Shiguango, con cédula de ciudadanía 1501051682 y Sairy Amauta Segovia Yamberla, con cédula de ciudadanía 1004849392, autor (a) (s) del trabajo de investigación titulado: Aplicación web progresiva para el manejo de inventario en la farmacia de la Coordinación de Salud del Ministerio de Salud Pública zona 3, certificamos que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de nuestra exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 13 de mayo del 2022.



Flavio Wilson Tapuy Shiguango

C.I:

1501051682



Sairy Amauta Segovia Yamberla

C.I:

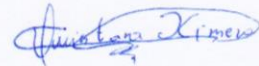
1004849392

DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Aplicación web progresiva para el manejo de inventario en la farmacia de la coordinación de salud zona 3, presentado por Flavio Wilson Tapuy Shiguango, con cédula de identidad número 1501051682 y Sairy Amauta Segovia Yamberla, con cédula de identidad número 1004849392, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

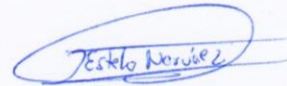
De conformidad a la normativa aplicable firmamos, en Riobamba 13 de mayo del 2022.

PhD. Ximena Alexandra Quintana López
PRESIDENTE DEL TRIBUNAL DE
GRADO



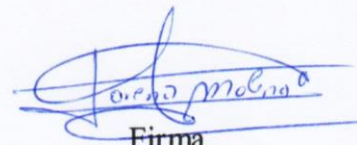
Firma

PhD. Miryan Estela Narváez Vilema
MIEMBRO DEL TRIBUNAL DE
GRADO



Firma

PhD. Lorena Paulina Molina Valdiviezo
MIEMBRO DEL TRIBUNAL DE
GRADO



Firma

Mgs. Milton Paul López Ramos
TUTOR



Firma



Flavio Wilson Tapuy Shiguango

C.I:

1501051682



Sairy Amauta Segovia Yamberla

C.I:

1004849392

CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación Aplicación web progresiva para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3, presentado por Flavio Wilson Tapuy Shiguango, con cédula de identidad número 1501051681 y Sairy Amauta Segovia Yamberla, con cédula de identidad número 1004849392, bajo la tutoría de MsC. Milton López; certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 13 de mayo del 2022.

Presidente del Tribunal de Grado
PhD. Ximena Alexandra Quintana López

Firma

Miembro del Tribunal de Grado
PhD. Miryan Estela Narváez Vilema

Firma

Miembro del Tribunal de Grado
PhD. Lorena Paulina Molina Valdiviezo

Firma



Dirección
Académica
VICERRECTORADO ACADÉMICO

en movimiento

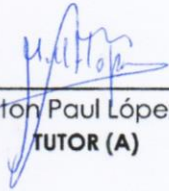


UNACH-RGF-01-04-02.20
VERSIÓN 02: 06-09-2021

CERTIFICACIÓN

Que, FLAVIO WILSON TAPUY SHIGUANGO con CC: 1501051682, y SAIRY AMAUTA SEGOVIA YAMBERLA con CC: 1004849392, estudiantes de la Carrera SISTEMAS Y COMPUTACIÓN, **NO VIGENTE**, Facultad de INGENIERÍA; han trabajado bajo mi tutoría el trabajo de investigación titulado "APLICACIÓN WEB PROGRESIVA PARA EL MANEJO DE INVENTARIO EN LA FARMACIA DE LA COORDINACIÓN DE SALUD ZONA 3", cumple con el 4%, de acuerdo al reporte del sistema Anti plagio URKUND, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 04 de mayo de 2022


Mgs. Milton Paul López Ramos
TUTOR (A)

ÍNDICE GENERAL

| | |
|---|----|
| RESUMEN | 13 |
| ABSTRACT | 14 |
| INTRODUCCION | 15 |
| CAPITULO I | 16 |
| 1. PLANTEAMIENTO DEL PROBLEMA | 16 |
| 1.1. Problema y justificación de la investigación..... | 16 |
| 1.2. Objetivos..... | 17 |
| 1.2.1. Objetivo General..... | 17 |
| 1.2.2. Objetivos específicos..... | 17 |
| CAPÍTULO II | 18 |
| 2. MARCO TEÓRICO | 18 |
| 2.1. Aplicación web progresiva (PWA)..... | 18 |
| 2.1.1. Características | 18 |
| 2.1.2. Comparativa de una PWA frente a otras tipologías | 19 |
| 2.2. Patrón MVC (Modelo-Vista-Controlador)..... | 19 |
| 2.2.1. Capas del patrón MVC..... | 19 |
| 2.3. Tecnologías de una PWA..... | 20 |
| 2.3.1. Service Workers | 20 |
| 2.3.2. Web app Manifest | 20 |
| 2.3.3. App Shell..... | 20 |
| 2.4.1. Vue Js | 20 |
| 2.4.2. Node Js | 21 |
| 2.4.3. MySQL..... | 21 |
| 2.4.4. Apache JMeter..... | 21 |
| 2.4.5. Api Rest..... | 21 |
| 2.4.6. Vuetify..... | 21 |
| 2.4.7. JavaScript | 21 |
| 2.4.8. ASP.NET Core | 21 |
| 2.5.1. Rapid Application Development (RAD)..... | 22 |
| 2.5.3. Fases de la metodología | 23 |
| 2.5. Pruebas de rendimiento..... | 23 |

| | |
|---|-----------|
| 2.5.1. Pruebas de carga..... | 23 |
| 2.5.2. Pruebas de estabilidad..... | 24 |
| 2.6. Métricas de la ISO/IEC 25023..... | 24 |
| 2.6.1. Tiempos de respuesta..... | 24 |
| CAPÍTULO III | 25 |
| 3. METODOLOGÍA | 25 |
| 3.1.1. Variable independiente..... | 25 |
| 3.1.2. Variable dependiente..... | 25 |
| 3.4. Operacionalización de variables..... | 26 |
| 3.5. Desarrollo del software de inventario..... | 27 |
| 3.5.1. Modelado de gestión..... | 27 |
| 3.5.1.1 Proceso de inventario en la Coordinación..... | 27 |
| 3.5.2. Modelado de datos..... | 46 |
| 3.5.3. Modelado de procesos..... | 47 |
| 3.5.4. Generación de aplicaciones..... | 49 |
| 3.5.5. Pruebas y entrega..... | 56 |
| CAPÍTULO IV | 61 |
| 4. RESULTADOS Y DISCUSION | 61 |
| 4.1. Resultados..... | 61 |
| 4.1.1. Una aplicación PWA para el manejo de inventario..... | 61 |
| 4.2. Discusión..... | 67 |
| CAPÍTULO V | 69 |
| 5. CONCLUSIONES Y RECOMENDACIONES | 69 |
| 5.1. Conclusiones..... | 69 |
| 5.2. Recomendaciones..... | 69 |
| Anexo 1: Manual de usuario..... | 72 |
| Anexo 2: Manual técnico..... | 104 |
| Anexo 3: Modelo Físico..... | 108 |

ÍNDICE FIGURAS

| | |
|---|----|
| Figura 1. Metodología R.A.D..... | 22 |
| Figura 2. Proceso de desarrollo de la metodología RAD..... | 27 |
| Figura 3. Proceso de distribución de medicamentos en la Coordinación Zona 3..... | 28 |

| | |
|--|----|
| Figura 4. Semaforización de caducidad y stock | 28 |
| Figura 5. Base de datos en MySQL..... | 47 |
| Figura 6. Arquitectura de una Api Rest..... | 48 |
| Figura 7. Servicios de la tabla beneficiario-pacientes | 48 |
| Figura 8. Servicios de la tabla beneficiario-districtos | 48 |
| Figura 9. Servicios de la tabla usuario..... | 49 |
| Figura 10. Actualizar stock comprobantes o boletas..... | 49 |
| Figura 11. Actualizar stock pedido..... | 50 |
| Figura 12. Actualizar stock pedido..... | 50 |
| Figura 13. Distribución de carpetas del Back End | 51 |
| Figura 14. Conformación del paquete Inventario _Datos | 51 |
| Figura 15. Conformación de paquete Inventario _Entidades | 52 |
| Figura 16. Conformación de paquete Inventario _Web | 52 |
| Figura 17. Conexión a la base de datos | 53 |
| Figura 18. Archivos vue de la carpeta components..... | 53 |
| Figura 19. Código de la vista login.vue..... | 54 |
| Figura 20. Configuración del archivo Service Worker | 55 |
| Figura 21. Configuración del archivo Manifest | 56 |
| Figura 22. Grupo de usuarios recurrentes para pruebas de carga..... | 57 |
| Figura 23. Pruebas de estabilidad..... | 57 |
| Figura 24. Vista administrador | 58 |
| Figura 25. Vista bodega general..... | 58 |
| Figura 26. Vista distrito..... | 59 |
| Figura 27. Vista unidad operativa – farmacia | 59 |
| Figura 28. Vista médico | 60 |
| Figura 29. Aplicación PWA ejecutándose en el ordenador..... | 61 |
| Figura 30. Aplicación PWA ejecutándose en un dispositivo móvil..... | 62 |
| Figura 31. Resultado pruebas de carga..... | 63 |
| Figura 32. Resultados de las pruebas de carga al path Login..... | 63 |
| Figura 33. Resultados de las pruebas de carga al path Productos | 64 |
| Figura 34. Resultados de las pruebas al path Boletas..... | 64 |
| Figura 35. Resultados de las pruebas de carga al path Cuentas | 65 |
| Figura 36. Resultados de las pruebas de carga al path Lotes | 65 |
| Figura 37. Resultados de las pruebas de carga al path Transferencias..... | 66 |
| Figura 38. Prueba de estabilidad | 67 |
| Figura 39. Resultados - prueba de estabilidad..... | 67 |
| Figura 40. Formulario acceso al sistema | 73 |
| Figura 41. Formulario iniciar sesión | 74 |
| Figura 42. Menú principal | 74 |
| Figura 43. Página de Usuarios..... | 75 |
| Figura 44. Módulo de registro de usuarios | 75 |
| Figura 45. Registro de datos del usuario | 76 |
| Figura 46. Página de registro de monedas..... | 76 |
| Figura 47. Registro de monedas | 77 |

| | |
|--|----|
| Figura 48. Notificación de registro exitoso | 77 |
| Figura 49. Página de Productos | 78 |
| Figura 50. Botón “NUEVO” | 78 |
| Figura 51. Formulario registro de Productos..... | 78 |
| Figura 52. Registro de Nombre y Categoría..... | 79 |
| Figura 53. Registro de Nueva categoría | 79 |
| Figura 54. Mensaje de registro correcto | 80 |
| Figura 55. Registro de Nueva Forma Farmacéutica | 80 |
| Figura 56. Registro de nueva unidad..... | 80 |
| Figura 57. Registro de Nuevo proyecto..... | 81 |
| Figura 58. Registro de Nuevo grupo | 81 |
| Figura 59. Registro de Nueva Cuenta Presupuestaria | 81 |
| Figura 60. Registro de Nueva Cuenta Contable | 82 |
| Figura 61. Registro de Nuevo Programa | 82 |
| Figura 62. Registro completo de datos del Producto..... | 83 |
| Figura 63. Mensaje de registro correcto | 83 |
| Figura 64. Página de Beneficiarios..... | 84 |
| Figura 65. Botón “NUEVO” | 84 |
| Figura 66. Registro completo de Nuevo Paciente | 84 |
| Figura 67. Mensaje de registro correcto | 85 |
| Figura 68. Registro completo de nuevo Distrito | 85 |
| Figura 69. Página de Unidades Operativas..... | 85 |
| Figura 70. Formulario de registro de Nuevas Unidades Operativas | 86 |
| Figura 71. Página de registro de lotes | 86 |
| Figura 72. Formulario de registro de Lotes | 87 |
| Figura 73. Formulario de asignación de lote | 87 |
| Figura 74. Página de registro de tipos de cuentas | 88 |
| Figura 75. Formulario de registro de tipo de cuenta | 88 |
| Figura 76. Página de Comprobante | 89 |
| Figura 77. Formulario de Registro de Comprobantes | 89 |
| Figura 78. Realización de ingresos..... | 90 |
| Figura 79. Selección de beneficiario | 90 |
| Figura 80. Selección de moneda..... | 91 |
| Figura 81. Ventana de selección de los productos | 91 |
| Figura 82. Búsqueda de Productos | 92 |
| Figura 83. Módulo de registro de Comprobante | 92 |
| Figura 84. Realización de transferencias..... | 93 |
| Figura 85. Selección de unidad operativa en beneficiario..... | 93 |
| Figura 86. Módulo de registro de Transferencia Temporal..... | 94 |
| Figura 87. Módulo de Comprobantes | 94 |
| Figura 88. Página de Transferencias | 95 |
| Figura 89. Opciones de Transferencias | 95 |
| Figura 90. Vista previa al realizar la transacción | 96 |
| Figura 91. Estado pendiente de la transferencia..... | 96 |

| | |
|---|-----|
| Figura 92. Ventana de aceptación de la transferencia | 97 |
| Figura 93. Página de registro de Pacientes | 97 |
| Figura 94. Formulario de registro de Paciente | 98 |
| Figura 95. Formulario de egreso | 98 |
| Figura 96. Selección de productos para egresos | 99 |
| Figura 97. Módulo de rol Médico..... | 99 |
| Figura 98. Página de lotes - rol médico..... | 100 |
| Figura 99. Tabla “Unidad” | 100 |
| Figura 100. Excel de los datos a importar | 101 |
| Figura 101. Selección de formato de archivo Excel..... | 101 |
| Figura 102. Mensaje para confirmar el respaldo del archivo | 101 |
| Figura 103. Excel de datos previos a ser importados | 102 |
| Figura 104. Página para importar datos a la tabla “Unidad” | 102 |
| Figura 105. Selección del archivo a importar..... | 103 |
| Figura 106. Blog de notas de los datos del archivo Excel..... | 103 |
| Figura 107. Opciones específicas al formato | 103 |
| Figura 108. Tabla “Unidad” con datos importados | 104 |
| Figura 109. Instalación desde un navegador. | 107 |
| Figura 110. Instalación desde un dispositivo móvil | 107 |
| Figura 111. Modelo físico | 108 |

ÍNDICE TABLAS

| | |
|---|----|
| Tabla 1. Características de una aplicación web progresiva | 18 |
| Tabla 2. Comparativa de una PWA frente a otras tipologías | 19 |
| Tabla 3. Operacionalización de variables | 26 |
| Tabla 4. Requerimientos funcionales del sistema..... | 28 |
| Tabla 5. Requerimientos no funcionales | 31 |
| Tabla 6. Asignación de roles | 31 |
| Tabla 7. Ingreso de categorías - gestión de bodega | 32 |
| Tabla 8. Ingreso de lotes - gestión de bodega..... | 32 |
| Tabla 9. Registro de productos - gestión de bodega..... | 33 |
| Tabla 10. Registro de proveedores - gestión de bodega | 34 |
| Tabla 11. Registro de conceptos- gestión de bodega..... | 34 |
| Tabla 12. Registro de programas - gestión de bodega..... | 35 |
| Tabla 13. Registro de presentaciones - gestión de bodega | 35 |
| Tabla 14. Registro de pedidos - gestión de bodega | 36 |
| Tabla 15. Registro de ingresos de productos - gestión de bodega..... | 37 |
| Tabla 16. Registro de egresos de productos - gestión de bodega | 37 |
| Tabla 17. Registro de reingreso de productos - gestión de bodega | 38 |
| Tabla 18. Ingreso de categorías - gestión de bodega | 38 |
| Tabla 19. Ingreso de categorías - gestión de unidad operativa..... | 39 |
| Tabla 20. Ingreso de lotes - gestión de unidad operativa | 40 |
| Tabla 21. Registro de productos - gestión de unidad operativa..... | 40 |

| | |
|---|----|
| Tabla 22. Registro de pedidos de pedidos - gestión de unidad operativa..... | 41 |
| Tabla 23. Ingreso de productos - gestión de unidad operativa | 42 |
| Tabla 24. Egresos de productos - gestión de bodega..... | 42 |
| Tabla 25. Registro de reingreso de productos - gestión de bodega | 43 |
| Tabla 26. Kardex - gestión de unidad operativa | 44 |
| Tabla 27. Ingreso de beneficiarios(paciente) - gestión de unidad operativa | 44 |
| Tabla 28. Ingreso de usuarios-Gestión de administrador | 45 |
| Tabla 29. Ingreso de beneficiarios(entidad) - gestión de administrador | 45 |
| Tabla 30. Ingreso de moneda - gestión de administrador | 46 |
| Tabla 31. Resultados de la prueba de carga..... | 62 |

RESUMEN

En el presente proyecto de investigación se desarrolló una Aplicación Web Progresiva (PWA), que ofrece una experiencia similar a una aplicación nativa de dispositivos móviles, beneficios como: interfaz de usuario responsiva, trabajo sin conexión y notificaciones. El aplicativo PWA se lo realizó con el objetivo de llevar un control de inventario, el cual genera notificaciones de desabastecimiento y caducidad de medicamentos en la farmacia de la Coordinación de Salud Zona 3.

Se utilizó la metodología Desarrollo Rápido de Aplicaciones (RAD) para el desarrollo del software, basado en la información de los procesos de inventario de la Coordinación de Salud se estableció los requerimientos funcionales y no funcionales, la base de datos fue creada en MySQL, se construyó servicios Api Rest y se desarrolló una PWA utilizando el framework Vue Js.

La aplicación final fue evaluada por el personal de la Coordinación de Salud ajustándose a sus requerimientos, se realizaron pruebas de rendimiento, aplicando cargas simuladas de usuarios recurrentes de nivel medio-alto con el objetivo de medir los tiempos de respuestas obtenidas con la herramienta Apache JMeter. Los resultados obtenidos en relación con la métrica tiempo de respuesta fueron menores a 200 ms valores aceptados dentro de la norma ISO/IEC 25023 y Google. Como conclusión una aplicación PWA con todas sus características permite gestionar adecuadamente procesos como los de inventarios en la Coordinación de Salud Zona 3.


Palabras clave: Inventario, RAD, Vue js, Aplicación Web Progresiva.

ABSTRACT

In this research project, a Progressive Web Application (PWA) was demonstrated, which offers an experience similar to a native mobile application, with benefits such as a responsive user interface, offline work, and notifications. The PWA application was made to carry out an inventory control, which generates reports of shortages and expiration of medicines in the Zone 3 Health Coordination pharmacy. The Rapid Application Development (RAD) methodology for software development was learned based on the information from the inventory processes of the Health Coordination. The functional and non-functional requirements were established. The database was created in MySQL, Built Rest API services, and developed a PWA using the Vue Js framework. The final application was evaluated by the Health Coordination staff adjusting to their requirements, and performance tests were carried out, applying simulated loads of recurring users of medium-high level to measure the response times obtained with the Apache JMeter tool. The results brought about the response time metric were less than 200 ms, values accepted within the ISO/IEC 25023 and Google standards. In conclusion, a PWA application with all its characteristics allows for adequately managing processes such as those of inventories in the Zone 3 Health Coordination.

Keywords: Inventory, DRA, Vue js, Progressive Web Application.

DARIO
JAVIER
CUTIOPAL
A LEON



Firmado
digitalmente por
DARIO JAVIER
CUTIOPALA LEON
Fecha: 2022.05.18
01:22:36 -05'00'

Reviewed by:

Lic. Dario Javier Cutiopala Leon
ENGLISH PROFESSOR
c.c. 0604581066

INTRODUCCION

En la actualidad las aplicaciones web han tomado un papel muy importante en la sociedad, debido a las ilimitadas posibilidades que estas ofrecen al estar disponibles en la Internet, permitiendo a las personas estar mucho más comunicadas e interrelacionadas que en épocas anteriores. Las aplicaciones web son cada día más comunes hasta el punto de que se tornarán en algo esencial para la vida diaria.

Las típicas aplicaciones web estáticas en la actualidad no son muy recomendadas, debido a que presentan poca información y posteriormente no se puede incluir nuevos contenidos o procesos, de tal forma que no cumple con las expectativas de la empresa en cuanto a gestión (Barzanallana, 2019).

La propuesta planteada fue el desarrollo de una aplicación web progresiva (PWA), este tipo de aplicación reúne lo mejor de las aplicaciones web y de las aplicaciones nativas. Es una tecnología actual que hace que su estética y funcionamiento se asemeje enormemente a una aplicación nativa, por ejemplo, mediante la ejecución en segundo plano, se accede a ellas a través de un navegador, que se puede anclar a la pantalla principal del dispositivo (López, 2020).

Se creó servicios API REST para el back end, que contienen métodos de acceso a datos como: Post (creación de nuevos recursos), get (para obtener un recurso), put (modificar), patch (para modificar un recurso) y delete (para borrar un recurso), a una base de datos en MySQL, y se utilizó el framework Vue.js para el desarrollo del front end.

El presente proyecto de investigación está estructurado en cinco capítulos, que se detallan a continuación:

Capítulo I: Planteamiento del problema, justificación y los objetivos del proyecto.

Capítulo II: Marco Teórico, presenta un estudio de la estructura, características y tecnología de una aplicación web progresiva, el modelo de desarrollo MVC (Modelo-Vista-Controlador), la metodología de desarrollo de software a aplicar en el proyecto y las métricas de evaluación del software.

Capítulo III: Metodología, describe el tipo de investigación, las técnicas de investigación y la metodología de desarrollo.

Capítulo IV: Resultados y discusión, describe los resultados que se obtuvieron en el desarrollo del proyecto de investigación.

Capítulo V: Conclusiones y recomendaciones, en esta sección se presentan las conclusiones del proyecto realizado y las recomendaciones sugeridas sobre el proyecto.

CAPITULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Problema y justificación de la investigación

En la actualidad, los procesos para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3 lo realizan en Microsoft Access, que es un motor de base de datos para pequeñas y medianas empresas, presentando limitaciones tales como seguridad, el tiempo de respuesta se ve afectado por las peticiones que el usuario realice, entre otros factores. Además, utilizan matrices en Excel, lo que dificulta la generación de reportes, debido a que los datos se manejan en diferentes archivos, hoy en día se puede ofrecer muchas alternativas para mejorar este control de inventario, por ejemplo, a través de una aplicación web progresiva que genere alertas de stock y caducidad, que funcione con un gestor de base de datos más escalable, ofreciendo un rendimiento superior, mayor seguridad e integridad de los datos almacenados.

En la bodega de la coordinación de salud zona 3, los productos (medicamentos, insumos, vacunas covid-19, dispositivos, etc.), llegan mediante adquisiciones y donaciones de diferentes proveedores, estas pasan por diferentes puntos hasta llegar al usuario final. El primer punto al que llegan todos los productos es bodega central, en donde se dividen en 3 secciones: medicamentos, dispositivos médicos y vacunas, después estas son trasladadas a los distritos pertenecientes a la zona 3, y por último pasan a las unidades de salud, que es el lugar en donde son distribuidos los medicamentos al usuario final. Para cumplir con todo este proceso, el personal de la coordinación de salud zona 3 trabaja con diferentes herramientas para registrar, procesar y validar la información de los productos. Utilizando herramientas como Excel para cada área, manteniendo los datos por separado debido a que no cuentan con un software que integre toda la información en una sola base de datos. En cuanto a medicamentos, se manejan alertas de stock y caducidad no automatizadas en base a rangos establecidos para la semaforización (Rojo, amarillo y verde), como consecuencia estas pueden ser pasadas por alto, al no contar con notificaciones y alertas automatizadas. Debido a todo lo mencionado anteriormente, no se realiza un adecuado manejo del inventario.

Estos inconvenientes motivaron el desarrollo de una aplicación web progresiva PWA (Progressive Web App), que mediante service workers, manifest y app shell permite ejecutarse en segundo plano sin tener que vivir dentro del navegador, se utilizó el framework vue.js para el desarrollo del front-end y para el back-end se creó servicios que se encargan de realizar las peticiones a la base de datos de MySQL.

La presente investigación es factible de realizar por la necesidad de la Coordinación de Salud Zona 3 de contar con un sistema que integre toda la información en una sola base de datos. De la misma forma por la disponibilidad de herramientas tecnológicas de código abierto (open source) para el desarrollo de software. Se cuenta además con el apoyo de los líderes

del proyecto para que el trabajo de investigación se lleve a cabo correctamente. Finalmente, se dispone de documentación científica que respaldan el desarrollo de la investigación.

La aplicación web progresiva para el manejo de inventario, ayudará a mejorar el control de productos farmacéuticos en la farmacia de la Coordinación de Salud Zona 3.

1.2. Objetivos

1.2.1. Objetivo General

Implementar una Aplicación web progresiva para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3.

1.2.2. Objetivos específicos

- Analizar las principales características y beneficios de las aplicaciones web progresivas.
- Desarrollar una aplicación web progresiva para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3.
- Evaluar el rendimiento de la aplicación web progresiva con la herramienta Apache JMeter.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. Aplicación web progresiva (PWA)

Son aplicaciones web móviles que aprovechan las nuevas posibilidades y APIs que brindan las nuevas tecnologías Web, cómo los Service Worker (script que se ejecuta en segundo plano y permite la implementación de funcionalidades que no requieren de una página web ni interacción del usuario) y Web App Manifest (archivo JSON que permite especificar metadatos de la aplicación tales como nombre, color e icono que la distingue). Esto permite que una aplicación web pueda incorporar algunas de las características exclusivas de las aplicaciones nativas, tales como funcionamiento offline, la recepción de notificaciones push y disponer de un ícono de acceso en el lanzador de aplicaciones (Aguirre, Ortu, & Delía, 2019).

2.1.1. Características

Tabla 1. Características de una aplicación web progresiva

| Aplicación web progresiva | |
|----------------------------------|---|
| Universalidad | Funciona sin problemas para cada usuario, independientemente de su navegador web. |
| Responsiva | Debe funcionar con cualquier dispositivo, ordenador portátil, tablet o teléfono inteligente. |
| Diseño | Debe imitar las aplicaciones móviles nativas, con una interactividad simple para funciones avanzadas. |
| Seguridad | PWA deben usar siempre HTTPS para mantener seguros los datos del usuario. |
| Actualizaciones | Se mantienen actualizados y ofrecen las últimas versiones de un servicio o sitio. |
| Instalación | Desde cualquier navegador sin pasos adicionales y sin necesidad de descargas desde tiendas de aplicaciones. |
| Compartir | PWA solo requiere una única URL para compartir, sin ninguna instalación. |

Autor: (Ranchal, 2019)

2.1.2. Comparativa de una PWA frente a otras tipologías

Tabla 2. Comparativa de una PWA frente a otras tipologías

| | Página Web | Web App | Progressive Web App (PWA) | Aplicación Nativa |
|---|--|---|--|---|
| Plataforma | Independiente de la plataforma. | Independiente de la plataforma. | Independiente de la plataforma. | Independiente de la plataforma |
| Almacenamiento de datos | En el servidor. | En el servidor. | Al ser App creadas con HTML5, el código de la aplicación y los datos pueden guardarse temporalmente. | En el dispositivo del usuario. |
| Utilización de las funciones del dispositivo | No es posible. | No es posible en la mayor parte de los casos. Solo algunas veces de forma limitada. | Es posible en casi todas las Apps. | Es posible utilizarla en su totalidad. |
| Instalación | No es necesaria. | No es necesaria. | No es necesaria. | No es necesaria. |
| Actualizaciones | Las implementa el proveedor y están disponibles para todos los usuarios. | Las implementa el proveedor y están disponibles para todos los usuarios. | Las implementa el proveedor y están disponibles para todos los usuarios. | Deben instalarse. |
| Conexión a internet | Siempre. | Casi siempre. | No es necesaria para que funcione la App. | Depende de la aplicación muchas de ellas no lo necesitan. |

Autor: (López, 2020)

2.2. Patrón MVC (Modelo-Vista-Controlador)

Es un modelo que se divide en tres partes que conforman una aplicación, el modelo, las vistas y los controladores, permitiendo la implementación por separado de cada elemento, garantizando así la actualización y mantenimiento del software de forma sencilla y en un reducido espacio de tiempo. Mediante el uso de frameworks basados en el patrón MVC se puede lograr una mejor organización del trabajo (Díaz & Fernández, 2015).

2.2.1. Capas del patrón MVC

El Modelo es el objeto que representa los datos del programa, manejando los datos y controla todas sus transformaciones.

La Vista es el objeto que maneja la presentación visual de los datos representados por el modelo, en pocas palabras la interfaz de usuario para que este pueda interactuar con el sistema. Interactúa preferentemente con el controlador, pero es posible que trate directamente con el modelo a través de una referencia al propio modelo.

El Controlador es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el Modelo, centra toda la interacción entre la vista y el modelo (Díaz & Fernández, 2015).

2.3. Tecnologías de una PWA

2.3.1. Service Workers

Esta tecnología es una especie de proxy entre el servidor o la red y el dispositivo o la aplicación. Es un JavaScript que se instala en el navegador y funciona detectando eventos. Requieren del uso de HTTPS. Funcionan independientemente de la aplicación, en segundo plano. Gracias a ellos, la Progressive Web App puede ser utilizada sin conexión, puesto que se guardan datos en caché y hace un almacenamiento offline (López, 2020).

2.3.2. Web app Manifest

El manifiesto de la aplicación web tiene la función de proporcionar información sobre una aplicación (como su nombre, autor, icono y descripción) en un archivo de texto JSON, debe informar los detalles de los sitios web instalados en la pantalla de inicio de un dispositivo, proporcionando a los usuarios un acceso más rápido y una mejor experiencia (Tandel & Jamadar, 2018).

2.3.3. App Shell

Es la mínima cantidad de HTML, CSS y JavaScript requeridos para activar la interfaz de usuario, cuando se almacena en caché sin conexión puede asegurar un rendimiento instantáneo y de alta confiabilidad para los usuarios en las visitas repetidas. De esta manera, la shell de la app no se carga desde la red en cada visita del usuario, solo se carga el contenido necesario de la red (Osmani, 2019).

2.4. Herramientas y framework de desarrollo

2.4.1. Vue Js

Es un marco progresivo para construir interfaces de usuario, diferente a otros marcos monolíticos, está diseñado desde cero para ser adaptable gradualmente. La biblioteca principal se centra solo en la capa de vista y es muy fácil de recoger e integrar con otras bibliotecas o proyectos existentes. Por otro lado, también es perfectamente capaz de realizar aplicaciones sofisticadas de una sola página cuando se utiliza en combinación con herramientas modernas y bibliotecas de apoyo. Es un desarrollador de front end experimentado (Kyriakidis et al., 2017).

2.4.2. Node Js

Es una plataforma de software de ejecución de JavaScript V8 de Chrome para construir una red escalable aplicaciones sin esfuerzo. Node.js usa un modelo de E / S sin bloqueo controlado por eventos que lo hace ligero y eficiente, perfecto para datos intensivos en tiempo real aplicaciones que se ejecutan en distribuidos dispositivos (Hota & Prabhu, 2016).

2.4.3. MySQL

Es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL (Licencia Pública General) le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo (Santillan, 2014).

2.4.4. Apache JMeter

Es una aplicación de escritorio, que está diseñada para probar y medir el rendimiento y comportamiento funcional de aplicaciones cliente/servidor, como aplicaciones web o FTP, es una herramienta de prueba de código abierto. Su función principal es cargar el cliente / servidor de prueba, permitiendo probar aplicaciones que utilizan servidores HTTP o FTP, que se basa en Java que es altamente extensible a través de una API proporcionada (Halili, 2015).

2.4.5. Api Rest

Es un estilo arquitectural que permite la comunicación entre la capa del modelo y la capa web, presentadas en formato JSON O XML. Su principal característica es que define una interfaz uniforme entre los componentes del sistema, además utiliza los métodos HTTP de manera explícita lo que hace más estándar la construcción de estos; por otro lado, no mantiene estado y expone la URL con forma de directorios (Hernández, 2018).

2.4.6. Vuetify

Es una biblioteca de interfaz de usuario de Vue con componentes de materiales desing, permitiendo acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes listos para ser utilizados (Sharo, 2020).

2.4.7. JavaScript

Lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Perez, 2019).

2.4.8. ASP.NET Core

Es un marco de presentación liviano, de código abierto y modular, para la creación de aplicaciones y servicios sobre Windows, Linux y Mac. ASP.NET Core, proporciona una

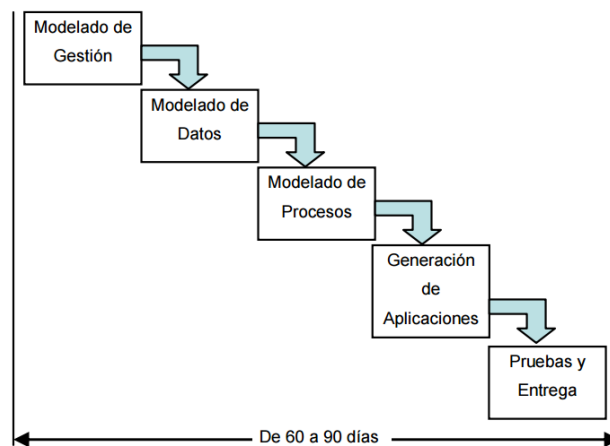
forma basada en patrones para crear sitios web dinámicos que permite una separación limpia de código, que incorpora el tipo de arquitectura MVC (Smith, 2017).

2.5. Metodología de desarrollo de software

2.5.1. Rapid Application Development (RAD)

El desarrollo rápido de aplicaciones es un enfoque de desarrollo de software ágil que se centra más a las entregas e iteraciones rápidas de prototipos. por tanto, prioriza la creación sobre la planificación costosa. Este modelo permite tratar los proyectos de software como arcilla, en lugar de acero, que es como los tratan las prácticas tradicionales de desarrollo como Waterfall. RAD es menos charla y más trabajo, es decir, menos palabras y más acciones. Para ello se realizan muchas pruebas y se siguen una serie de fases o pasos, a pesar de que RAD desestima la planificación estricta (Pelález, 20220).

Figura 1. Metodología R.A.D



Fuente: (SISINGGROUP, 2017)

2.5.2. Características de la Metodología RAD

A continuación, se listan las principales características (Zea, 2016).

- Equipos híbridos: Los desarrolladores deben ser analistas, diseñadores y programadores en uno.
- Herramientas especializadas: Desarrollo visual, Herramientas colaborativas, Interfaces estándares, Control de versiones, Calendario grupal y Componentes reusables.
- Timeboxing: Las funciones secundarias son eliminadas. Técnica que consiste en fijar el tiempo máximo para conseguir unos objetivos, tomar una decisión o realizar unas tareas.
- Prototipos iterativos: Se reúnen los usuarios y los desarrolladores. Lluvia de ideas para obtener un borrador inicial de los requisitos.

- El facilitador: Tiene clara las metas, prepara la agenda de asuntos y escribe un reporte final

2.5.3. Fases de la metodología

La metodología RAD consta de 5 fases que se detallan a continuación (Arbeláez et al., 2015).

- Modelado de gestión: el flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la proceso?.
- Modelado de datos: el flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.
- Modelado de proceso: los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.
- Generación de aplicaciones: La metodología RAD (Desarrollo rápido de aplicaciones) asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.
- Pruebas de entrega: Como el proceso DRA (Desarrollo rápido de aplicaciones) enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

2.5. Pruebas de rendimiento

2.5.1. Pruebas de carga

Es una forma de evaluar el comportamiento de cualquier aplicación cuando se aplica con diferentes cargas (usuarios/tráfico), ayuda a comprender el funcionamiento del programa cuando cierta cantidad de usuarios están usando la plataforma a la vez. Asimismo, las pruebas de carga se aplican para identificar los cuellos de botella o errores cometidos en el proceso de desarrollo, que son responsables del bajo rendimiento o falla de un software en particular. Esto ayuda a los desarrolladores a corregir los errores para asegurarse de que todo funcione sin problemas (Lee, 2020).

2.5.2. Pruebas de estabilidad

Conocido también como Soak Testing, sirve para determinar si la aplicación puede aguantar una carga esperada continuada en un bucle infinito. Generalmente esta prueba se realiza para determinar si hay alguna fuga de memoria en la aplicación (Sánchez, 2018).

2.6. Métricas de la ISO/IEC 25023

La Norma ISO/IEC 25023 define las propiedades medibles de calidad relacionadas de un producto de sistema/software, centrándose en el rendimiento de un dispositivo en función de su comportamiento temporal, el uso de recursos y la capacidad o límites máximos de funcionamiento. Constituida por los tiempos de respuesta y procesamiento de una aplicación en ejecución en condiciones determinadas (Piñero et al., 2021).

2.6.1. Tiempos de respuesta

Para que los usuarios realmente perciban si un sistema es óptimo, es fundamental reducir el tiempo de respuesta que se realizan al servidor al mínimo. Según la norma ISO/IEC 25023, el tiempo de respuesta más cercano a cero es el mejor, de igual forma, en las directrices de Google un buen tiempo de respuesta del servidor debe estar por debajo de los 200 milisegundos (Villa, 2017).

Donde:

X: Tiempo de respuesta.

A: Tiempo de envío de petición.

B: Tiempo en recibir la primera respuesta.

Formula:

$$X = B - A$$

CAPÍTULO III

3. METODOLOGÍA

3.1. Identificación de las Variables

3.1.1. Variable independiente

Aplicación web progresiva.

3.1.2. Variable dependiente

El rendimiento de la aplicación web progresiva en el manejo de inventario de la farmacia de la Coordinación de Salud Zona 3.

3.2. Tipo de investigación

Aplicada: Porque la investigación busca dar solución a un problema determinado, con el objetivo de encontrar soluciones que puedan incorporar, en este caso optimizar el manejo de inventario de la farmacia de la Coordinación de Salud Zona 3.

Explicativa: Se utiliza debido a que se parte desde el origen del problema, donde existe una relación causa-efecto, es decir, analizar las causas y efectos que se presentan al no llevar un correcto manejo de inventario.

3.3. Técnicas de investigación

Para el levantamiento de información del presente proyecto se utilizarán las siguientes técnicas de investigación cuantitativas, que facilitarán el proceso de recopilación de datos indispensables para el desarrollo del sistema.

- **La observación:** Mediante esta técnica se podrá percibir el comportamiento de la bodega, distrito y unidades operativas con respecto al manejo de los procesos de inventario que se están llevando en la Coordinación de Salud Zona 3.
- **La entrevista:** Con esta técnica se podrá entrevistar a los responsables de las siguientes áreas: Tics, bodega, distrito, unidades operativas-farmacias, con el fin de identificar los procesos de inventario que llevan por separado en diferentes softwares informáticos, para posteriormente levantar los requerimientos funcionales que cada área contemple necesario para el sistema.

3.4. Operacionalización de variables

Tabla 3. Operacionalización de variables

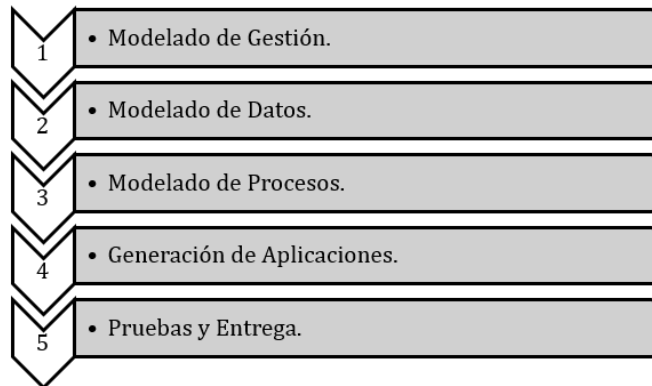
| PROBLEMA | TEMA | OBJETIVOS | VARIABLES | INDICADORES |
|--|---|---|--|--|
| El manejo de inventario en la farmacia de la Coordinación de Salud Zona 3 mediante el desarrollo de una aplicación web progresiva. | Aplicación web progresiva para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3. | General | Independiente | Independiente |
| | | Implementar una Aplicación web progresiva para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3. | Aplicación web progresiva. | Número de archivos configurados (manifest, ServiceWorker, etc.). |
| | | Específicos | Dependiente | Dependiente |
| | | Analizar las principales características, funcionalidades y beneficios de las aplicaciones web progresivas. | El rendimiento de la aplicación web progresiva en el manejo de inventario de la farmacia de la Coordinación de Salud Zona 3 con Apache JMeter. | Número de alertas generadas. |
| | | Desarrollar una aplicación web progresiva para la farmacia del Ministerio de Salud Pública de la Zona 3. | | Pruebas de carga de usuarios concurrentes. |
| | | Evaluar el rendimiento de la aplicación web progresiva mediante la herramienta Apache Jmeter. | | Pruebas de estabilidad de la PWA. |

Fuente: Elaboración propia

3.5. Desarrollo del software de inventario

Se utilizó la metodología de desarrollo de aplicaciones ágil RAD (Rapid Application Development). La metodología consta de 5 fases que se detallaran a continuación:

Figura 2. Proceso de desarrollo de la metodología RAD



Fuente: Elaboración propia

3.5.1. Modelado de gestión

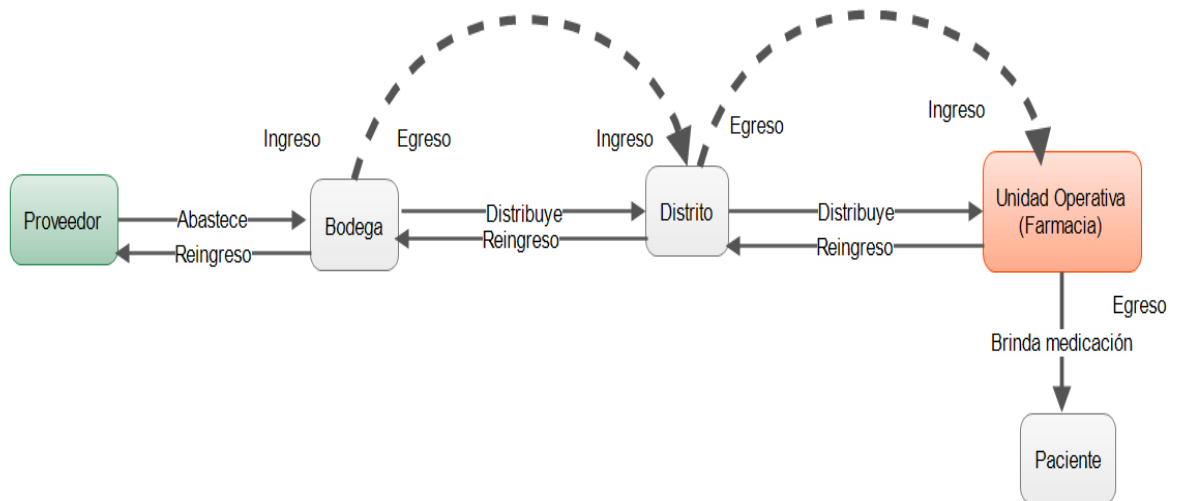
En esta fase se realizó una visita al personal encargado de la Coordinación de Salud para conocer el proceso de inventario de farmacia y realizar el levantamiento de los requerimientos funcionales del sistema.

3.5.1.1. Proceso de inventario en la Coordinación

En el proceso de inventario que se lleva en la Coordinación de Salud intervienen las siguientes entidades: proveedor, bodega general, distritos, unidades operativas y pacientes.

El proceso comienza cuando el proveedor abastece los productos farmacéuticos a la bodega general de la coordinación zonal, que posteriormente se encarga de distribuir dichos productos a los distritos, y estos distribuyen a las farmacias de las unidades operativas, para que finalmente puedan medicar a los pacientes, como se presenta en la Figura 3.

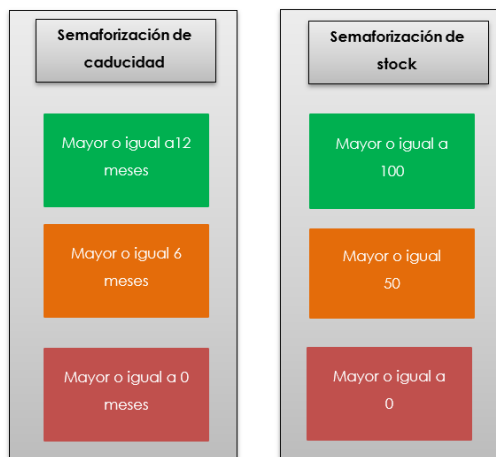
Figura 3. Proceso de distribución de medicamentos en la Coordinación de Salud Zona 3



Fuente: Elaboración propia

En cuanto al control de caducidad y stock de productos lo manejan de acuerdo con una semaforización verde, amarillo, rojo. Siendo este último color el punto más crítico que necesita ser atendida lo más pronto posible, como se muestra en la Figura 4.

Figura 4. Semaforización de caducidad y stock



Fuente: Elaboración propia

3.5.1.2. Requerimientos funcionales y no funcionales del sistema

En este apartado se definió los requerimientos funcionales del proyecto, de acuerdo a las necesidades del usuario y los requerimientos no funcionales que son propios del sistema, como seguridad y rendimiento.

Tabla 4. Requerimientos funcionales del sistema

LISTADO DE REQUERIMIENTOS FUNCIONALES

| RF° | Nombre requerimiento | Especificaciones | Rol |
|------------|-----------------------------|---|------------|
| R1 | Gestión de bodega | <ul style="list-style-type: none"> • El aplicativo debe especificar bajo que concepto se realiza el ingreso de productos. • Los productos deben dividirse en categorías: vacunas, insumos médicos y medicamentos. • Debe permitir manejar los productos por lotes y fechas de caducidad. • Permitir al usuario realizar el ingreso y actualización de datos. • Debe permitir al usuario realizar ingresos, egresos y reingresos de productos. • El usuario deberá poder generar los reportes de los ingresos, reingresos y egresos, donde se detalle la fecha de emisión. • El aplicativo notificará las alertas de stock y caducidad de acuerdo con la semaforización establecida: verde, naranja y roja. | Bodega |
| R2 | Gestión de distrito | <ul style="list-style-type: none"> • Los productos deben dividirse en categorías • Debe permitir manejar los productos por lotes y fechas de caducidad. • Permitir al usuario realizar el ingreso y actualización de datos. • Debe permitir al usuario realizar ingresos, egresos y reingresos de productos. • El usuario deberá poder generar los reportes de los ingresos, reingresos y egresos, donde se detalle la fecha de emisión. • Deberá permitir al usuario consultar el stock de productor por mes. • El aplicativo notificara las alertas de stock y caducidad de acuerdo con la | Distrito |

| | | | |
|-----------|--------------------------------|---|---------------|
| | | semaforización establecida: verde, naranja y roja. | |
| R3 | Gestión de unidades operativas | <ul style="list-style-type: none"> • Los productos deben dividirse en categorías: genérico, maternidad, micronutrientes, insumos. • Debe permitir manejar los productos por lotes y fechas de caducidad. • Debe permitir al usuario realizar ingresos, egresos y reingresos de productos. • El usuario deberá poder generar los reportes de los egresos, reingresos y egresos, donde se detalle la fecha de emisión. • El usuario podrá consultar el stock de su unidad operativa. • El aplicativo notificara las alertas de stock y caducidad de acuerdo a la semaforización establecida: verde, naranja y roja. • Deberá poder realizar el ingreso de pacientes y dichos registros queden almacenados. | Farmacia |
| R4 | Gestión de administrador | <ul style="list-style-type: none"> • Permitirá al usuario crear y tener acceso a las bodegas, distritos unidades operativas, usuarios con su respectivo rol. • El usuario contará con privilegios de administrador donde podrá crear, editar, activar/inactivar o eliminar componentes. • El usuario podrá crear usuarios-médicos acorde a la unidad operativa que correspondan. | Administrador |
| R5 | Gestión de medico | <ul style="list-style-type: none"> • El usuario podrá consultar el stock de su unidad operativa. | Médico |

Fuente: Elaboración propia

Tabla 5. Requerimientos no funcionales

| LISTADO DE REQUERIMIENTOS NO FUNCIONALES | | |
|---|---------------|---|
| Identificador | Nombre | Descripción |
| RNF1 | Rendimiento | La aplicación deberá soportar el ingreso de usuarios concurrentes. |
| RNF2 | Usabilidad | La interfaz de usuario será responsiva y amigable para una mejor interacción usuario – sistema. |
| RNF3 | Seguridad | El usuario puede acceder al sistema por un Login, ya que las páginas estarán protegidas de modo que no puedan acceder con la url a un componente. |

Fuente: Elaboración propia

3.5.1.3. Historias de usuario

Tabla 6. Asignación de roles

| | | |
|---|---|---|
| Identificador: HU-1 | | Nombre: Asignación de roles |
| Tipo: Necesario | Prioridad de desarrollo: Alta | Rol: Administrador |
| Entrada: Datos de los roles como: Código, nombre_rol, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Asignación exitosa de los roles |
| Descripción: El sistema consta de los siguientes roles que son asignados a los usuarios. <ul style="list-style-type: none"> • Administrador • Bodega • Distrito • Farmacia • Medico El usuario administrador puede listar y activar o inactivar los roles. El sistema permite exportar los datos a formato CSV y PDF. El rol administrador tiene acceso libre a todos los módulos del sistema. | | |

| |
|--|
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. |
|--|

Fuente: Elaboración propia

Tabla 7. Ingreso de categorías - gestión de bodega

| | | | |
|--|---|---|--|
| Identificador: HU-2 | | Nombre: Ingreso de categorías - Gestión de bodega | |
| Tipo: Necesario | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de las categorías tales como: Código, nombre_categoria, descripción, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de las categorías. | |
| Descripción: Se registran en el sistema las siguientes categorías de los productos. <ul style="list-style-type: none"> • Vacunas • Insumos médicos • Medicamentos Los usuarios bodega y distrito pueden crear, actualizar y listar las categorías. El usuario administrador puede crear, actualizar, listar, activar o inactivar las categorías. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 8. Ingreso de lotes - gestión de bodega

| | | | |
|---|---|--|--|
| Identificador: HU-3 | | Nombre: Ingreso de lotes - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de los lotes tales como: | | Salida: Registro exitoso de los lotes. | |

| |
|---|
| Código, producto, bodega, nombre_lote, fecha_elaboración, fecha_vencimiento, saldo, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. |
| <p>Descripción: El sistema permite ingresar los lotes por fechas de elaboración y caducidad. Los roles bodega y distrito pueden crear, actualizar, listar los lotes. El usuario administrador puede crear, actualizar, listar, activar o inactivar los lotes. El sistema notificara las alertas de caducidad de acuerdo a la semaforización establecida: verde, amarillo y rojo. El sistema permite exportar los datos a formato CSV y PDF.</p> |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |

Fuente: Elaboración propia

Tabla 9. Registro de productos - gestión de bodega

| | | | |
|--|---|---|--|
| Identificador: HU-4 | | Nombre: Registro de productos - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de productos tales como: Código, categoría, proveedor, presentación, nombre_producto, cantidad, precio_unitario, costo_referencial, stock_minimo, stock_maximo, punto_abastecimiento, observación, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de los productos. | |
| <p>Descripción: El sistema permite ingresar los productos. Los roles bodega y distritos pueden crear, actualizar, listar los productos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los productos. El sistema genera alertas de stock de acuerdo con la semaforización establecida: verde, amarillo y rojo. El sistema permite exportar los datos a formato CSV y PDF.</p> | | | |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> | | | |

Fuente: Elaboración propia

Tabla 10. Registro de proveedores - gestión de bodega

| | | | |
|--|---|---|--|
| Identificador: HU-5 | | Nombre: Registro de proveedores - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos del proveedor tales como: Código, nombre proveedor, dirección, ruc, teléfono, descripción, correo, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de los productos. | |
| Descripción: El sistema permite ingresar los proveedores. Los roles bodega y distrito pueden crear, actualizar, listar los proveedores. El rol administrador puede crear, actualizar, listar, activar o inactivar los proveedores. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 11. Registro de conceptos- gestión de bodega

| | | | |
|---|---|---|--|
| Identificador: HU-6 | | Nombre: Registro de conceptos - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de conceptos tales como: Código, nombre concepto, estado concepto, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de conceptos. | |
| Descripción: El sistema debe constar con los siguientes conceptos: | | | |

| |
|---|
| <ul style="list-style-type: none"> • Por compra. • Por donación. • Egreso. • Ingreso. • Reingreso. <p>Los roles bodega y distrito pueden crear, actualizar, listar los conceptos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los conceptos. El sistema permite exportar los datos a formato CSV y PDF.</p> <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |
|---|

Fuente: Elaboración propia

Tabla 12. Registro de programas - gestión de bodega

| | | | |
|---|---|---|--|
| Identificador: HU-7 | | Nombre: Registro de programas - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de programas tales como: Código, nombre_programa, fecha_inicio, fecha_fin, estado_programa, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de programas. | |
| Descripción: El usuario bodega puede crear, actualizar, listar los programas. El usuario administrador puede crear, actualizar, listar, activar o inactivar los programas. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 13. Registro de presentaciones - gestión de bodega

| | | | |
|-------------------------------|---|--|--|
| Identificador: HU-8 | | Nombre: Registro de presentaciones - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |

| | | |
|---|--|--|
| | | |
| Entrada: Datos de presentaciones tales como: Código, nombre_presentación, estado_presentación, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de programas. |
| Descripción: Los usuarios bodega y distrito pueden crear, actualizar, listar las presentaciones. El usuario administrador puede crear, actualizar, listar, activar o inactivar las presentaciones. El sistema permite exportar los datos a formato CSV y PDF. | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | |

Fuente: Elaboración propia

Tabla 14. Registro de pedidos - gestión de bodega

| | | | |
|---|---|---|--|
| Identificador: HU-9 | | Nombre: Registro de pedidos - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega y administrador | |
| Entrada: Datos de presentaciones tales como: Código, concepto, usuario, beneficiario, fecha, orden, número_comprobante, valor_neto, IVA, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de pedidos. | |
| Descripción: Los usuarios bodega y distrito pueden crear, actualizar, listar los pedidos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los pedidos. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 15. Registro de ingresos de productos - gestión de bodega

| | | | |
|--|---|---|--|
| Identificador: HU-10 | | Nombre: Registro de ingresos de productos - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de los ingresos tales como: Código, usuario, beneficiario, concepto, moneda, restricción, fecha, orden, número_comprobante, valor_net, iva, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de ingresos. | |
| Descripción: El número de comprobante es el mismo que el número de comprobante de pedidos para realizar el ingreso. La restricción no aplica para realizar ingresos. El sistema genera reportes donde se detalle la fecha y hora de emisión. Los usuarios bodega y distrito pueden crear, listar los ingresos. El usuario administrador puede crear, listar, activar o inactivar los ingresos. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 16. Registro de egresos de productos - gestión de bodega

| | | | |
|--|---|--|--|
| Identificador: HU-11 | | Nombre: Registro de egresos de productos - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de los egresos tales como: Código, usuario, beneficiario, concepto, moneda, restricción, fecha, orden, número_comprobante, valor_net, iva, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de egresos. | |

| |
|---|
| |
| <p>Descripción: La restricción aplica para realizar egresos, donde la bodega podrá realizar los egresos. El sistema genera reportes donde se detalle la fecha y hora de emisión. Los usuarios bodega y distrito pueden crear, actualizar, listar los egresos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los egresos.</p> |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |

Fuente: Elaboración propia

Tabla 17. Registro de reingreso de productos - gestión de bodega

| | | | |
|--|---|---|--|
| Identificador: HU-12 | | Nombre: Registro de reingresos de productos - Gestión de bodega | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Bodega, distrito y administrador | |
| Entrada: Datos de los reingresos tales como: Código, usuario, beneficiario, concepto, moneda, restricción, fecha, orden, número_comprobante, valor_netto, IVA, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de reingresos. | |
| Descripción: La restricción no aplica para realizar reingresos. El sistema genera reportes donde se detalle la fecha y hora de emisión. Los usuarios bodega y distrito pueden crear, actualizar, listar los reingresos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los reingresos. El número de comprobante es el mismo que de egreso para realizar el reingreso. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 18. Ingreso de categorías - gestión de bodega

| | | | |
|--------------------------------|---------------------------------|---|--|
| Identificador: HU-13 | | Nombre: Kardex - Gestión de beneficiario – bodega | |
| Tipo: | Prioridad de desarrollo: | Rol: | |

| | | |
|--|------|--|
| Necesaria | Alta | Bodega, distrito y administrador |
| Entrada: Datos del Kardex tales como: Código, producto, lote, beneficiario, concepto, fecha_transacción, cantidad, precio_unitario, precio_total, saldo, numero_comprobante, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso del Kardex. |
| Descripción: Los usuarios bodega y distrito pueden crear, actualizar, listar los campos del Kardex. El usuario administrador puede crear, actualizar, listar, activar o inactivar los campos del Kardex. El sistema permite exportar los datos a formato CSV y PDF. | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | |

Fuente: Elaboración propia

Tabla 19. Ingreso de categorías - gestión de unidad operativa

| | | | |
|--|---------------------------------|---|--|
| Identificador: | | Nombre: | |
| HU-14 | | Ingreso de categorías - Gestión de unidad operativa | |
| Tipo: | Prioridad de desarrollo: | Rol: | |
| Necesario | Alta | Farmacia y administrador | |
| Entrada: Datos de las categorías tales como: Código, nombre_categoria, descripción, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de las categorías. | |
| Descripción: Se registrarán en el sistema las siguientes categorías de los productos. <ul style="list-style-type: none"> • Genérico. • Maternidad. • Micronutrientes. • Insumos médicos. El usuario farmacia puede crear, actualizar, listar las categorías. | | | |

El usuario administrador puede crear, actualizar, listar, activar o inactivar las categorías.
El sistema permite exportar los datos a formato CSV y PDF.

Pruebas de aceptación:
Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.

Fuente: Elaboración propia

Tabla 20. Ingreso de lotes - gestión de unidad operativa

| | | | |
|---|---|--|--|
| Identificador: HU-15 | | Nombre: Ingreso de lotes - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos de los lotes tales como: Código, producto, bodega, nombre_lote, fecha_elaboración, fecha_vencimiento, saldo, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de las categorías. | |
| Descripción: El sistema permite ingresar los lotes por fechas de elaboración y caducidad. El usuario farmacia puede crear, actualizar, listar los lotes. El usuario administrador puede crear, actualizar, listar, activar o inactivar los lotes. El sistema notificara las alertas de caducidad de acuerdo con la semaforización establecida: verde, amarillo y rojo. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

Tabla 21. Registro de productos - gestión de unidad operativa

| | | | |
|--|---|---|--|
| Identificador: HU-16 | | Nombre: Registro de productos - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos de productos tales como: Código, categoría, proveedor, presentación, nombre_producto, | | Salida: Registro exitoso de los productos. | |

| |
|--|
| <p>cantidad, precio_unitario, costo_referencial, stock_minimo, stock_maximo, punto_abastecimiento, observación, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica .</p> |
| <p>Descripción: El usuario farmacia puede crear, actualizar, listar los productos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los productos. El sistema generará alertas de stock de acuerdo con la semaforización establecida: verde, naranja y roja. El usuario farmacia puede consultar el stock de productos de su unidad operativa. El usuario administrador puede consultar el stock de productos de todas las unidades operativas. El sistema permite exportar los datos a formato CSV y PDF. El administrador puede saber el total de consumos de productos e insumos dado fecha inicio – fecha fin de las unidades operativas.</p> |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |

Fuente: Elaboración propia

Tabla 22. Registro de pedidos de pedidos - gestión de unidad operativa

| | | | |
|---|---|---|--|
| Identificador: HU-17 | | Nombre: Registro de pedidos - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos de presentaciones tales como: Código, concepto, usuario, beneficiario, fecha, orden, número_comprobante, valor_neto, IVA, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de pedidos. | |
| Descripción: El usuario farmacia puede crear, actualizar, listar los pedidos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los pedidos. El sistema permite exportar los datos a formato CSV y PDF. | | | |

| |
|--|
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |
|--|

Fuente: Elaboración propia

Tabla 23. Ingreso de productos - gestión de unidad operativa

| | | | |
|--|---|--|--|
| Identificador: HU-18 | | Nombre: Ingreso de productos - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos de los ingresos tales como: Código, usuario, beneficiario, concepto, moneda, restricción, fecha, orden, número_comprobante, valor_neto, IVA, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de ingresos. | |
| Descripción: El número de comprobante es el mismo que el número de comprobante de pedidos para realizar el ingreso. La restricción aplica solo a bodega. El sistema genera reportes donde se detalle la fecha y hora de emisión. El usuario farmacia puede crear, listar los ingresos. El usuario administrador puede crear, listar, activar o inactivar los ingresos. | | | |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> | | | |

Fuente: Elaboración propia

Tabla 24. Egresos de productos - gestión de bodega

| | | | |
|--|---|--|--|
| Identificador: HU-19 | | Nombre: Egresos de productos - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos de los egresos tales como: Código, usuario, beneficiario, concepto, moneda, restricción, fecha, orden, | | Salida: Registro exitoso de egresos. | |

| |
|---|
| número_comprobante, valor_neto, IVA, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. |
| <p>Descripción: La restricción aplica solo a bodega. El sistema genera reportes donde se detalle la fecha y hora de emisión. El usuario farmacia puede crear, listar los egresos. El usuario administrador puede crear, listar, activar o inactivar los egresos.</p> |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |

Fuente: Elaboración propia

Tabla 25. Registro de reingreso de productos - gestión de bodega

| | | |
|---|---|---|
| Identificador: HU-20 operativa | Nombre: Registro de reingresos de productos - Gestión de unidad | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador |
| Entrada: Datos de los reingresos tales como: Código, usuario, beneficiario, concepto, moneda, restricción, fecha, orden, número_comprobante, valor_neto, IVA, total, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | Salida: Registro exitoso de reingresos. | |
| <p>Descripción: La restricción aplica solo a bodega. El sistema genera reportes donde se detalle la fecha y hora de emisión. El usuario farmacia puede crear, actualizar, listar los reingresos. El usuario administrador puede crear, actualizar, listar, activar o inactivar los reingresos.</p> | | |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> | | |

Fuente: Elaboración Propia

Tabla 26. Kardex - gestión de unidad operativa

| | | | |
|---|---|--|--|
| Identificador: HU-21 | | Nombre: Kardex - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos del Kardex tales como: Código, producto, lote, beneficiario, concepto, fecha_transacción, cantidad, precio_unitario, precio_total, saldo, numero_comprobante, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso del Kardex. | |
| Descripción: El usuario farmacia puede crear, actualizar, listar los campos del Kardex. El usuario administrador puede crear, actualizar, listar, activar o inactivar los campos del Kardex. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración Propia

Tabla 27. Ingreso de beneficiarios (paciente) - gestión de unidad operativa

| | | | |
|---|---|---|--|
| Identificador: HU-22 | | Nombre: Ingreso de beneficiarios (paciente) - Gestión de unidad operativa | |
| Tipo: Necesaria | Prioridad de desarrollo: Alta | Rol: Farmacia y administrador | |
| Entrada: Datos del beneficiario tales como: Código, programa, tipo_beneficiario, nombre_beneficiario, ruc, DNI, dirección, teléfono, correo, responsable, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso del beneficiario-paciente. | |
| Descripción: | | | |

| |
|--|
| <p>El usuario farmacia puede crear, actualizar, listar los beneficiarios-pacientes. El usuario administrador puede crear, actualizar, listar, activar o inactivar los beneficiarios-pacientes. El sistema permite exportar los datos a formato CSV y PDF. Si el usuario es una persona natural se deberá ingresar su DNI.</p> |
| <p>Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente.</p> |

Fuente: Elaboración Propia

Tabla 28. Ingreso de usuarios-Gestión de administrador

| | | | |
|---|---|--|--|
| Identificador: HU-23 | | Nombre: Ingreso de usuarios | |
| Tipo: Necesario | Prioridad de desarrollo: Alta | Rol: Administrador | |
| Entrada: Datos de los roles tales como: Código, rol, nombre, apellido, DNI, teléfono, dirección, correo, contraseña, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | | Salida: Registro exitoso de los usuarios | |
| Descripción: El usuario administrador puede crear, actualizar, listar, activar o inactivar los usuarios. El sistema permite exportar los datos a formato CSV y PDF. | | | |
| Pruebas de aceptación: Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración Propia

Tabla 29. Ingreso de beneficiarios(entidad) - gestión de administrador

| | | | |
|---|---|---|--|
| Identificador: HU-24 | | Nombre: Ingreso de beneficiarios(entidad) | |
| Tipo: Necesario | Prioridad de desarrollo: Alta | Rol: Administrador | |
| Entrada: Datos del beneficiario tales como: | | Salida: Registro exitoso de los beneficiarios | |

| | |
|---|--|
| Código, programa, tipo_beneficiario, nombre_beneficiario, ruc, DNI, dirección, teléfono, correo, responsable, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica. | |
| Descripción: | |
| <p>El usuario administrador puede crear, actualizar, listar, activar o inactivar los beneficiarios (distritos, unidades operativas-farmacias).</p> <p>Si el usuario es una entidad se deberá ingresar su ruc.</p> <p>El sistema permite exportar los datos a formato CSV y PDF.</p> | |
| Pruebas de aceptación: | |
| Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | |

Fuente: Elaboración propia

Tabla 30. Ingreso de moneda - gestión de administrador

| | | | |
|---|---------------------------------|-------------------------------|--|
| Identificador: | | Nombre: | |
| HU-25 | | Ingreso de moneda | |
| Tipo: | Prioridad de desarrollo: | Rol: | |
| Necesario | Alta | Administrador | |
| Entrada: | | Salida: | |
| <p>Datos de la moneda tales como:</p> <p>Código, nombre_moneda, estado, usuario_creación, fecha_creación, usuario_modifica y fecha_modifica.</p> | | Registro exitoso de la moneda | |
| Descripción: | | | |
| <p>El usuario administrador puede crear, actualizar, listar, activar o inactivar la moneda.</p> <p>El sistema permite exportar los datos a formato CSV y PDF.</p> | | | |
| Pruebas de aceptación: | | | |
| Mostrar un mensaje de confirmación cuando el registro se ha realizado exitosamente. | | | |

Fuente: Elaboración propia

3.5.2. Modelado de datos

Se desarrolló la relación entre objetos con sus respectivos atributos acorde a los datos recopilados y los requerimientos del sistema de inventario, se realizó el esquema físico de la base de datos para ser implementada en el servidor de base de datos MySQL.

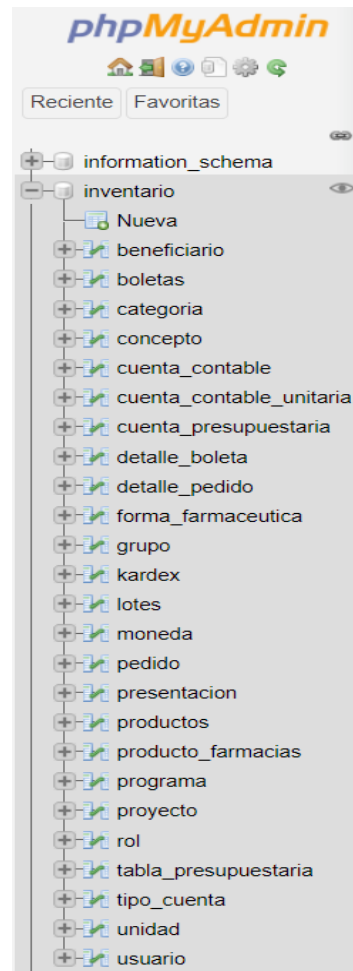
3.5.2.1 Modelo físico

Se diseñó la base de datos de acuerdo con los procesos de inventario en la farmacia de la Coordinación de Salud Zona 3, obteniendo como resultado un total de 25 tablas efectuadas en el modelo físico, como se presenta en el anexo 3, Figura 111.

3.5.2.2. Modelo implementado en el servidor

Se implementó la base de datos inventario en el servidor de base de datos MySQL.

Figura 5. Base de datos en MySQL



Fuente: Elaboración propia

3.5.3. Modelado de procesos

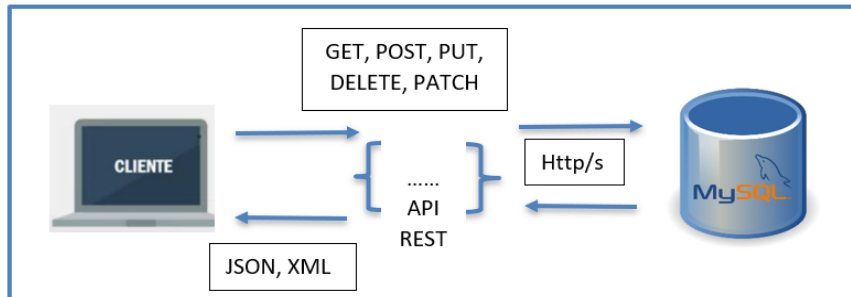
En esta fase se detalla el acceso a los datos y la comunicación entre objetos mediante métodos de petición a implementar en un servicio Api Rest.

3.5.3.1 Api Rest

Para la comunicación entre objetos se ha visto la necesidad de utilizar servicios Api Rest, reglas que permiten que diferentes programas se comuniquen entre sí, donde el cliente realizará peticiones al Api Rest utilizando un método (Get, Post, Put, Delete, Patch) al

servidor mediante el protocolo https y el servidor devolverá recursos en formatos JSON, tal como se presenta en la Figura 6.

Figura 6. Arquitectura de una Api Rest



Fuente: Elaboración propia

3.5.3.2. Servicios Api Rest implementados

En este apartado se presenta algunos de los servicios implementados en ASP.NET Core, los datos visualizados pertenecen a la base de datos MySQL, la petición se realiza mediante una Api Rest, visualizándose en formato Json.

Figura 7. Servicios de la tabla beneficiario-pacientes

```

[{"idBeneficiario":6,"idPrograma":1,"programa":"Prueba","tipoBeneficiario":"Pacien
e","nombreBeneficiario":"Sairy
Segovia","historiaClinica":"2","ruc":null,"dni":"1002576521","pasaporte":"ss10478"
"direccion":"Otavalo","telefono":"0984501795","correo":"sairy@prueba.com","respons
ble":null,"usuarioCrea":"Sairy Segovia","fechaCrea":"2021-11-
13:02:18:13","usuarioModifica":"Sairy Segovia","fechaModifica":"2021-12-
28:18:37:11","estado":1}]

```

Fuente: Elaboración propia

Figura 8. Servicios de la tabla beneficiario-districtos

```

[{"idBeneficiario":1,"idPrograma":1,"programa":"Prueba","tipoBeneficiario":"Distrit
o","nombreBeneficiario":"Juan
Perez","historiaClinica":null,"ruc":"0651475102","dni":null,"pasaporte":null,"direc
cion":"Otavalo","telefono":"0984849254","correo":"prueba@prueba.com","responsable":
"Sairy","usuarioCrea":"Sairy Segovia","fechaCrea":"2021-09-
23:00:00:00","usuarioModifica":"Sairy Segovia","fechaModifica":"2022-02-
03:18:16:01","estado":0},
{"idBeneficiario":2,"idPrograma":1,"programa":"Prueba","tipoBeneficiario":"Distrito
","nombreBeneficiario":"Distrito","historiaClinica":null,"ruc":"1007453698","dni":n
ull,"pasaporte":null,"direccion":"Otavalo","telefono":"0984657134","correo":"admini
strador@prueba.com","responsable":"Sairy","usuarioCrea":"Sairy
Segovia","fechaCrea":"2021-09-22:00:00:00","usuarioModifica":"Sairy
Segovia","fechaModifica":"2021-09-22:00:00:00","estado":1}]

```

Fuente: Elaboración propia

Figura 11. Actualizar stock pedido

```
1 BEGIN
2 DECLARE stockactual DECIMAL(10,2);
3 DECLARE saldoactual int;
4 SET @concepto :=(SELECT max(idConcepto)
FROM pedido);
5 SET @stockactual :=(SELECT stock FROM
productos WHERE idProducto =
new.idProducto);
6 SET @saldoactual :=(SELECT saldo FROM lote
WHERE idProducto = new.idProducto and
idLote = new.idLote);
7 IF @concepto = 1 then
8 UPDATE productos SET stock = @stockactual
+ new.cantidadDespachada WHERE idProducto
= new.idProducto;
9 UPDATE lote SET saldo = @saldoactual +
new.cantidadDespachada WHERE idProducto =
new.idProducto and idLote = new.idLote;
10 ELSEIF @concepto = 2 then
11 UPDATE productos SET stock = @stockactual
+ new.cantidadDespachada WHERE idProducto
= new.idProducto;
12 UPDATE lote SET saldo = @saldoactual +
new.cantidadDespachada WHERE idProducto =
new.idProducto and idLote = new.idLote;
13 END IF;
14 END
```

Fuente: Elaboración propia

Figura 12. Actualizar stock pedido

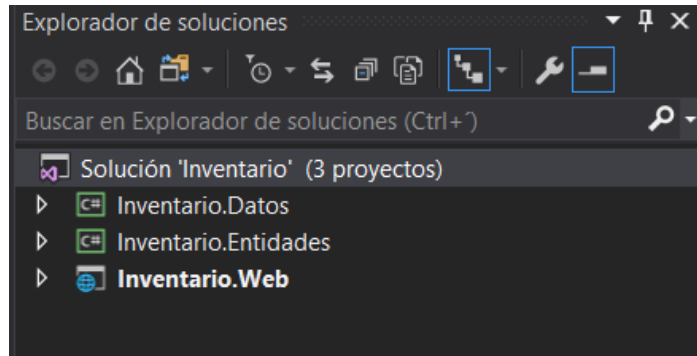
```
1 BEGIN
2 DECLARE pt double(10,2);
3 DECLARE pu double(10,2);
4 DECLARE nc int;
5 DECLARE s int;
6 SET @concepto :=(SELECT idConcepto FROM
boleta ORDER BY idBoleta DESC limit 1);
7 SET @pt :=(SELECT precio * cantidad FROM
detalle_boleta WHERE idProducto =
new.idProducto ORDER BY idDetalleBoleta
desc LIMIT 1);
8 SET @pu :=(SELECT precio FROM
detalle_boleta WHERE idProducto =
new.idProducto ORDER BY idDetalleBoleta
desc LIMIT 1);
9 SET @s :=(SELECT saldo FROM kardex WHERE
new.idLote ORDER BY idKardex desc limit
1);
10 SET @nc :=(SELECT max(numComprobante)
FROM boleta);
11 SET new.precioTotal = @pt;
12 SET new.precioUnitario = @pu;
13 SET new.numComprobante = @nc;
14 IF @concepto = 3 AND @s >= 0 then
15 SET new.saldo = @s - new.cantidad;
16 ELSEIF @concepto = 4 AND @s >= 0 then
17 SET new.saldo = @s + new.cantidad;
18 ELSEIF @concepto = 5 AND @s >= 0 then
19 SET new.saldo = @s + new.cantidad;
20 ELSEIF @nc > 0 THEN
21 SET new.saldo = new.cantidad;
22 END IF;
```

Fuente: Elaboración propia

3.5.4.2. Elaboración del Back End

El patrón de diseño implementado es el MVC (Modelo vista controlador), que consta de tres capas como: Inventario Datos, Inventario Entidades e Inventario Web, que en conjunto son los responsables en acceder a los datos.

Figura 13. Distribución de carpetas del Back End

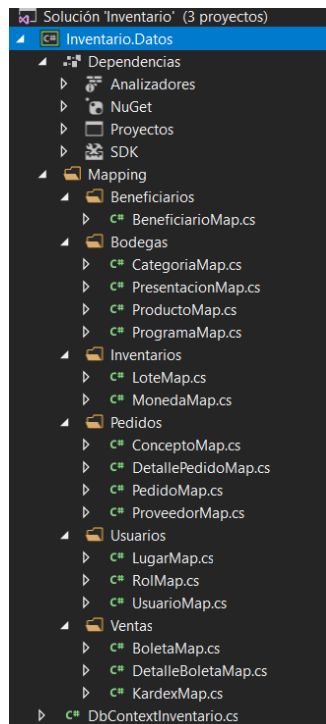


Fuente: Elaboración propia

3.5.4.3. Capa inventario datos

Son los datos que acceden a las entidades, para hacerlos coincidir con sus campos de destino.

Figura 14. Conformación del paquete Inventario _Datos

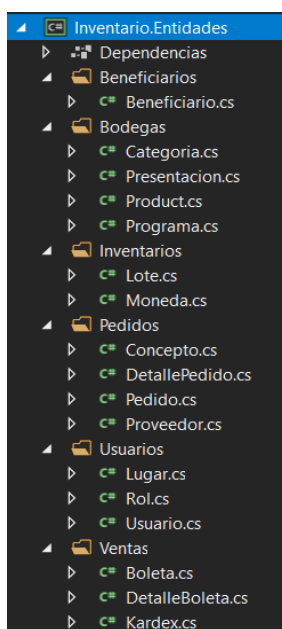


Fuente: Elaboración propia

3.5.4.4. Capa inventario entidades

Contiene las entidades y sus atributos de la base datos, con sus respectivos métodos de acceso get y set.

Figura 15. Conformación de paquete Inventario _Entidades

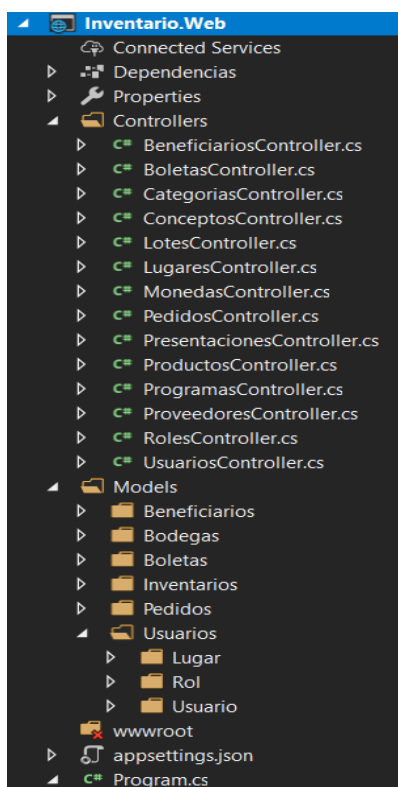


Fuente: Elaboración propia

3.5.4.5. Capa inventario web

Contiene los modelos y los controladores que determinan qué respuesta devolver al usuario cuando realiza una solicitud y por último la conexión a la base de datos en el archivo appsettings.json.

Figura 16. Conformación de paquete Inventario _Web



Fuente: Elaboración propia

Figura 17. Conexión a la base de datos

```
1  {
2    "ConnectionStrings": {
3      "Conexion": "Server=localhost;Database=inventario;uid=root;password="
4    },
5    "Logging": {
6      "LogLevel": {
7        "Default": "Warning"
8      }
9    },
10   "AllowedHosts": "*",
11   "Jwt": {
12     "Key": "Clave Personalizada",
13     "Issuer": "https://localhost:44337/"
14   }
15 }
16
```

Fuente: Elaboración propia

3.5.4.6. Elaboración del Front End

En el front End se diseñó la interfaz de usuario para el acceso a la aplicación. El consumo de la aplicación para la Api Rest se lo realiza a través de los formularios de las vistas que utilizan la librería axios.

3.5.4.7. Vistas del proyecto

En la carpeta components se encuentran archivos en formato Vue, donde se realiza la codificación de las interfaces o vistas, como se presenta en la Figura 18.

Figura 18. Archivos vue de la carpeta components



Fuente: Elaboración propio

Figura 19. Código de la vista login.vue

```
1 <template>
2   <v-layout align-center justify-center>
3     <v-flex xs12 sm8 md6 lg5 xl4>
4       <v-card>
5         <v-toolbar dark color="blue darken-3">
6           <v-toolbar-title>
7             Acceso al Sistema
8           </v-toolbar-title>
9         </v-toolbar>
10        <v-card-text>
11          <v-text-field v-model="correo" autofocus color="accent" label="Correo" required></v-text-field>
12          <v-text-field v-model="contrasena" type="password" color="accent" label="Contraseña" required></v-text-field>
13          <v-flex v-if="error" class="red--text">
14            {{error}}
15          </v-flex>
16        </v-card-text>
17        <v-card-actions class="px-3 pb-3">
18          <v-flex text-xs-right>
19            <v-btn @click="ingresar()" color="primary">Iniciar Sesión</v-btn>
20          </v-flex>
21        </v-card-actions>
22      </v-card>
23    </v-flex>
24  </v-layout>
25 </template>
26 <script>
27 import axios from 'axios'
28 export default {
29   data() {
30     return{
31       correo:'',
32       contrasena:',
33       error:null
34     }
35   },
36   methods:{
37     ingresar(){
38       this.error=null;
39       axios.post('api/Usuarios/Login',{correo:this.correo,password:this.contrasena})
40         .then(respuesta => {
41           return respuesta.data
42         })
43         .then(data => {
44           this.$store.dispatch("guardarToken", data.token)
45           this.$router.push({name: 'home'})
46         })
47         .catch(err =>{
48           //console.log(err.response)
49           if (err.response.status==400) {
50             this.error="No es un correo válido";
51           }else
52           if (err.response.status==404) {
53             this.error="correo o contraseña incorrectos";
54           }else{
55             this.error="Ocurrio un error";
56           }
57         })
58     }
59   }
60 }
61 </script>
```

Fuente: Elaboración propia

3.5.4.8. Configuración Service Worker

Permite interceptar y controlar los requests o peticiones de la red y el almacenamiento del cache del navegador, permitiendo al usuario tener la experiencia de trabajar en modo offline, sin conexión a internet.

Figura 20. Configuración del archivo Service Worker

```
src > JS registerServiceWorker.js > ...
1  |/* eslint-disable no-console */
2
3  import { register } from 'register-service-worker'
4
5  if (process.env.NODE_ENV === 'production') {
6    register(`${process.env.BASE_URL}service-worker.js`, {
7      ready () {
8        console.log(
9          'App is being served from cache by a service worker.\n' +
10         'For more details, visit https://goo.gl/AFskqB'
11        )
12      },
13      registered () {
14        console.log('Service worker has been registered.')
15      },
16      cached () {
17        console.log('Content has been cached for offline use.')
18      },
19      updatefound () {
20        console.log('New content is downloading.')
21      },
22      updated () {
23        console.log('New content is available; please refresh.')
24      },
25      offline () {
26        console.log('No internet connection found. App is running in offline mode.')
27      },
28      error (error) {
29        console.error('Error during service worker registration:', error)
30      }
31    })
32  }
33
```

Fuente: Elaboración propia

3.5.4.9. Configuración Manifest

Esta tecnología permite añadir funcionalidades para que el proyecto sea más dinámico, como: definir el nombre de la aplicación, especificar el color del fondo de la aplicación y en especial agregar un icono a la aplicación. La configuración de este archivo permite que el proyecto pueda ejecutarse en segundo plano, sin la necesidad de estar dentro de un navegador, podrá ser instalado en la pantalla del ordenador de donde se acceda.

Figura 21. Configuración del archivo Manifest

```
{ } manifest.json U X
public > { } manifest.json > ...
1  {
2    "short_name": "Inventario",
3    "name": "Inventario: Coordinacion de salud zona 3",
4    "icons": [
5      {
6        "src": "/imagenes/zona1.png",
7        "type": "image/png",
8        "sizes": "192x192"
9      },
10     {
11       "src": "/imagenes/zona2.png",
12       "type": "image/png",
13       "sizes": "512x512"
14     }
15   ],
16   "start_url": "/",
17   "background_color": "#3367D6",
18   "display": "standalone",
19   "scope": "/",
20   "theme_color": "#3367D6"
21 }
22 }
```

Fuente: Elaboración propia

3.5.5. Pruebas y entrega

Se realizó las pruebas de rendimiento para evaluar cómo se comporta la aplicación bajo ciertas condiciones específicas de carga, medidas a través de la norma ISO/IEC 25023, con la herramienta Apache JMeter. En la última fase se realizó entrega del sistema con el diseño final de las interfaces.

3.5.5.1. Pruebas de carga

Este tipo de prueba se lo realizó bajo una cantidad esperada de peticiones que simulan ser los usuarios recurrentes en un intervalo tiempo o un solo bucle de tiempo.

Los PATHS que fueron sometidos a las pruebas de carga son: login, productos, transferencias, cuentas contables, comprobantes o boletas y lotes, a fin de identificar cuellos de botella del aplicativo para gestionar de mejor manera cada módulo, con el propósito de que el software pueda soportar más usuarios en un determinado intervalo de tiempo.

A continuación, se presenta el modo de configuración para las pruebas de carga en JMeter, se identifican los parámetros: users representa los usuarios recurrentes, seconds el tiempo de carga e infinite que representa el número de veces a ejecutarse.

Figura 22. Grupo de usuarios recurrentes para pruebas de carga

The screenshot shows the 'Thread Group' configuration window. The 'Name' field is 'Usuario recurrentes'. The 'Comments' field is empty. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. The 'Thread Properties' section includes: 'Number of Threads (users):' set to 3; 'Ramp-up period (seconds):' set to 1; 'Loop Count:' with 'Infinite' unchecked and '1' entered; 'Same user on each iteration' checked; 'Delay Thread creation until needed' unchecked; 'Specify Thread lifetime' unchecked; 'Duration (seconds):' empty; and 'Startup delay (seconds):' empty.

Fuente: Elaboración propia

3.5.5.2. Pruebas de estabilidad o soak testing

En este apartado se procede a realizar el soak testing a fin de probar si el aplicativo soporta una carga continua en un bucle infinito, Figura 23.

Figura 23. Pruebas de estabilidad

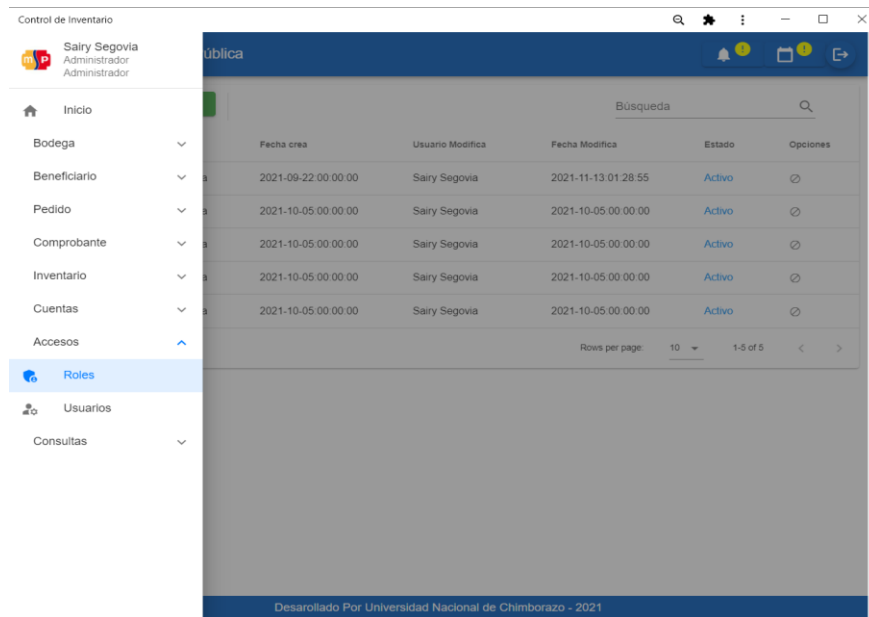
The screenshot shows the 'Thread Group' configuration window. The 'Name' field is 'Grupo de usuarios'. The 'Comments' field is 'Carga continua'. Under 'Action to be taken after a Sampler error', the 'Continue' radio button is selected. The 'Thread Properties' section includes: 'Number of Threads (users):' set to 100; 'Ramp-up period (seconds):' set to 1; 'Loop Count:' with 'Infinite' checked; 'Same user on each iteration' checked; 'Delay Thread creation until needed' unchecked; 'Specify Thread lifetime' unchecked; 'Duration (seconds):' empty; and 'Startup delay (seconds):' empty.

Fuente: Elaboración propia

3.5.5.3. Diseño final de la interfaz

A continuación, se observa algunos de los diseños finales de las interfaces de administrador, distritos, unidades operativas y roles de usuario. Estos diseños contienen diferentes opciones en el menú asociado de acuerdo con el rol establecido, cada ventana identificará: nombre de usuario, ubicación y rol.

Figura 24. Vista administrador



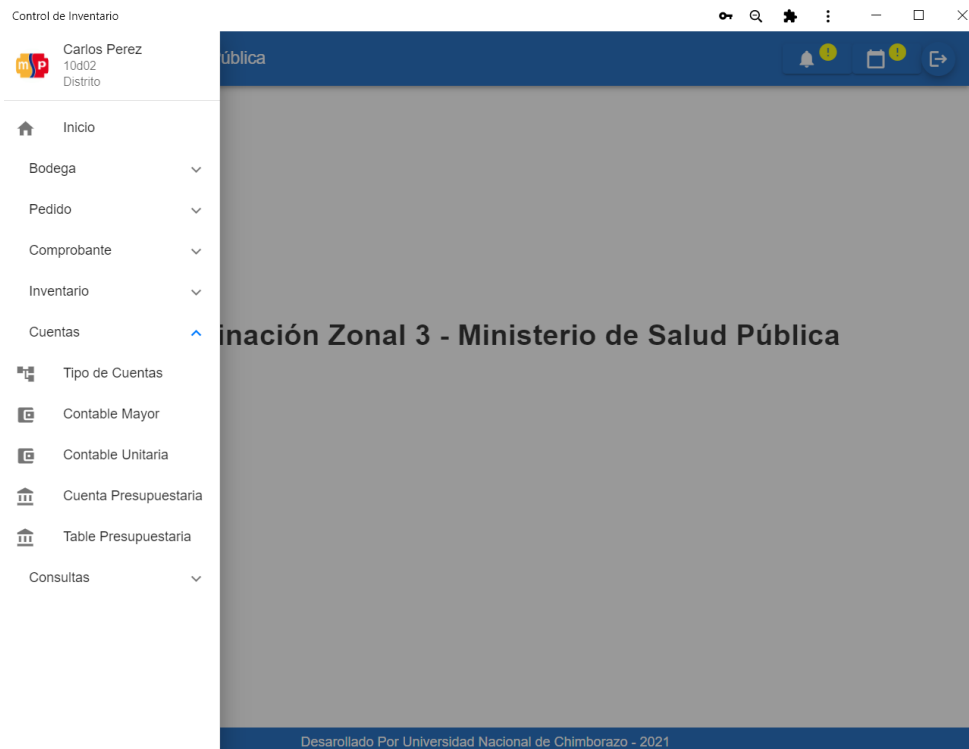
Fuente: Elaboración propia

Figura 25. Vista bodega general



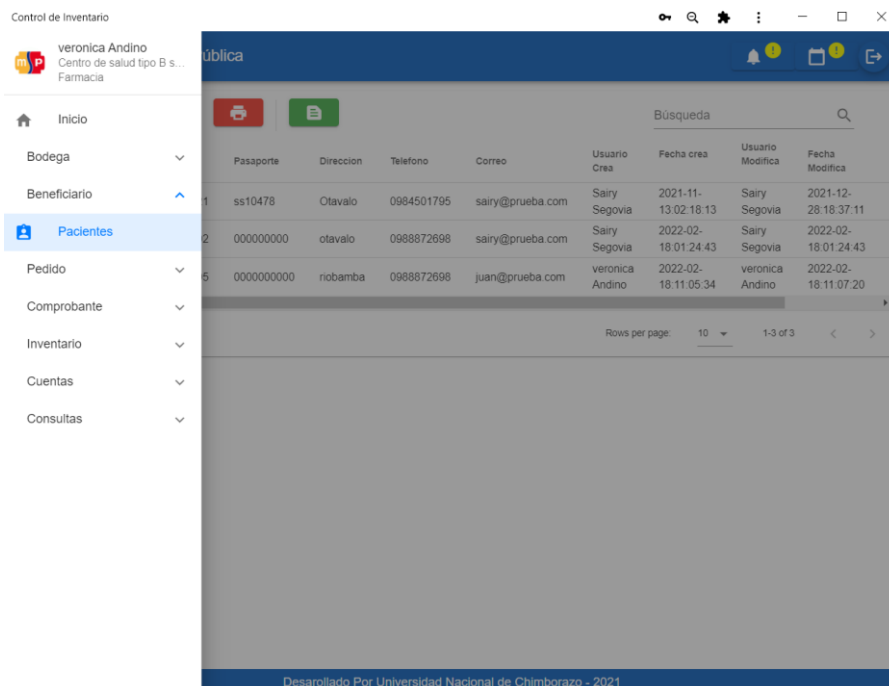
Fuente: Elaboración propia

Figura 26. Vista distrito



Fuente: Elaboración propia

Figura 27. Vista unidad operativa – farmacia



Fuente: Elaboración propia

Figura 28. Vista médico

Control de Inventario

Raul Aldaz
Centro de salud tipo B s...
Medico

ública

Búsqueda

| Proveedor | Presentación | Forma Farmacéutica | Unidad Contenido | Programa | Proyecto | Grupo | Cuenta presupuestaria | Cuenta contable |
|---------------|--------------|--------------------|------------------|----------------|------------|---------|-----------------------|-----------------|
| Chino sinofar | 3ml | Solido | 1 g | nuevo programa | 002proyect | 01Grupo | 530809 | 1310109 |

Rows per page: 10 1-1 of 1

Desarrollado Por Universidad Nacional de Chimborazo - 2021

Fuente: Elaboración propia

CAPÍTULO IV

4. RESULTADOS Y DISCUSION

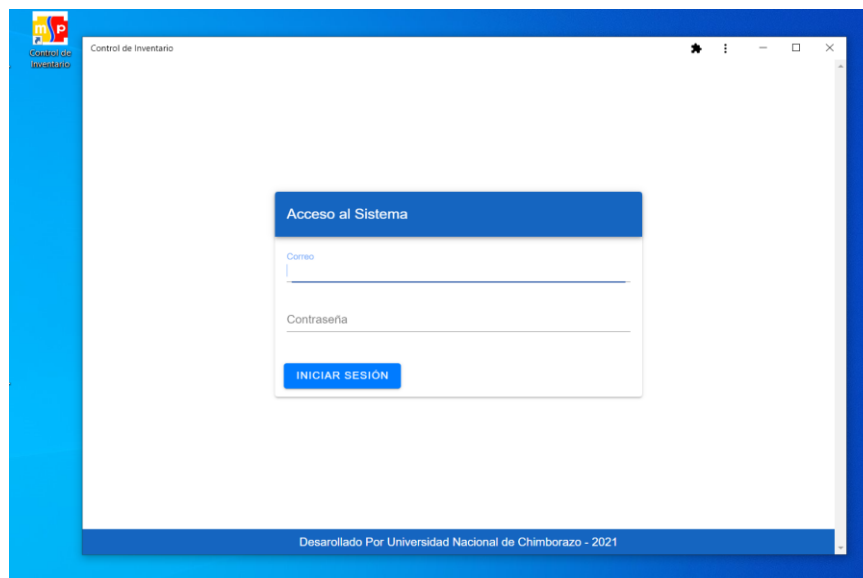
4.1. Resultados

4.1.1. Una aplicación PWA para el manejo de inventario.

Las aplicaciones Web Progresivas según (Pérez, 2021) son una clara evolución de las aplicaciones web tradicionales, realizando tareas que normalmente solo las aplicaciones nativas podían llevar a cabo, convirtiéndose en una opción óptima para mejorar el manejo de medicamentos en la farmacia de la Coordinación de Salud Zona 3. El conjunto de tecnologías actualizadas le permiten ser una aplicación escalable y moderna, para su utilización en cada uno de los departamentos relacionados, bodega, unidad operativa, distrito y roles de usuario.

Para que la aplicación funcione como una web progresiva, se incorporó el archivo manifest.json, que facilita la selección del nombre e icono de la aplicación, para que sea instalable en el dispositivo que se esté utilizando, sin importar el sistema operativo que tenga instalado, además archivos en formato java script como el service worker para que el software trabaje de manera offline, estas dos características son fundamentales para la creación de una aplicación web progresiva. La figura 29, muestra la aplicación ejecutándose como una web progresiva, tanto en el ordenador como en el dispositivo móvil.

Figura 29. Aplicación PWA ejecutándose en el ordenador



Fuente: Elaboración propia

Figura 30. Aplicación PWA ejecutándose en un dispositivo móvil



Fuente: Elaboración propia

4.1.2. JMeter como herramienta de testing

Es una herramienta que permite realizar pruebas de rendimiento a una aplicación web, con el fin de determinar los cuellos de botella de la aplicación, para posteriormente, una vez identificadas, puedan ser tratadas como, por ejemplo: eliminando código innecesario y reestructurando los métodos que contienen cada módulo.

Los resultados obtenidos se evalúan conforme las métricas de la ISO/IEC 25023 y Google, que se enfocan en las propiedades medibles del desempeño o rendimiento del software bajo ciertas condiciones, donde los tiempos de respuesta más cercanos a cero es mejor y debe estar por debajo de los 200 ms.

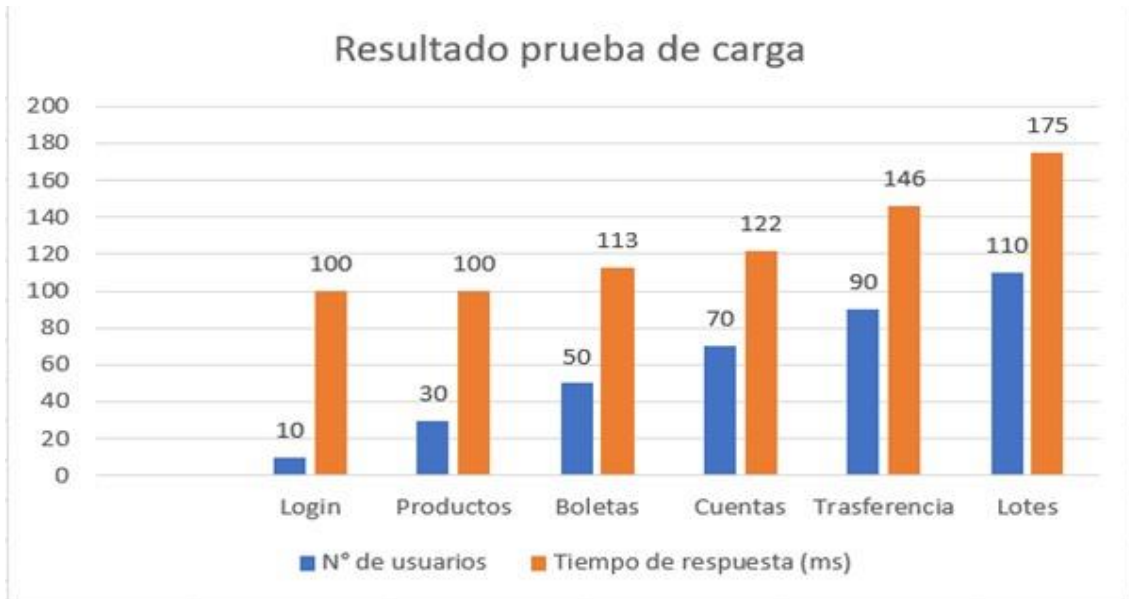
Tabla 31. Resultados de la prueba de carga

| Path | N° de usuarios | Cálculos $X = B - A$ | Tiempo de respuesta | Estado |
|------------------|----------------|---------------------------|---------------------|--------|
| Login | 10 | $X=316-216= 100\text{ms}$ | 100ms | OK |
| Productos | 30 | $X=379-279=100\text{ms}$ | 100ms | OK |
| Boletas | 50 | $X=829-716=113\text{ms}$ | 113ms | OK |
| Cuentas | 70 | $X=346-224=122\text{ms}$ | 122ms | OK |

| | | | | |
|---------------------|-----|--------------------------|-------|----|
| Trasferencia | 90 | $X=422-276=146\text{ms}$ | 146ms | OK |
| Lotes | 110 | $X=403-228=175\text{ms}$ | 175ms | OK |

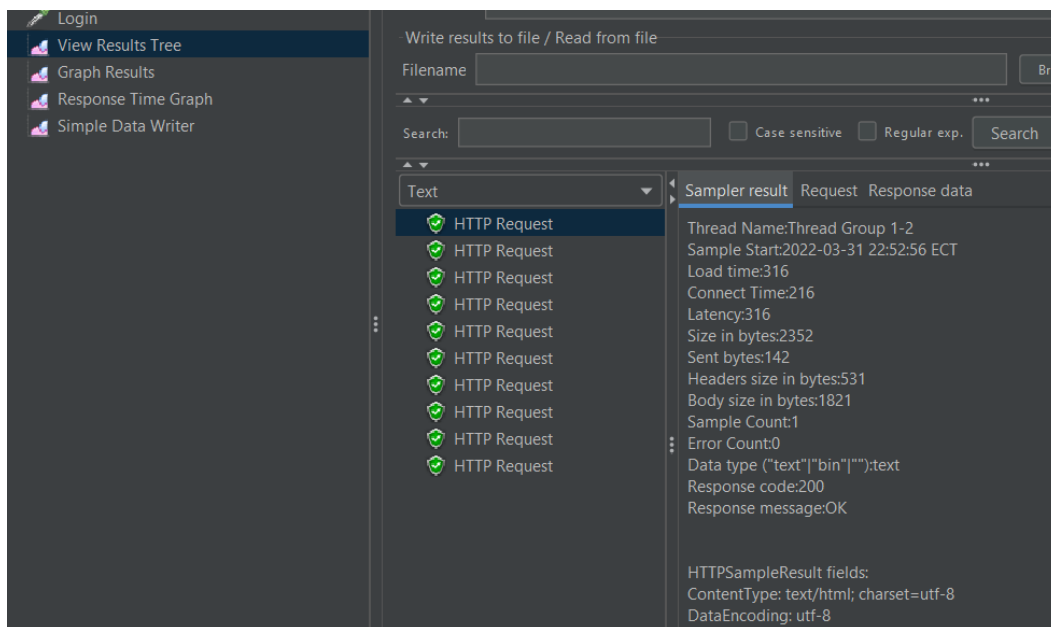
Fuente: Elaboración propia

Figura 31. Resultado pruebas de carga



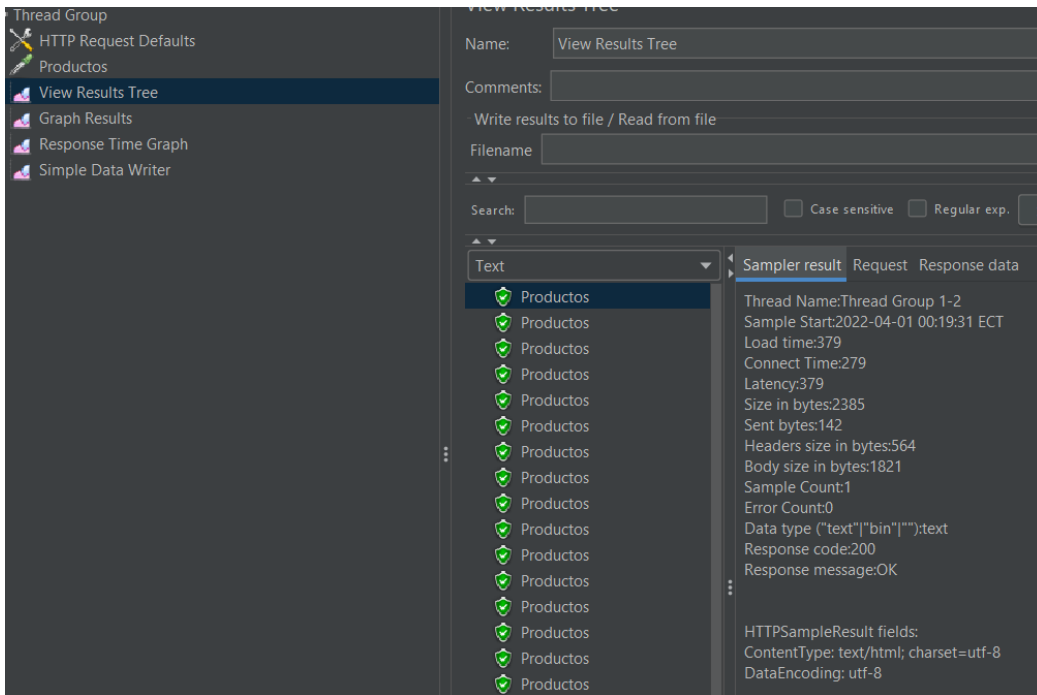
Fuente: Elaboración propia

Figura 32. Resultados de las pruebas de carga al path Login



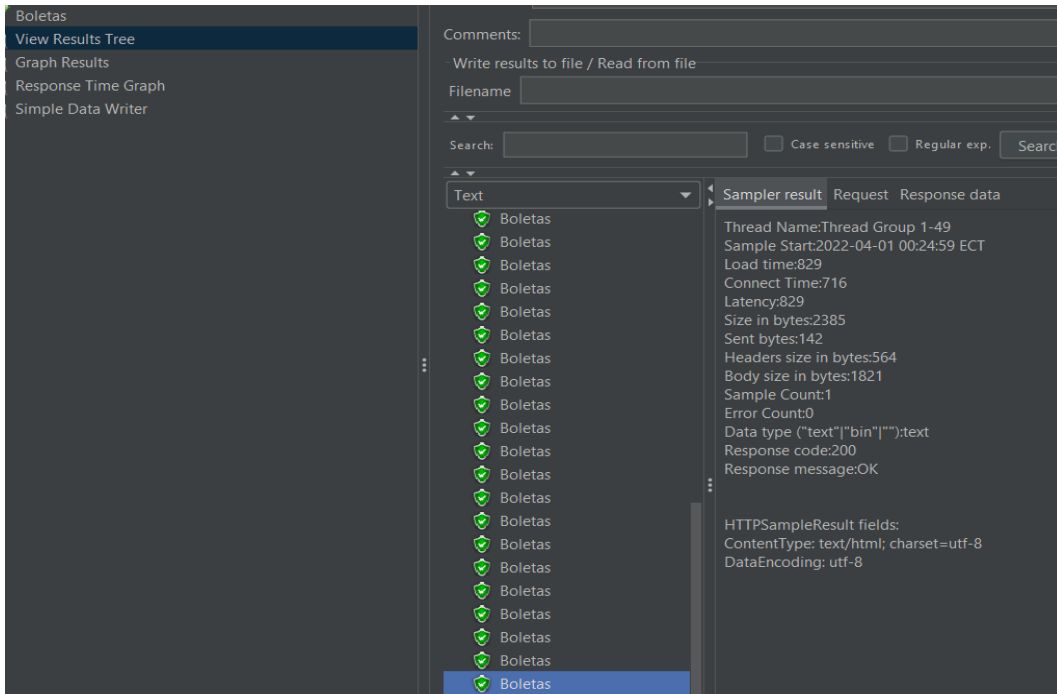
Fuente: Elaboración propia

Figura 33. Resultados de las pruebas de carga al path Productos



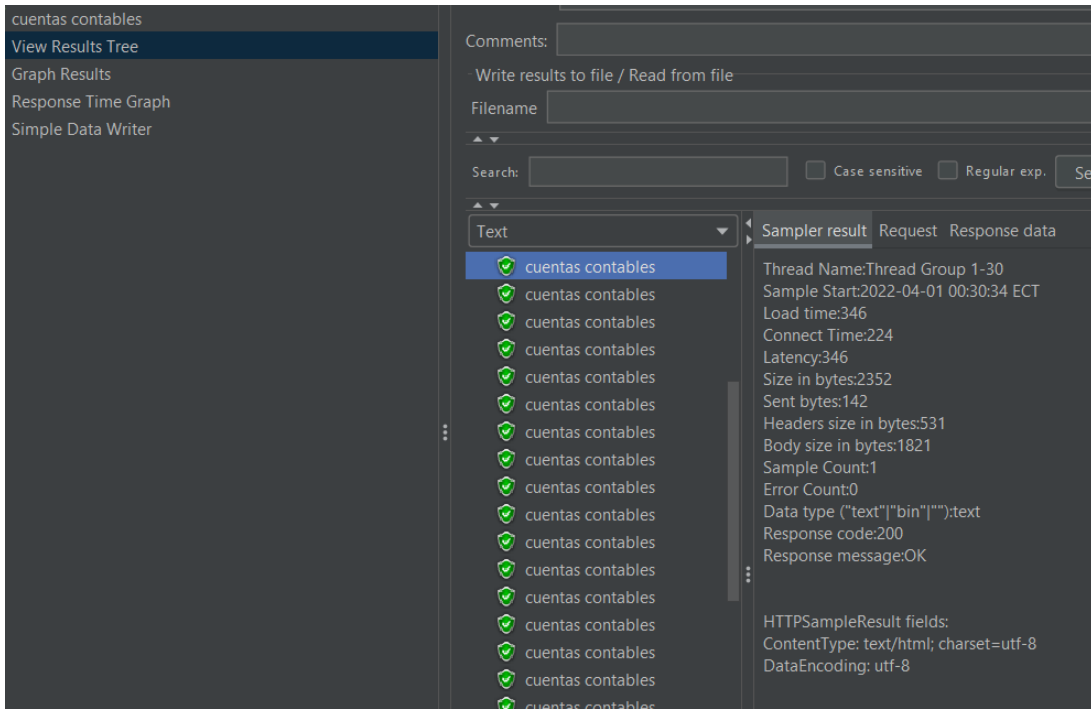
Fuente: Elaboración propia.

Figura 34. Resultados de las pruebas al path Boletas



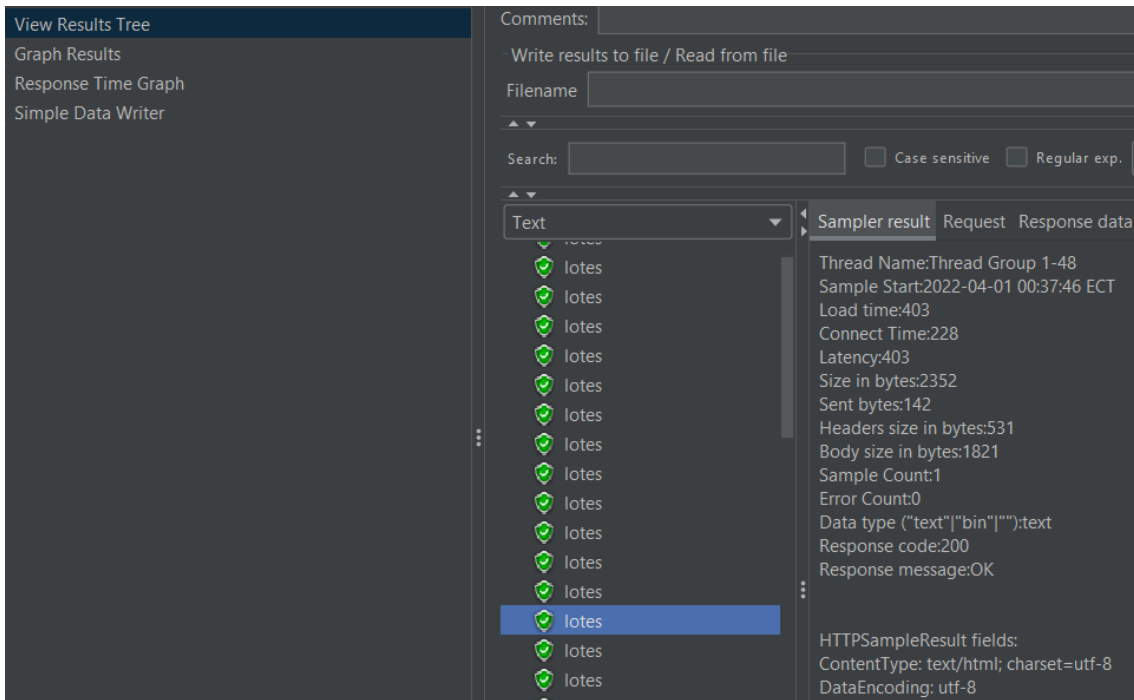
Fuente: Elaboración propia

Figura 35. Resultados de las pruebas de carga al path Cuentas



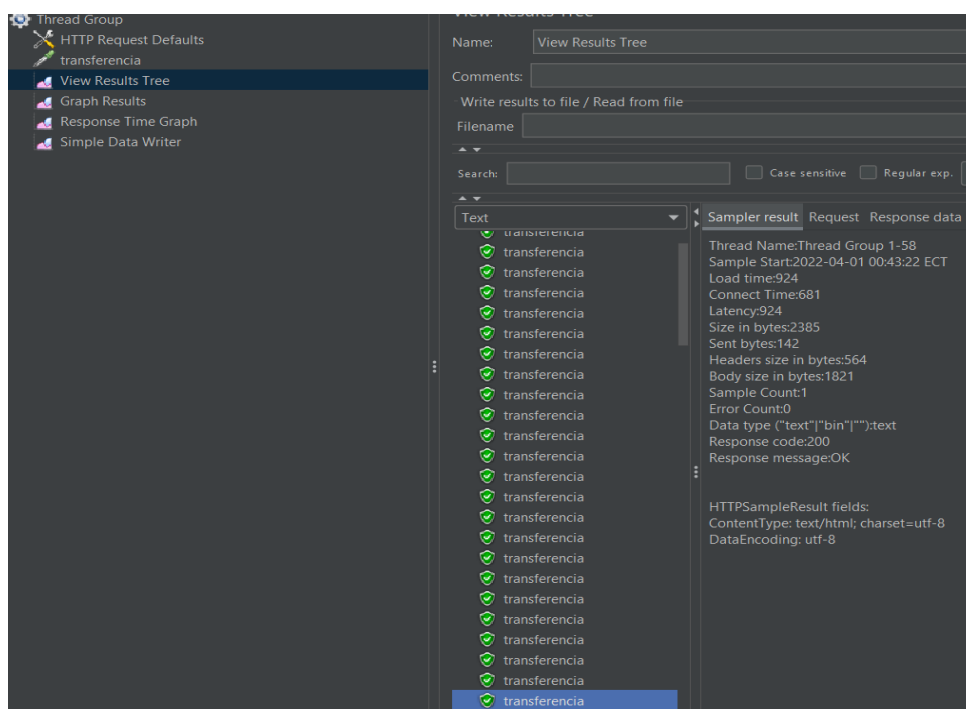
Fuente: Elaboración propia

Figura 36. Resultados de las pruebas de carga al path Lotes



Fuente: Elaboración propia

Figura 37. Resultados de las pruebas de carga al path Transferencias



Fuente: Elaboración propia

En las pruebas de carga se obtiene los resultados presentados en la tabla 31, se evidencian resultados favorables incluso con 110 usuarios recurrentes, los tiempos de mayor retardo que se obtuvieron son de 146 ms a 175 ms, estos valores de acuerdo con la métrica de evaluación están por debajo de los 200 ms, que es el valor aceptable de acuerdo con la norma ISO/IEC 25023 y la métrica de evaluación de Google.

Para la prueba de estabilidad se realiza con la página principal del sistema, el path home, con 100 usuarios en un minuto, pero esta vez en un bucle infinito, es decir, se repetirá cuantas veces sea posible las solicitudes hasta que el usuario interrumpa dicha acción, esto se lo realiza para comprobar si el sistema es estable o tiene fugas de memoria. Obteniendo resultados óptimos, como se puede apreciar en la figura 37, donde a partir del segundo intervalo se mantiene sin presentar variaciones considerables.

permite realizar ingresos, egresos, transferencias, reportes, notificaciones de control de stock y caducidad, para un correcto manejo de inventarios.

La Aplicación Web Progresiva trabaja independientemente del navegador ejecutándose en segundo plano, permitiendo utilizarlas como aplicaciones nativas y en modo offline, a diferencia de las aplicaciones web tradicionales que no presentan estas características.

El uso de la metodología RAD según (Castro, 2019) permite crear aplicaciones de forma rápida y económica para satisfacer las necesidades empresariales sin invertir tanto tiempo y dinero. Gracias al desarrollo rápido de aplicaciones, el software es bastante maleable, permitiendo realizar futuras adaptaciones a los prototipos o iteraciones.

Se desarrolló la aplicación web progresiva para el manejo de inventario de la farmacia de la Coordinación de Salud utilizando la metodología Desarrollo Rápido de Aplicaciones. Para evaluar el rendimiento del aplicativo se empleó métricas de la ISO/IEC 25023, que se enfocan en las pruebas de rendimiento de carga y estabilidad, estos tipos de pruebas son útiles para simular el acceso de usuarios recurrentes a los diferentes módulos o path del sistema. El aplicativo se ajusta a las necesidades para un correcto manejo de inventario en la Coordinación de Salud Zona 3. Al analizar los resultados de las pruebas de carga y estabilidad se obtuvieron buenos resultados, evidenciando que los tiempos de respuesta a las peticiones de los usuarios están por debajo del rango establecido de 200 ms, con un máximo de 110 usuarios, resultando adecuado para que sea utilizado en la Coordinación de Salud, organismo que cuenta con 24 unidades operativas o farmacias. Por lo que en base a las directrices y las pruebas realizadas presenta un rendimiento apropiado.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- A través de un estudio minucioso de las aplicaciones web progresivas se determinó las funcionalidades de las tecnologías modernas que estas incorporan, destacándose el archivo Manifest.json, responsable de que la aplicación se ejecute en segundo plano, característica que le permite funcionar como una aplicación nativa sin importar el tipo de sistema operativo del dispositivo u ordenador con el que se accede.
- Se desarrolló el aplicativo web progresivo para el manejo de inventario en la farmacia de la Coordinación de Salud Zona 3, utilizando la metodología de desarrollo de software RAD (Desarrollo rápido de aplicaciones), que permite crear sistemas estratégicamente importantes de manera más rápida, se utilizó MySQL como base de datos, la librería IU (interfaz de usuario) de Vue Js que facilita la configuración de la estructura de la aplicación web progresiva en modo gráfico y Axios para el consumo de los servicios, optimizando así el tiempo de desarrollo del software.
- Con las pruebas de carga y estabilidad se determinó que el sistema presenta un rendimiento favorable, soportando una carga de nivel alto de hasta 110 usuarios recurrentes por minuto, obteniendo tiempos de respuesta que están por debajo de los 200 ms, valor aceptable bajo las métricas de la norma ISO/IEC 25023 y de Google.

5.2. Recomendaciones

- Para analizar el desarrollo de las aplicaciones web progresivas se recomienda investigar el conjunto de técnicas que estas incorporan con un sustento científico, para comprender de mejor forma las tecnologías que permiten crear este tipo de aplicaciones.
- El proyecto de investigación desarrollado al satisfacer una necesidad de la Coordinación de Salud Zona 3, se proyecta con la posibilidad de seguir creciendo en módulos adicionales, por lo cual se recomienda que futuras adecuaciones manejen el mismo estándar de desarrollo en el Front End y Web Services en el Back End.

- Los resultados obtenidos de las pruebas de rendimiento presentan tiempos de respuesta aceptables de acuerdo con las métricas establecidas por Google y la norma ISO/IEC 25023, para futuros trabajos se recomienda aplicar otros tipos de pruebas adicionales por ejemplo la de estrés, con la finalidad de comprobar si el sistema soporta una carga por encima de lo esperado, bajo determinadas circunstancias.

BIBLIOGRAFÍA

- Arbeláez , O., Medina , F., & Chaves, A. (2015). *HERRAMIENTAS PARA EL DESARROLLO RÁPIDO DE APLICACIONES WEB*. Obtenido de <http://metodologiarad.weebly.com/uploads/1/5/6/7/15678332/511-261-1-pb.pdf>
- Aguirre, V., Ortu, A., & Delía, L. (2019). PWA para unificar el desarrollo Desktop, Web y Mobile. *XXV Congreso Argentino de Ciencias de la Computación* , 9.
- Arcos-Medina, G. M. (30 de 01 de 2018). *Knowledge E*. Obtenido de Knowledge E: <https://www.knepublishing.com/index.php/KnE-Engineering/article/view/1498>
- Barzanallana, R. (12 de 02 de 2019). *Páginas web dinámicas*. Obtenido de <https://webcache.googleusercontent.com/search?q=cache:oaigQ-ThwQkJ:https://www.um.es/docencia/barzana/DAWEB/2017-18/daweb-tema-13-paginas-web-dinamicas.html+&cd=2&hl=es&ct=clnk&gl=ec>
- Campaña, R. (04 de 2015). *El proceso de desarrollo rápido de aplicaciones (DRA) de software*. Obtenido de https://www.researchgate.net/publication/303839299_El_proceso_de_desarrollo_rapido_de_aplicaciones_DRA_de_software_Un_aporte_practico_en_el_Instituto_Geografico_Militar
- Castro, M. (12 de 12 de 2019). *Incentro*. Obtenido de <https://www.incentro.com/es-ES/blog/metodologia-rad-desarrollo-rapido-aplicaciones>

- Díaz, Y., & Fernández, Y. (12 de 06 de 2015). *Patrón Modelo-Vista-Controlador*. Obtenido de <https://revistatelematica.cujae.edu.cu/index.php/tele/article/view/15>
- Halili, E. (2015). *Apache JMeter*. Mithil Kulkarni.
- Hernández, C. (02 de 05 de 2018). *Servicios REST*. Obtenido de <http://repositorio.konradlorenz.edu.co/handle/001/134>
- Hota, A., & Prabhu, M. (2016). Node Js. *Technology Update*, 2.
- Hugon, J. (2018). *C# 7: desarrolle aplicaciones Windows con Visual Studio 2017*. Barcelona: ENI.
- Kyriakidis, A., Maniatis, K., & You, E. (2017). *The Majesty of Vue.js 2*. leanpub.
- Lee, G. (04 de 11 de 2020). *loadview*. Obtenido de Pruebas de rendimiento, Consejos técnicos: <https://www.loadview-testing.com/es/blog/5-ejemplos-de-pruebas-de-carga-de-jmeter/#:~:text=JMeter%20es%20una%20herramienta%20que,identificar%20la%20capacidad%20de%20carga>.
- Lopez, S. (5 de Junio de 2020). *Digital55*. Obtenido de Digital55: <https://www.digital55.com/desarrollo-tecnologia/que-es-pwa-ventajas-desventajas/>
- Osmani, A. (14 de 05 de 2019). *El modelo de "shell de app"*. Obtenido de <https://developers.google.com/web/fundamentals/architecture/app-shell#app-shell-requirements>
- Peláez, B. (21 de 1 de 20220). *¿Qué es el Desarrollo rápido de aplicaciones (RAD)?* Obtenido de <https://www.capterra.es/blog/1218/que-es-el-desarrollo-rapido-de-aplicaciones-rad>
- Perez, J. E. (2019). *Introduccion a Java Script*.
- Pérez, S. D. (21 de 10 de 2021). *Intelequia*. Obtenido de <https://intelequia.com/blog/post/3028/qu%C3%A9-es-una-aplicaci%C3%B3n-web-progresiva>
- Piñero, G. M., Marin, D. A., Trujillo, C. Y., & Buedo, H. D. (15 de 01 de 2021). *Buenas prácticas para prevenir los riesgos de la eficiencia del desempeño en los productos de software*. Obtenido de <https://www.redalyc.org/journal/3783/378366538006/html/#B6>
- Ranchal, J. (26 de 09 de 2019). *Qué son, cómo funcionan y ejecutan las aplicaciones web progresivas*. Obtenido de <https://www.muycomputer.com/2019/09/26/aplicaciones-web-progresivas/>
- Sánchez, J. F. (02 de 07 de 2018). *Pruebas de rendimiento con JMeter*. Obtenido de [https://www.sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos#:~:text=PRUEBA%20DE%20ESTABILIDAD%20\(SOAK%20TESTING,de%20memoria%20en%20la%20aplicaci%C3%B3n](https://www.sdos.es/blog/pruebas-de-rendimiento-con-jmeter-ejemplos-basicos#:~:text=PRUEBA%20DE%20ESTABILIDAD%20(SOAK%20TESTING,de%20memoria%20en%20la%20aplicaci%C3%B3n).
- Santillan, C. (2014). Bases de datos en MySQL.
- Sharo, Z. (2020). *Maintenance Management System*.
- Shirivo. (30 de 06 de 2015). *Shirivo*. Obtenido de Shirivo: <https://shirivo.wordpress.com/2015/06/30/modelo-vista-vista-modelo-eso-es-un-patron/>

- SISINGGROUP. (03 de Abril de 2017). Obtenido de Metodologia RAD:
<https://sisingblog.wordpress.com/2017/04/03/metodologia-rad/>
- Smith, S. (30 de 09 de 2017). *Overview of ASP.NET Core MVC*. Obtenido de
<https://studreadywork.ru/wp-content/uploads/2021/09/Overview-of-ASP.NET-Core-MVC.pdf>
- Tandel, S. S., & Jamadar, A. (2018). Impact of Progressive Web Apps on Web App. *IJIRSET*, 6.
- Villa, J. G. (05 de 09 de 2017). *Reducir el tiempo de respuesta del servidor*. Obtenido de
<https://useo.es/reducir-el-tiempo-de-respuesta-del-servidor/#:~:text=Para%20que%20los%20usuarios%20realmente,debajo%20de%20los%20200%20milisegundos.>
- Zea, J. T. (07 de 10 de 2016). *Progmatt*. Obtenido de Progmatt:
<http://www.progmatt.uaem.mx:8080/Vol8num3/vol8num3art7.pdf>

ANEXOS

Anexo 1: Manual de usuario

MANUAL DE USUARIO

SOFTWARE DE CONTROL DE INVENTARIO PARA LA FARMACIA DE LA COORDINACIÓN DE SALUD ZONA 3

Versión 1



Nota: Esta aplicación puede ser instalada desde cualquier navegador independientemente del sistema operativo (Windows, Android, IOS o Linux) del dispositivo con el que se accede.

I. Ingreso al sistema

Ingresar a la aplicación, visualizara la pantalla inicial como presenta la Figura 39.

Figura 40. Formulario acceso al sistema

Desarrollado Por Universidad Nacional de Chimborazo - 2021

Fuente: Elaboración propia

Ingresar correo y contraseña, esto dependiendo del rol al que pertenezca, el usuario por defecto será el administrador, con las siguientes credenciales:

Correo: administrador@prueba.com

Contraseña: Administrador1

Dar clic en el botón.



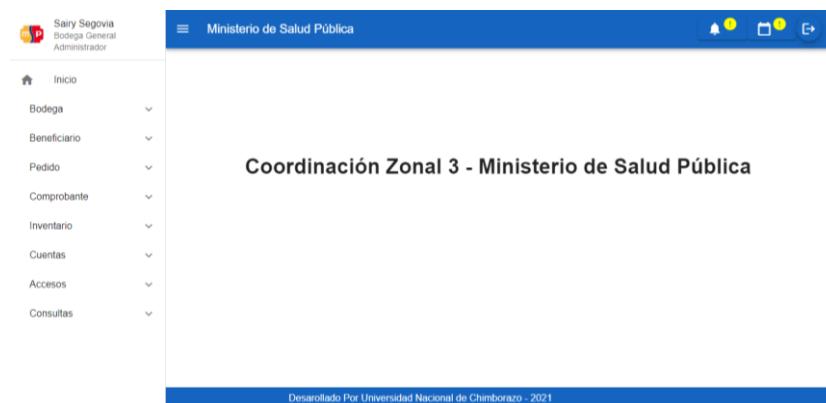
Figura 41. Formulario iniciar sesión

Una captura de pantalla de un formulario web. El título es "Acceso al Sistema" en un encabezado azul. Hay dos campos de entrada: "Correo" con el valor "administrador@prueba.com" y "Contraseña" con caracteres ocultos por puntos. Debajo de los campos hay un botón azul con el texto "INICIAR SESIÓN".

Fuente: Elaboración propia

Se visualiza la pantalla principal del administrador.

Figura 42. Menú principal



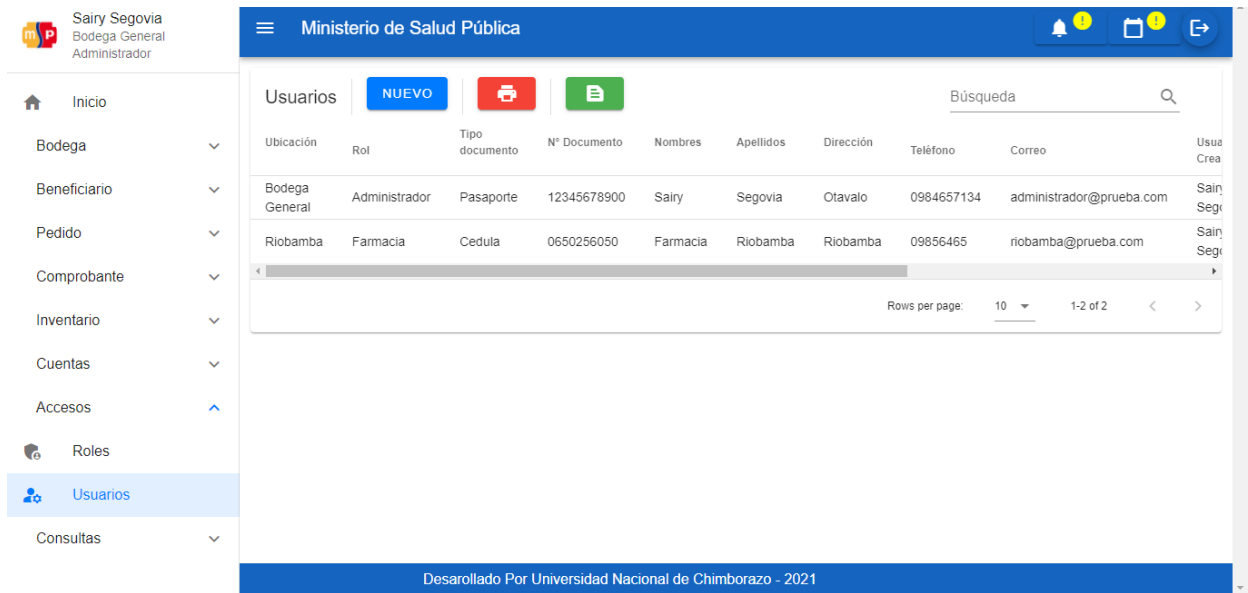
Fuente: Elaboración propia

II. Registro de nuevos usuarios

Para registrar un nuevo usuario dar clic en la opción “Accesos”, se desplegarán dos opciones (Roles y Usuarios), escoger “Usuarios” y dar clic en el botón.



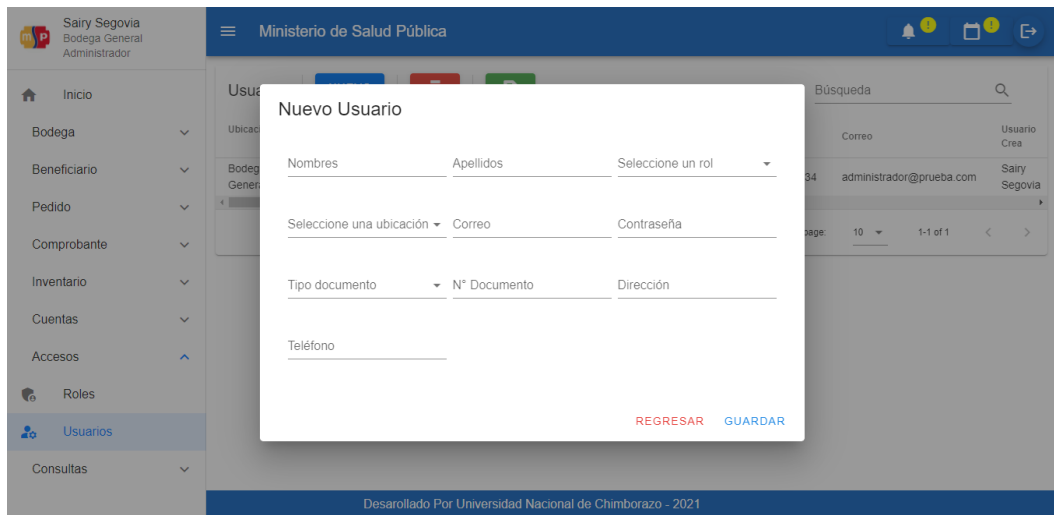
Figura 43. Página de Usuarios



Fuente: Elaboración propia

Se presenta el formulario de nuevo ingreso.

Figura 44. Módulo de registro de usuarios



Fuente: Elaboración propia

Llenar todos los campos correctamente, ingresar Nombres, Apellidos, seleccionar un rol, seleccionar una ubicación, ingresar el correo (Mediante este correo se ingresará al sistema), ingresar una contraseña (Como mínimo 8 caracteres, 1 letra Mayúscula, 1 letra minúscula y 1 número), escoger el tipo de documento (Cédula-Pasaporte), el N° documento (Deberá ser una cédula válida), ingresar la dirección y teléfono. Llenar como se muestra a continuación y hacer clic en el botón.

GUARDAR

Figura 45. Registro de datos del usuario

The screenshot shows a web application interface for 'Ministerio de Salud Pública'. A modal window titled 'Nuevo Usuario' is open, displaying a form with the following fields and values:

| | | | | | |
|--------------------------|----------------|--------------|----------------------|-------------------|----------|
| Nombres | Flavio | Apellidos | Tapuy | Seleccione un rol | Bodega |
| Seleccione una ubicación | Bodega General | Correo | flaviotswl@gmail.com | Contraseña | ***** |
| Tipo documento | Cedula | N° Documento | 1501051682 | Dirección | Riobamba |
| Teléfono | 0998421154 | | | | |

At the bottom of the form are two buttons: 'REGRESAR' (red) and 'GUARDAR' (blue). The background shows a sidebar menu with 'Usuarios' selected and a table with columns for 'Correo' and 'Usuario Crea'.

Fuente: Elaboración propia

III. Registro de monedas

En el menú de opciones seleccionar “Inventario” y dar clic en el botón.

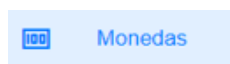


Figura 46. Página de registro de monedas

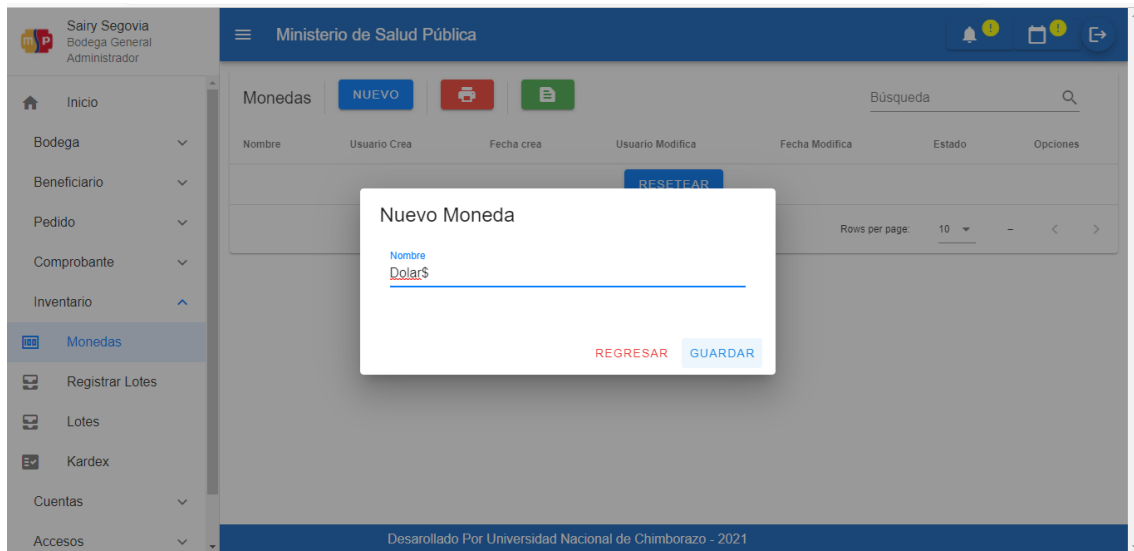
The screenshot shows the 'Monedas' page in the Sairy Segovia system. The page has a blue header with 'Ministerio de Salud Pública' and a sidebar menu on the left with 'Monedas' selected. The main content area features a 'Monedas' section with a 'NUEVO' button (blue), a printer icon (red), and a document icon (green). Below this is a table with the following columns: 'Nombre', 'Usuario Crea', 'Fecha crea', 'Usuario Modifica', 'Fecha Modifica', 'Estado', and 'Opciones'. A 'RESETEAR' button (blue) is positioned below the table. At the bottom right, there is a 'Rows per page: 10' dropdown and navigation arrows. The footer of the page reads 'Desarrollado Por Universidad Nacional de Chimborazo - 2021'.

Fuente: Elaboración propia

Dar clic en “Nuevo” e ingresar el nombre de la moneda y hacer clic en.



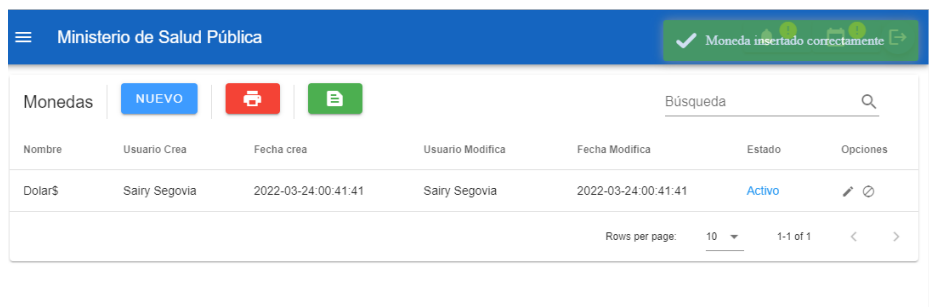
Figura 47. Registro de monedas



Fuente: Elaboración propia

Al ingresar el nombre de la moneda correctamente se muestra la alerta “Moneda ingresada correctamente”, se visualiza en la Figura 47.

Figura 48. Notificación de registro exitoso



Fuente: Elaboración propia

IV. Registro de Productos

Seleccionar la opción “Bodega” y dar clic en el botón.

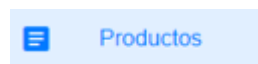
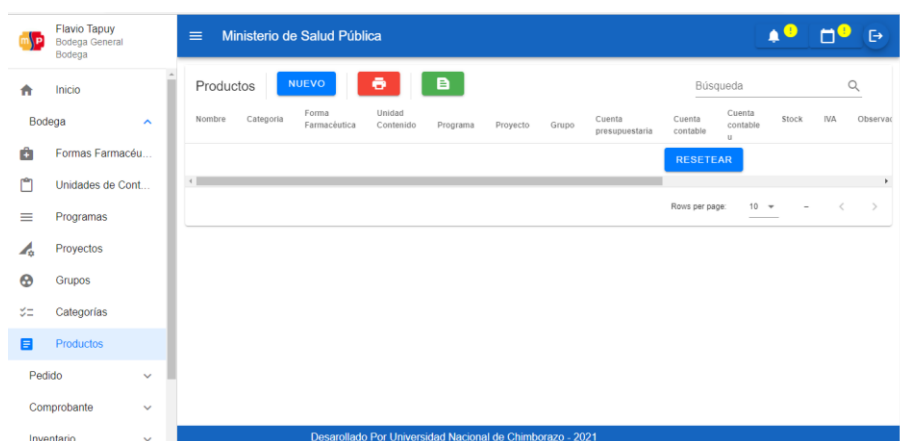


Figura 49. Página de Productos



Fuente: Elaboración propia

A continuación, dar clic en.



Figura 50. Botón “NUEVO”



Fuente: Elaboración propia

Se presenta el formulario de ingreso nuevo producto.

Figura 51. Formulario registro de Productos




Fuente: Elaboración propia

Ingresar el nombre del producto y escoger una categoría.

Figura 52. Registro de Nombre y Categoría

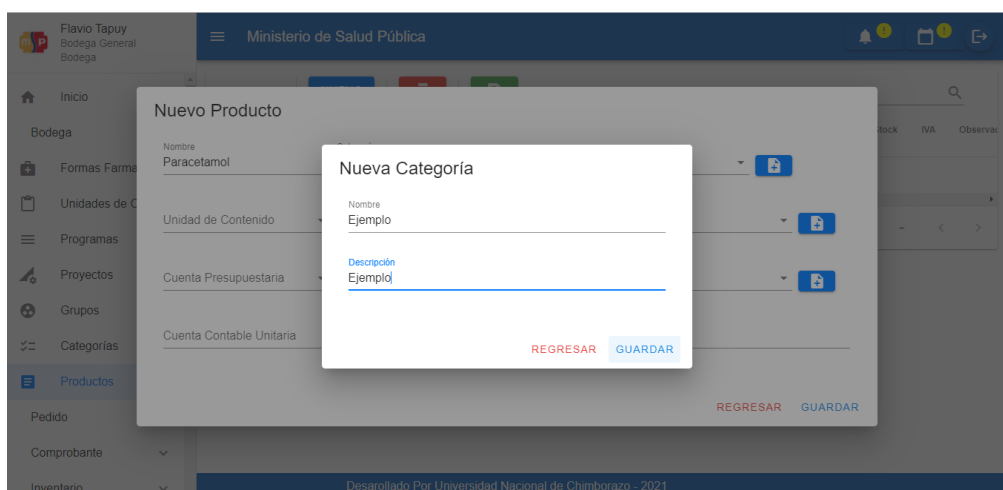


Fuente: Elaboración propia

En el caso de no tener una “Categoría”, dar clic en el botón  para agregar una nueva categoría.

Ingresar nombre y descripción y dar clic en  y se podrá seleccionar la nueva categoría.

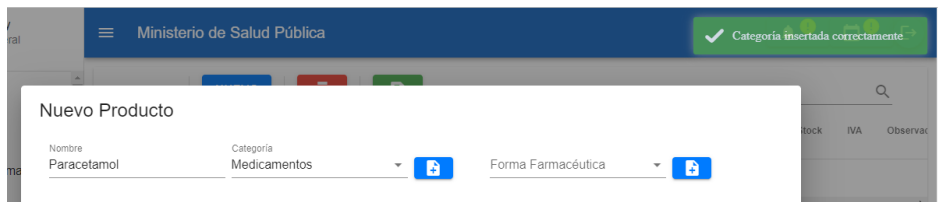
Figura 53. Registro de Nueva categoría



Fuente: Elaboración propia

Al ingresar una categoría correctamente se muestra la alerta “Categoría ingresada correctamente”, se visualiza en la Figura 53.

Figura 54. Mensaje de registro correcto



Fuente: Elaboración propia



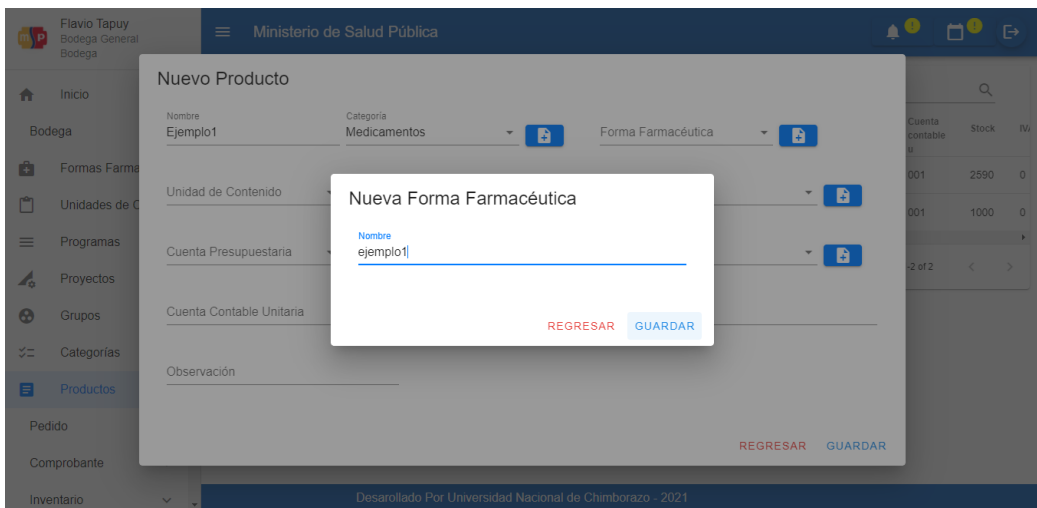
Seleccionar una “Forma Farmacéutica”, en el caso de no contar con una, agregar haciendo clic en el botón  e ingresar el nombre y dar clic en el botón .

Figura 55. Registro de Nueva Forma Farmacéutica



Fuente: Elaboración propia



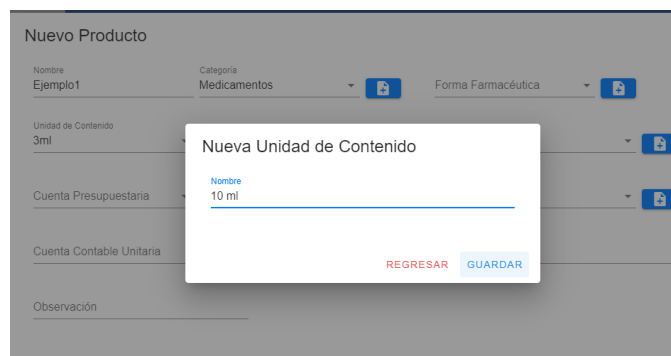
Seleccionar una “Unidad de Contenido”, en el caso de no contar con una, dar clic en el botón  e ingresar el nombre y dar clic en el botón .

Figura 56. Registro de nueva unidad



Fuente: Elaboración propia



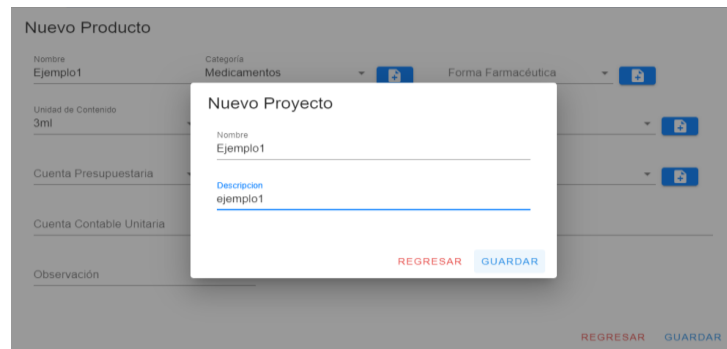
Seleccionar un “Proyecto”, en el caso de no tener registrado un proyecto, dar clic en el botón  e ingresar nombre y descripción, dar clic en el  botón.

Figura 57. Registro de Nuevo proyecto



Fuente: Elaboración propia



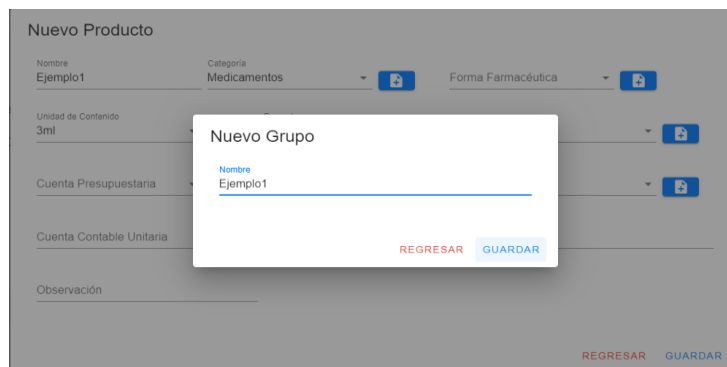
Seleccionar un “Grupo”, en el caso de no tener registrado un grupo, dar clic en el botón  e ingresar el nombre y dar clic en  el botón.

Figura 58. Registro de Nuevo grupo



Fuente: Elaboración propia



Seleccionar una “Cuenta Presupuestaria”, en el caso de no contar con una, dar clic en el botón  e ingresar los datos correspondientes, y hacer clic en el botón .

Figura 59. Registro de Nueva Cuenta Presupuestaria



Fuente: Elaboración propia




Seleccionar una “Cuenta Contable”, en el caso de no contar con una, dar clic en el botón  e ingresar los datos correspondientes y dar clic el botón. 

Figura 60. Registro de Nueva Cuenta Contable



La imagen muestra una interfaz de usuario para el registro de un nuevo producto. En primer plano, se encuentra un modal de diálogo titulado "Nueva Cuenta Contable". Este modal contiene los siguientes campos de entrada: "Codigo" con el valor "131 145", "Descripción" con el valor "ejemplo1", y "Tipo de Cuenta" con el valor "0001". Al final del modal, hay dos botones: "REGRESAR" en rojo y "GUARDAR" en azul. En el fondo, se puede ver parcialmente el formulario principal "Nuevo Producto" con campos como "Nombre" (Ejemplo1), "Unidad de Contenido" (3ml), "Cuenta Presupuestaria" (444) y "Cuenta Contable Unitaria".

Fuente: Elaboración propia



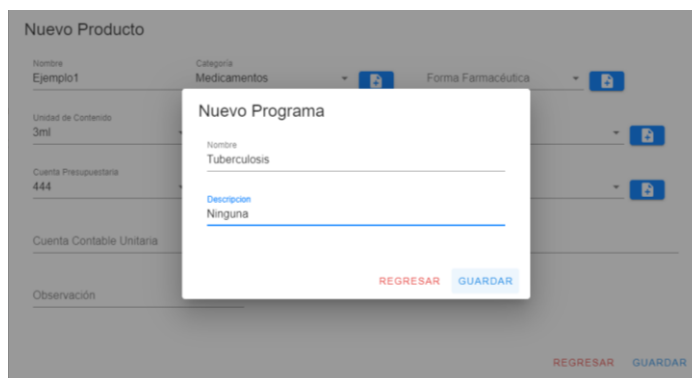
Seleccionar un “Programa”, en el caso de no tener registrado un programa, dar clic en el botón  e ingresar los datos correspondientes, y dar  clic en el botón.

Figura 61. Registro de Nuevo Programa



La imagen muestra la misma interfaz de usuario de registro de producto, pero con un modal de diálogo titulado "Nuevo Programa". Este modal contiene los campos "Nombre" con el valor "Tuberculosis" y "Descripción" con el valor "Ninguna". Al final del modal, hay dos botones: "REGRESAR" en rojo y "GUARDAR" en azul. El formulario de fondo "Nuevo Producto" muestra ahora los campos "Categoría" (Medicamentos) y "Forma Farmacéutica" (Forma Farmacéutica) seleccionados.

Fuente: Elaboración propia


La cuenta contable unitaria se generará automáticamente, por último, poner una descripción del producto, una vez completado todos los campos, dar clic en como se muestra en la Figura 61. 

Figura 62. Registro completo de datos del Producto

Nuevo Producto

Nombre: Paracetamol Categoría: Medicamentos Forma Farmacéutica: Sólido

Unidad de Contenido: 500mg Proyecto: Ejemplo1 Grupo: Ejemplo1

Cuenta Presupuestaria: 001 Cuenta Contable: 131.145 Programa: VIH

Cuenta Contable Unitaria: 131 VIH 001 IVA: 0 Observación: Ninguna

REGRESAR GUARDAR

Fuente: Elaboración propia

Al ingresar un producto correctamente se muestra la alerta “Producto ingresado correctamente”, se visualiza en la Figura 62.

Figura 63. Mensaje de registro correcto

Producto insertado correctamente

| Nombre | Categoría | Forma Farmacéutica | Unidad Contenido | Programa | Proyecto | Grupo | Cuenta presupuestaria | Cuenta contable | Cuenta contable u | Stock |
|-------------|--------------|--------------------|------------------|--------------|----------|----------|-----------------------|-----------------|-------------------|-------|
| paracetamol | Medicamentos | prueba1 | 3ml | VIH | prueba1 | ejemplo1 | 444 | 131.153 | 001 | 2590 |
| ibuprofeno | Medicamentos | prueba1 | 3ml | VIH | prueba1 | ejemplo1 | 444 | 131.153 | 001 | 1000 |
| Ejemplo1 | Medicamentos | ejemplo1 | 3ml | Tuberculosis | Ejemplo1 | ejemplo1 | 444 | 131.145 | 002 | 0 |

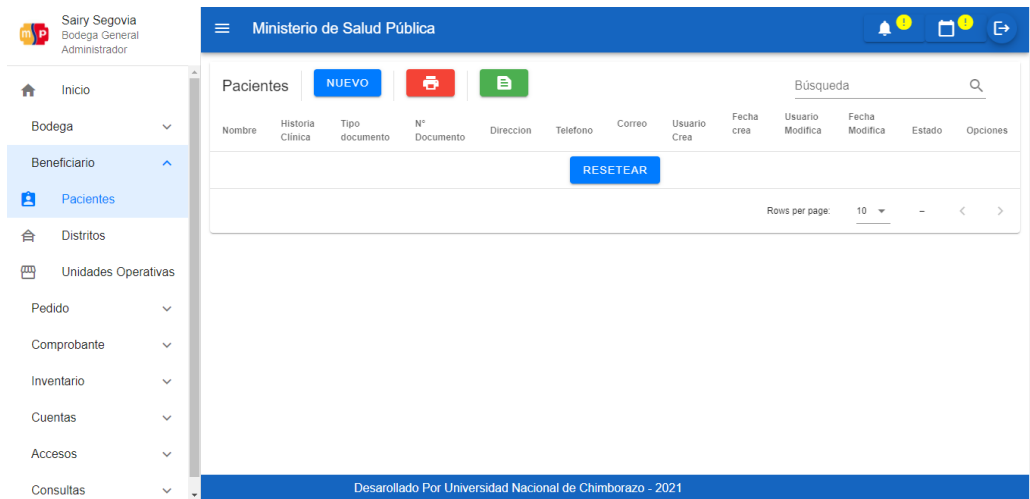
Rows per page: 10 1-3 of 3

Fuente: Elaboración propia

V. Registro de Beneficiarios

Para el registro de beneficiarios existe tres opciones: Pacientes, Distritos y Unidades Operativas.

Figura 64. Página de Beneficiarios

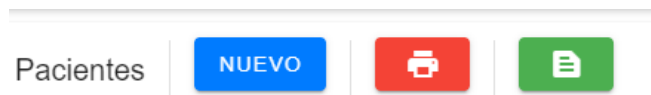


Fuente: Elaboración propia

Para registro de pacientes, hacer clic en “Pacientes” y seleccionar.



Figura 65. Botón “NUEVO”

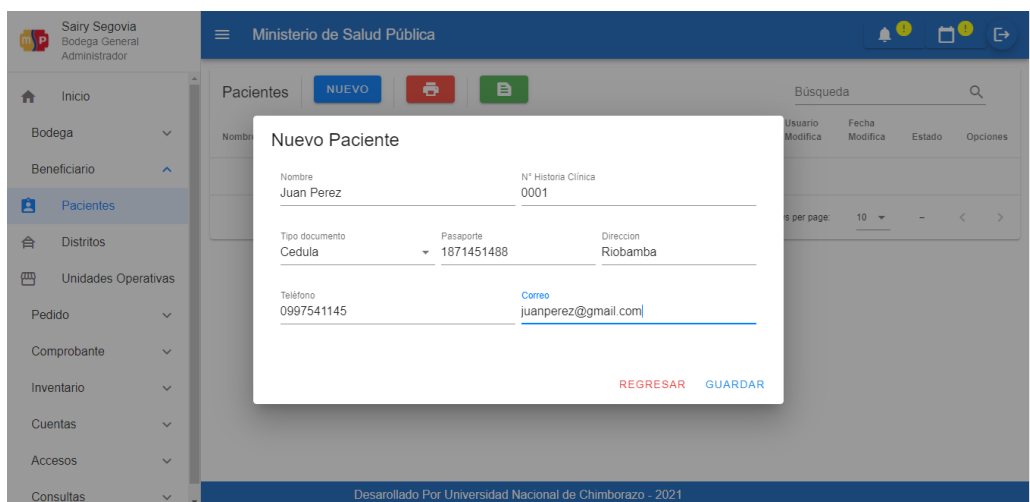


Fuente: Elaboración propia

Ingresar los datos y dar en el botón.



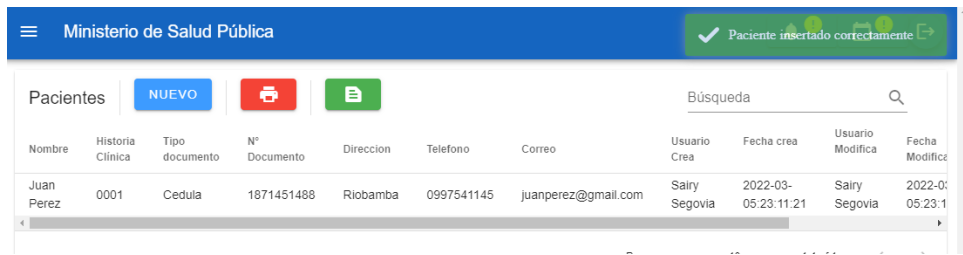
Figura 66. Registro completo de Nuevo Paciente



Fuente: Elaboración propia

Al ingresar un paciente correctamente se muestra la alerta “Paciente ingresado correctamente”, se visualiza en la Figura 66.

Figura 67. Mensaje de registro correcto



Fuente: Elaboración propia

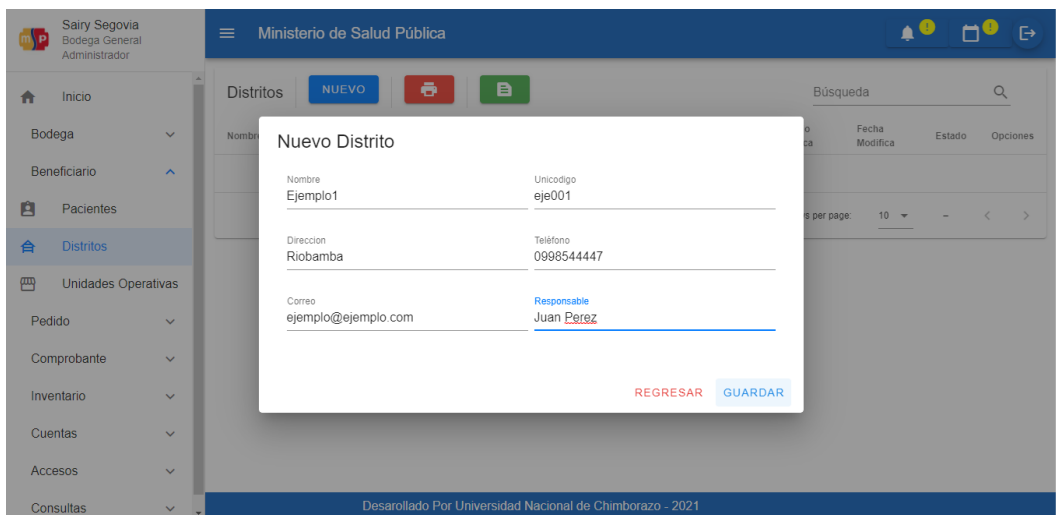
Para registrar distritos, dar clic en “Distritos” y dar clic en el botón.



Ingresar los datos y dar clic en el botón guardar.



Figura 68. Registro completo de nuevo Distrito



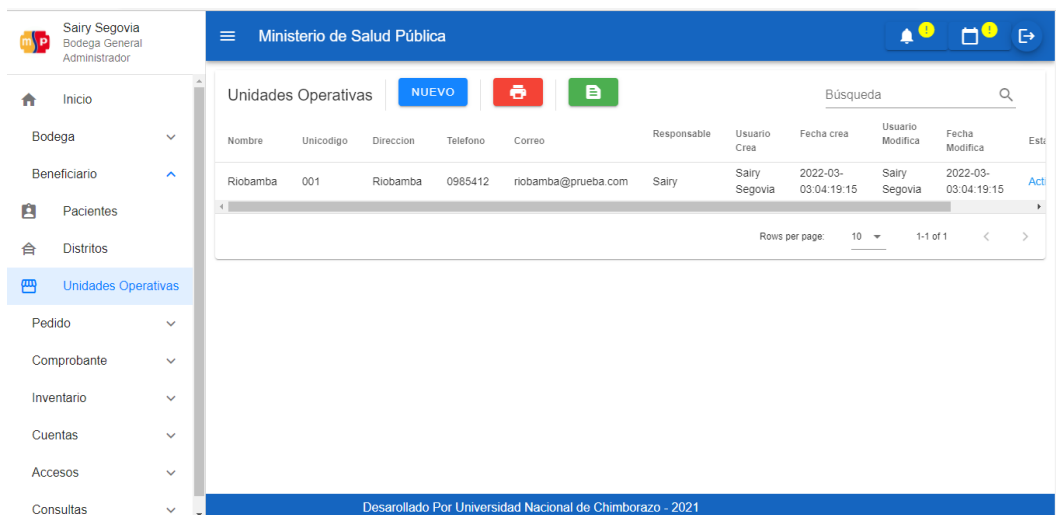
Fuente: Elaboración propia

Para registrar unidades operativas, seleccionar “Unidades Operativas” y dar clic en el botón.

Figura



69. Página de Unidades Operativas

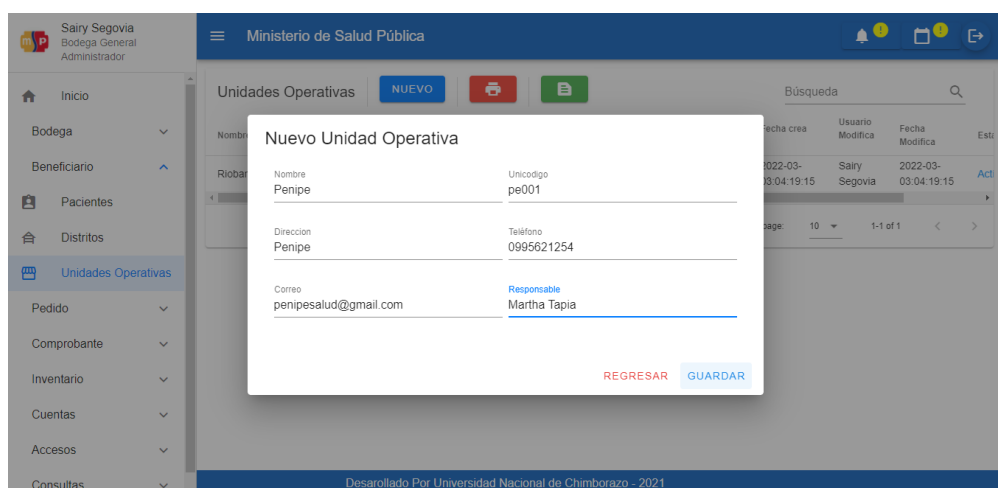


Fuente: Elaboración propia

Ingresar los datos y dar clic en el botón.



Figura 70. Formulario de registro de Nuevas Unidades Operativas

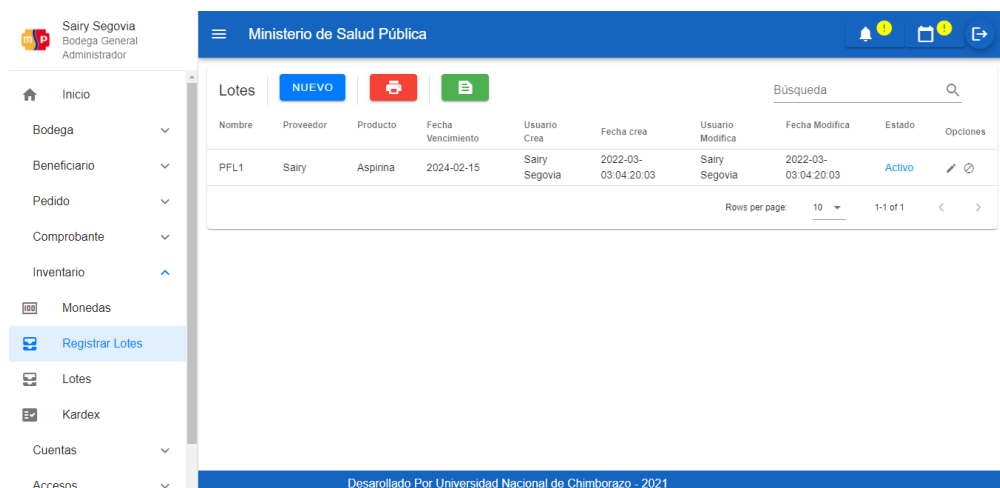


Fuente: Elaboración propia

VI. Registro de Lotes

Para el ingreso de lotes, dirigirse a “Inventario” en el menú, y seleccionar la opción “Registrar Lotes”.

Figura 71. Página de registro de lotes

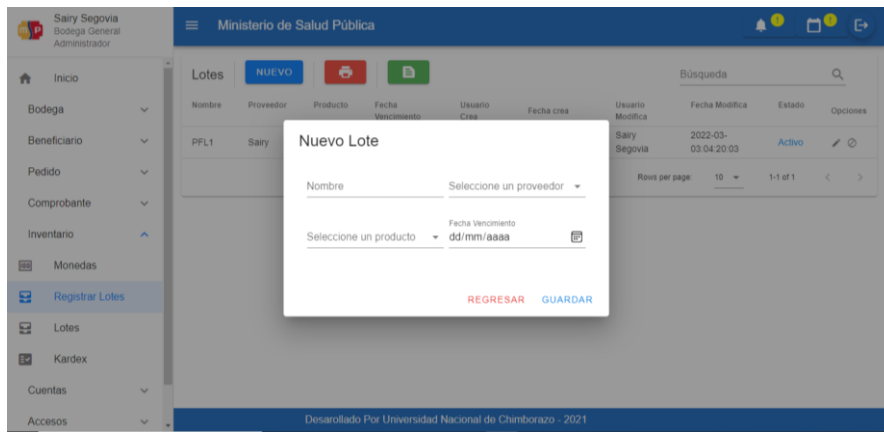


Fuente: Elaboración propia

Para registrar lotes, dar clic en el botón.



Figura 72. Formulario de registro de Lotes

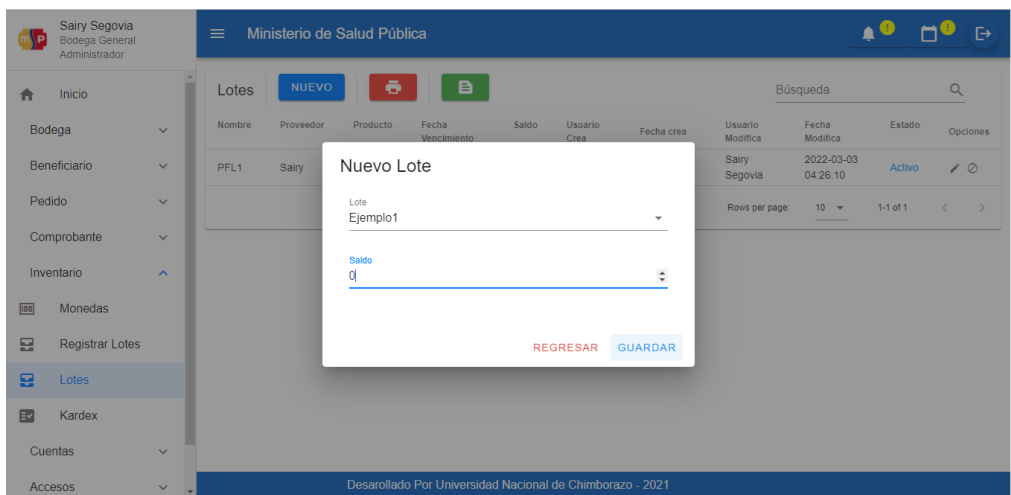


Fuente: Elaboración propia

Ingresar el nombre del lote, seleccionar un proveedor, un producto y la fecha de caducidad del producto, llenar los datos y dar clic en el botón. **GUARDAR**

Dirigirse a “Lotes”, dar clic en y **NUEVO** asignar el lote creado y clic en botón.

Figura 73. Formulario de asignación de lote



Fuente: Elaboración propia

VII. Registro de cuentas

Para el ingreso de cuentas, dirigirse a “Cuentas” en la parte de menú, y seleccionar “Tipos de cuentas”.

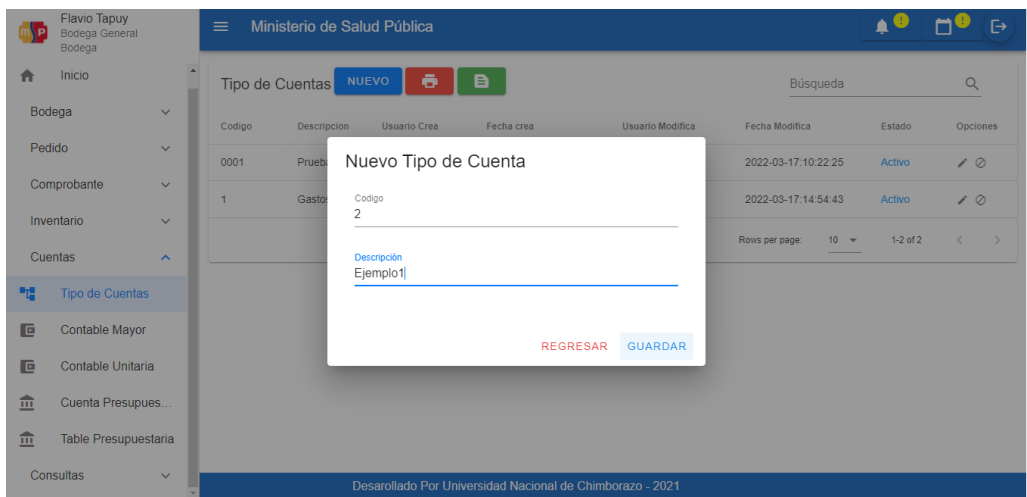
Figura 74. Página de registro de tipos de cuentas



Fuente: Elaboración propia

Para registrar cuentas, dar clic en **NUEVO** e ingresar los datos y dar clic en el botón.

Figura 75. Formulario de registro de tipo de cuenta



Fuente: Elaboración propia

VIII. Pasos para realizar ingresos, egresos, reingreso, transferencia temporal y transferencia completa

Para el ingreso, escoger la opción “Comprobante” en la parte de menú, y seleccionar “Comprobantes”.

Figura 76. Página de Comprobante

| Beneficiario | Concepto | Usuario | Moneda | Orden | Numero Comprobante | Valor neto | IVA | Total | Fecha | Usuario Crea | Fecha Crea | Usuario Modifi |
|----------------|------------------------|--------------|---------|-------|--------------------|------------|-----|-------|---------------------|--------------|---------------------|----------------|
| Penipe | Transferencia Completa | Flavio Tapuy | dolar\$ | 2 | 2 | 180 | 0 | 180 | 2022-03-17T15:26:50 | Flavio Tapuy | 2022-03-17:15:26:50 | Flavio Tapuy |
| Bodega General | Ingreso | Flavio Tapuy | dolar\$ | 1 | 1 | 900 | 12 | 1008 | 2022-03-17T15:23:40 | Flavio Tapuy | 2022-03-17:15:23:40 | Flavio Tapuy |

Fuente: Elaboración propia

Para registrar, dar clic en **NUEVO** y deberá aparecer la Figura 76, en esta parte también se puede ingresar lotes y asignarlos, la recomendación es realizar el proceso de lotes.

Figura 77. Formulario de Registro de Comprobantes

| Producto | Lote | Cantidad | Precio | Borrar |
|--|------|----------|--------|--------|
| No hay productos agregados al detalle. | | | | |

Fuente: Elaboración propia

Seleccionar el concepto “Ingreso”.

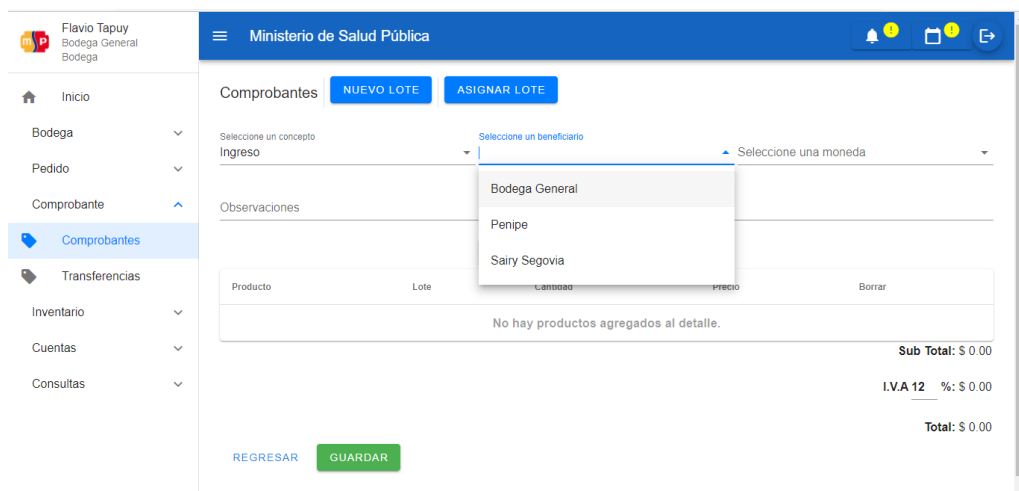
Figura 78. Realización de ingresos



Fuente: Elaboración propia

Seleccionar un beneficiario, en este caso, Bodega General.

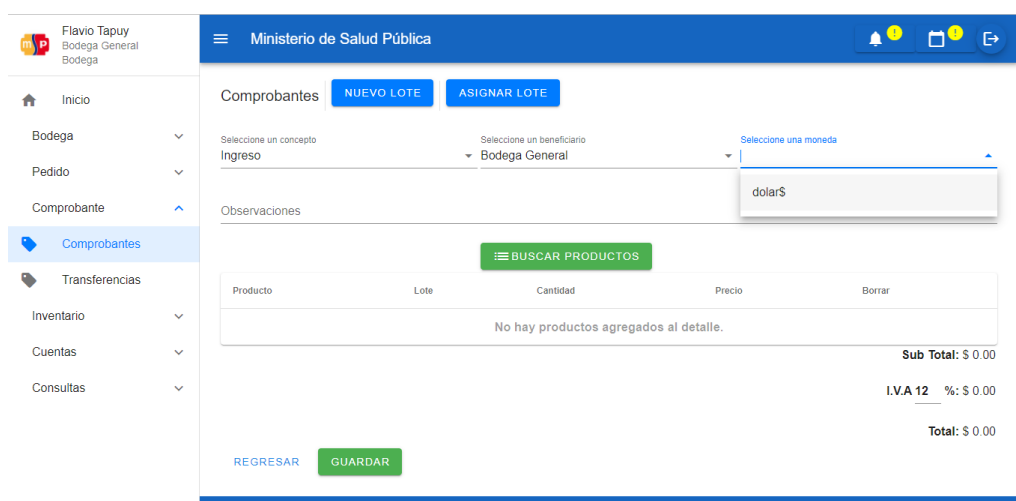
Figura 79. Selección de beneficiario



Fuente: Elaboración propia

Seleccionar una moneda.

Figura 80. Selección de moneda

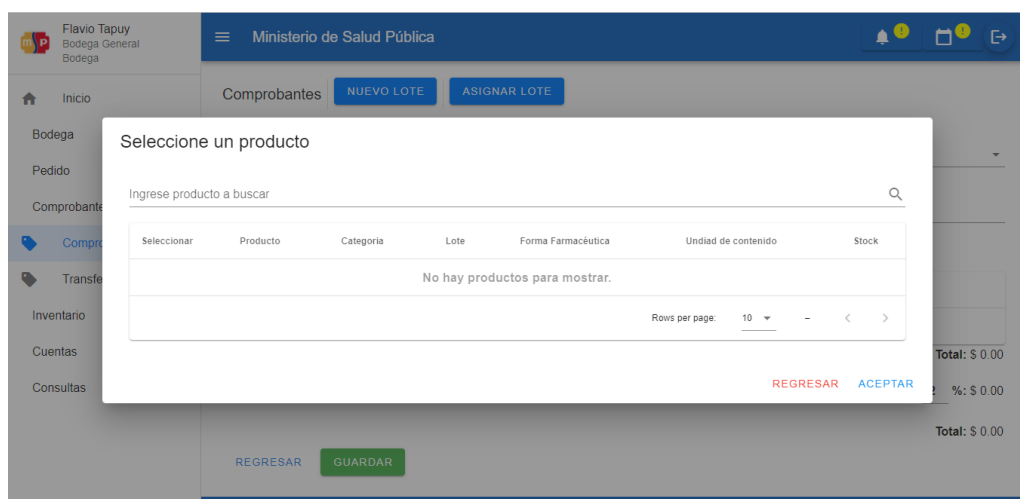


Fuente: Elaboración propia

Buscar el o los productos, para ello dar clic en el botón.

BUSCAR PRODUCTOS

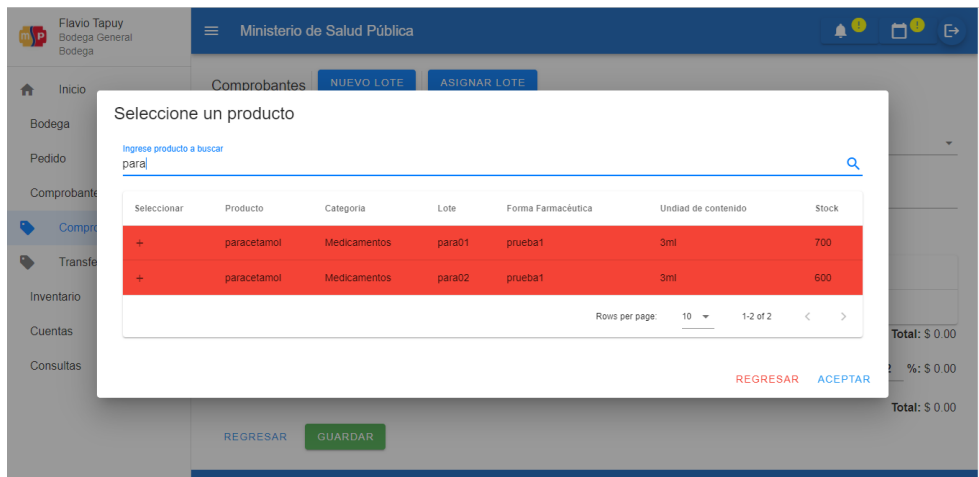
Figura 81. Ventana de selección de los productos



Fuente: Elaboración propia

Buscar productos colocando las iniciales y dar Enter, deberá aparecer el producto, de lo contrario, el producto no se encuentra registrado o asignado a un lote.

Figura 82. Búsqueda de Productos

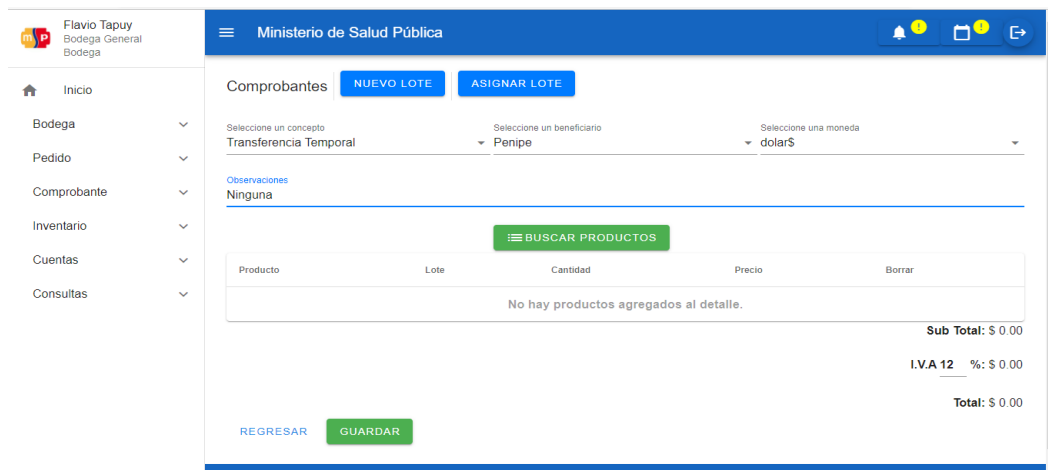


Fuente: Elaboración propia

Dar clic en el signo “+” para seleccionar el producto, una vez seleccionado los productos, dar clic en el botón **ACEPTAR**.

Ingresar la cantidad del producto y el precio. Además, ingresar observaciones y dar clic en el botón **GUARDAR**.

Figura 83. Módulo de registro de Comprobante



Fuente: Elaboración propia

IX. Realizar una transferencia

Para todos, es el mismo proceso lo único que cambia es el concepto.

Seleccionar “Transferencia temporal”.

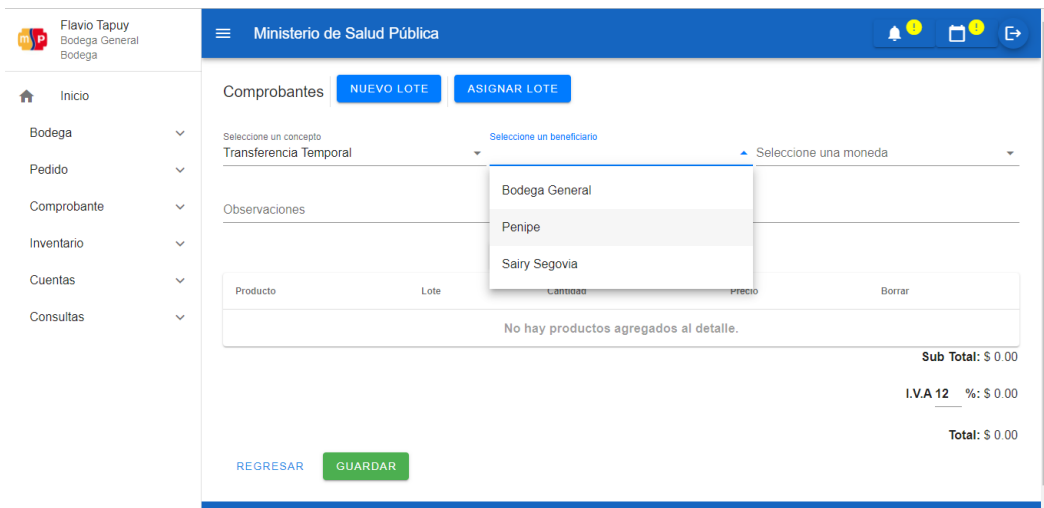
Figura 84. Realización de transferencias



Fuente: Elaboración propia

Seleccionar un beneficiario.

Figura 85. Selección de unidad operativa en beneficiario.



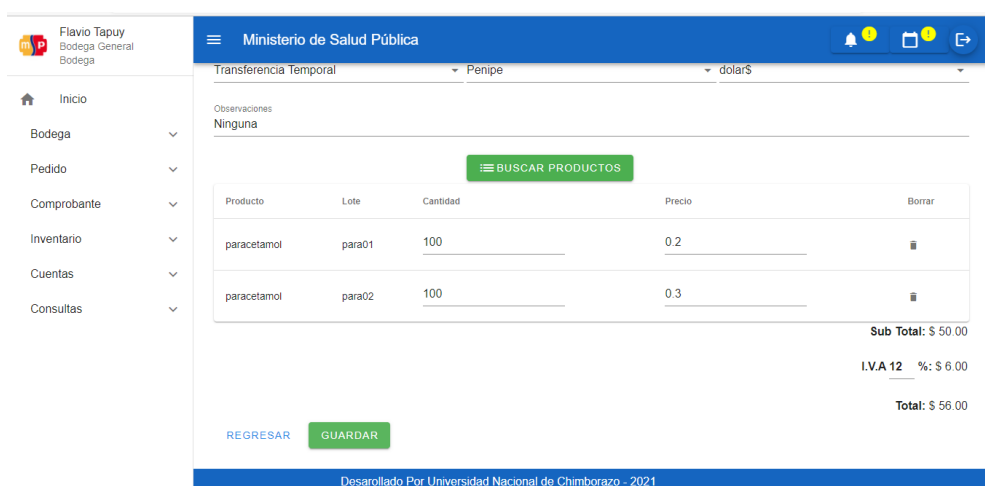
Fuente: Elaboración propia

Seleccionar una moneda.

Seleccionar productos y dar clic en **ACEPTAR**.

Ingresar la cantidad, precio del producto y dar en clic en el botón. **GUARDAR**

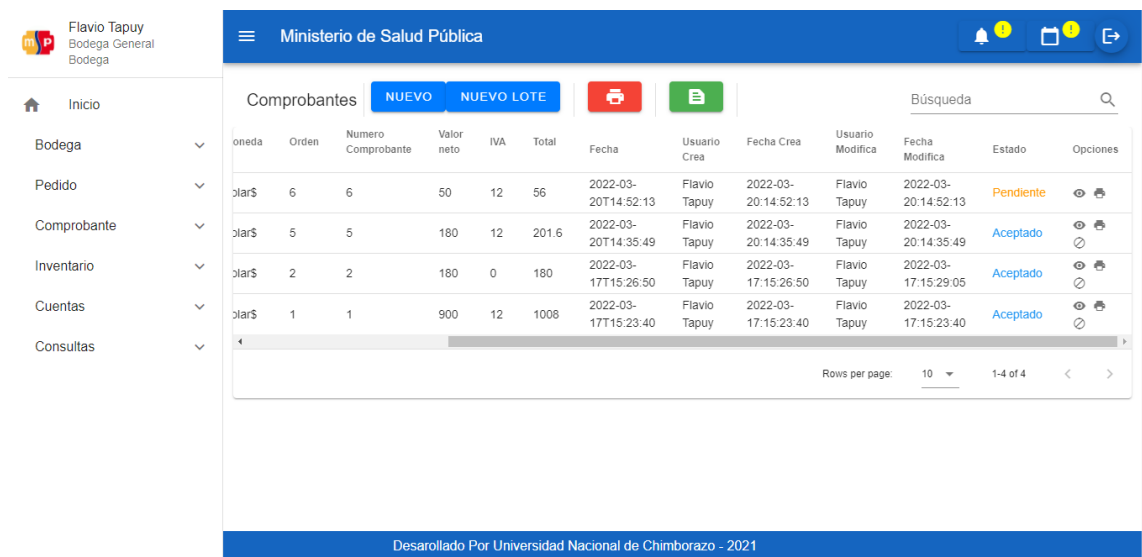
Figura 86. Módulo de registro de Transferencia Temporal



Fuente: Elaboración propia

Como se puede observar, la transferencia aparece con estado “Pendiente”, esperando la aprobación del beneficiario para que pueda aparecer el estado como “transacción completa”.

Figura 87. Módulo de Comprobantes



Fuente: Elaboración propia


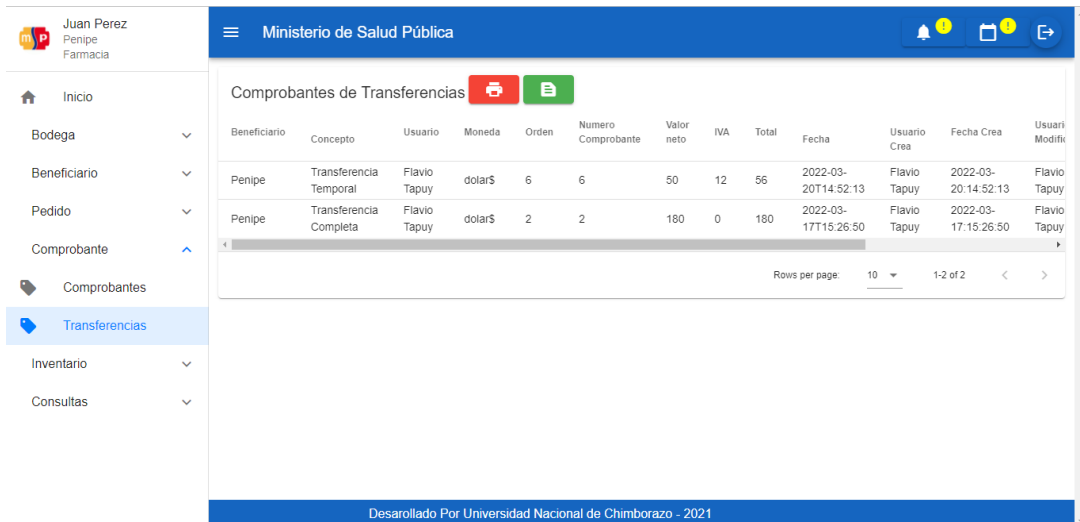
Para aprobar la transferencia, el beneficiario deberá dirigirse a “Comprobante” en el menú principal, y seleccionar 

Figura 88. Página de Transferencias



Fuente: Elaboración propia

Dar clic en el signo de vista previa para verificar si la información es correcta.

Figura 89. Opciones de Transferencias



Fuente: Elaboración propia

Para verificar la información dar clic en el botón.



Figura 90. Vista previa al realizar la transacción

The screenshot shows a web application interface for 'Ministerio de Salud Pública'. The user is logged in as 'Juan Perez Penipe Farmacia'. The page title is 'Ministerio de Salud Pública'. The main content area shows a transfer form for 'Penipe' with the following details:

- Orden: 6
- Número Comprobante: 6
- Observaciones: Ninguna
- Productos:

| Producto | Lote | Cantidad | Precio | Borrar |
|-------------|--------|----------|--------|--------|
| paracetamol | para01 | 100 | 0,2 | |
| paracetamol | para02 | 100 | 0,3 | |
- Sub Total: \$ 50.00
- I.V.A 12 %: \$ 6.00
- Total: \$ 56.00

At the bottom, there are two buttons: 'REGRESAR' and 'REALIZAR TRANSFERENCIA'.

Fuente: Elaboración propia

Dar clic el signo de visto para que pueda ser aceptada la transferencia.

Figura 91. Estado pendiente de la transferencia

The screenshot shows a table titled 'Comprobantes de Transferencias' with the following columns: Moneda, Orden, Numero Comprobante, Valor neto, IVA, Total, Fecha, Usuario Crea, Fecha Crea, Usuario Modifica, Fecha Modifica, Estado, and Opciones. The table contains two rows of data:

| Moneda | Orden | Numero Comprobante | Valor neto | IVA | Total | Fecha | Usuario Crea | Fecha Crea | Usuario Modifica | Fecha Modifica | Estado | Opciones |
|---------|-------|--------------------|------------|-----|-------|---------------------|--------------|---------------------|------------------|---------------------|-----------|----------|
| Jolar\$ | 6 | 6 | 50 | 12 | 56 | 2022-03-20T14:52:13 | Flavio Tapuy | 2022-03-20:14:52:13 | Flavio Tapuy | 2022-03-20:14:52:13 | Pendiente | 👁️ 🗑️ |
| Jolar\$ | 2 | 2 | 180 | 0 | 180 | 2022-03-17T15:26:50 | Flavio Tapuy | 2022-03-17:15:26:50 | Flavio Tapuy | 2022-03-17:15:29:05 | Aceptado | 👁️ 🗑️ |

At the bottom of the table, there is a 'Rows per page' dropdown set to 10 and a '1-2 of 2' indicator.

Fuente: Elaboración propia

Dar clic en **ACEPTAR**, el estado cambio a “Aceptado” y aparece la notificación que confirma la misma.

Figura 92. Ventana de aceptación de la transferencia



Fuente: Elaboración propia

X. Realizar un egreso a un paciente

Primero registrar un paciente. Realizar los siguientes pasos:

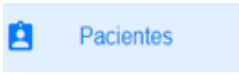
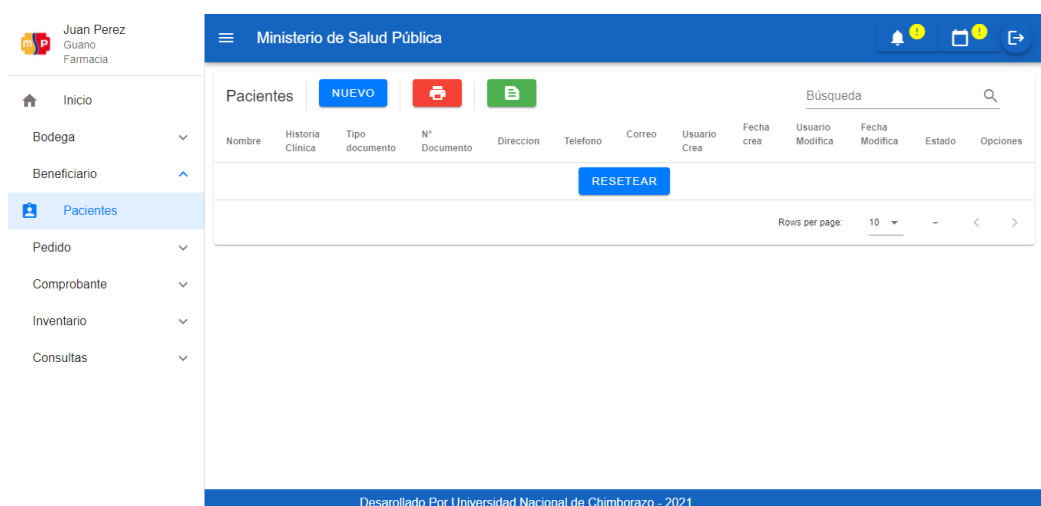
- Ingresar al sistema con el rol de unidad operativa, en el menú de opciones dirigirse a “Beneficiario” y dar clic en .

Figura 93. Página de registro de Pacientes



Fuente: Elaboración propia


A continuación, dar clic en el botón .

Figura 94. Formulario de registro de Paciente

Fuente: Elaboración propia

Siendo el único campo obligatorio “Nombres y Apellidos”.

Dirigirse a “Comprobantes para hacer un egreso”.

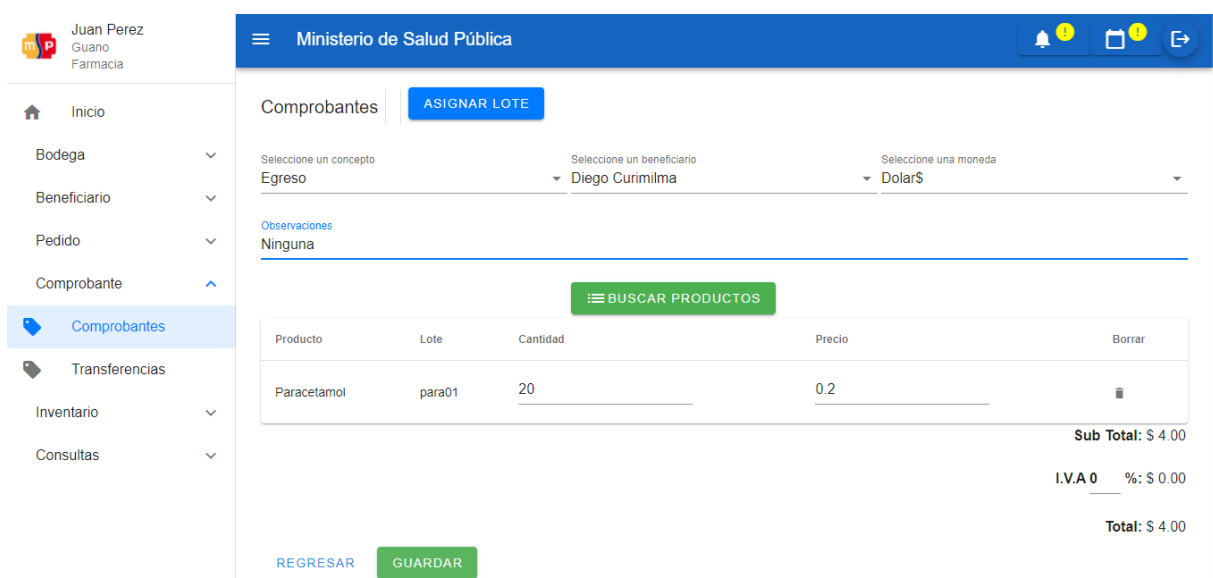
Dar clic en **NUEVO**. En “Conceptos” seleccionar “Egreso”. Seleccionar el nombre del beneficiario. Seleccionar una moneda.

Figura 95. Formulario de egreso

Fuente: Elaboración propia

Buscar los productos y poner la cantidad y precio. Y Por último llenar las observaciones, y dar clic en **GUARDAR**

Figura 96. Selección de productos para egresos



Fuente: Elaboración propia

XI. Ver el saldo de los productos desde el rol de “Médico”

Ingresar con el rol médico: en el menú, seleccionar Inventario y elegir

Lotes

Figura 97. Módulo de rol Médico



Fuente: Elaboración propia

Se presenta la pantalla (Figura 97) y se visualiza el stock del producto.

Figura 98. Página de lotes - rol médico

| Nombre | Proveedor | Producto | Fecha Vencimiento | Saldo | Usuario Crea | Fecha crea | Usuario Modifica | Fecha Modifica | Estado | Opciones |
|--------|-----------|-------------|-------------------|-------|--------------|---------------------|------------------|---------------------|--------|----------|
| para01 | Deltex | Paracetamol | 2022-05-26 | 80 | Flavio Tapuy | 2022-03-24 00:54:11 | Flavio Tapuy | 2022-03-24 00:54:11 | Activo | |

Fuente: Elaboración propia

XII. Migrar Datos

Importar los datos de un archivo Excel a una base de datos MySQL, la base de datos se administra mediante phpMyAdmin.

Ver la estructura de la tabla a la cual se va a importar los datos, en este caso la tabla “unidad”.

Figura 99. Tabla “Unidad”

```
SELECT * FROM `unidad`
```

| idUnidad | nombreUnidad | usuarioCrea | fechaCrea | usuarioModifica | fechaModifica | estado |
|----------|--------------|-------------|-----------|-----------------|---------------|--------|
|----------|--------------|-------------|-----------|-----------------|---------------|--------|

Fuente: Elaboración propia

Unir todos los datos con la misma estructura que tiene la base de datos.

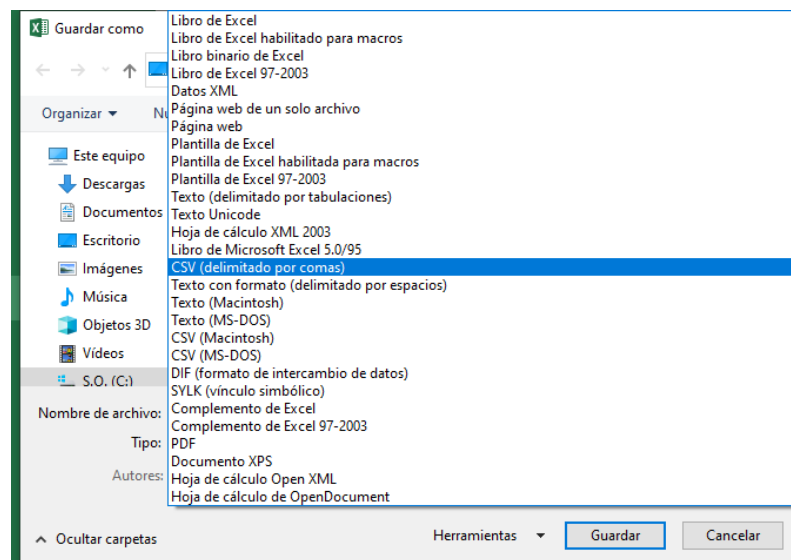
Figura 100. Excel de los datos a importar

| | A | B | C | D | E | F | G |
|---|---|------------|---------------|-------|---------------|-------|---|
| 1 | 1 | VIH | Sairy segovia | ##### | Sairy segovia | | 1 |
| 2 | 2 | GENERICOS | Sairy segovia | ##### | Sairy segovia | ##### | 1 |
| 3 | 3 | MICRONUTR | Sairy segovia | ##### | Sairy segovia | ##### | 1 |
| 4 | 4 | EMERGENCIA | Sairy segovia | ##### | Sairy segovia | ##### | 1 |

Fuente: Elaboración propia

Guardar el archivo en formato CSV.

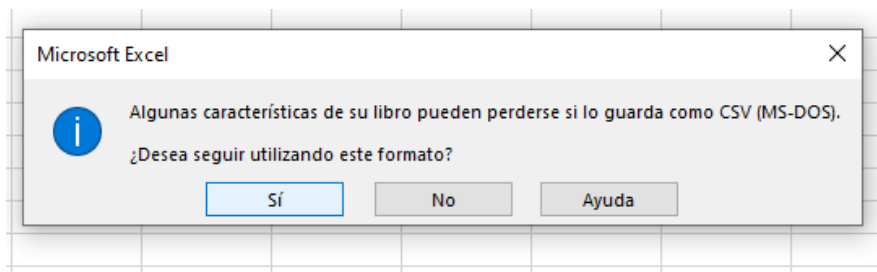
Figura 101. Selección de formato de archivo Excel



Fuente: Elaboración propia

Aparecerá la siguiente ventana, y dar clic en “Sí”.

Figura 102. Mensaje para confirmar el respaldo del archivo



Fuente: Elaboración propia

Dirigirse al nuevo archivo de Excel, y verificar que los datos se guardaron correctamente.

Figura 103. Excel de datos previos a ser importados

| | A | B | C | D | E | F | G |
|---|---|------------|---------------|-------|---------------|-------|---|
| 1 | 1 | VIH | Sairy segovia | ##### | Sairy segovia | | 1 |
| 2 | 2 | GENERICOS | Sairy segovia | ##### | Sairy segovia | ##### | 1 |
| 3 | 3 | MICRONUTR | Sairy segovia | ##### | Sairy segovia | ##### | 1 |
| 4 | 4 | EMERGENCIA | Sairy segovia | ##### | Sairy segovia | ##### | 1 |

Fuente: Elaboración propia

Dirigirse a la base de datos y seleccionar la tabla a la cual importar los datos, en este caso, la tabla “unidad”.


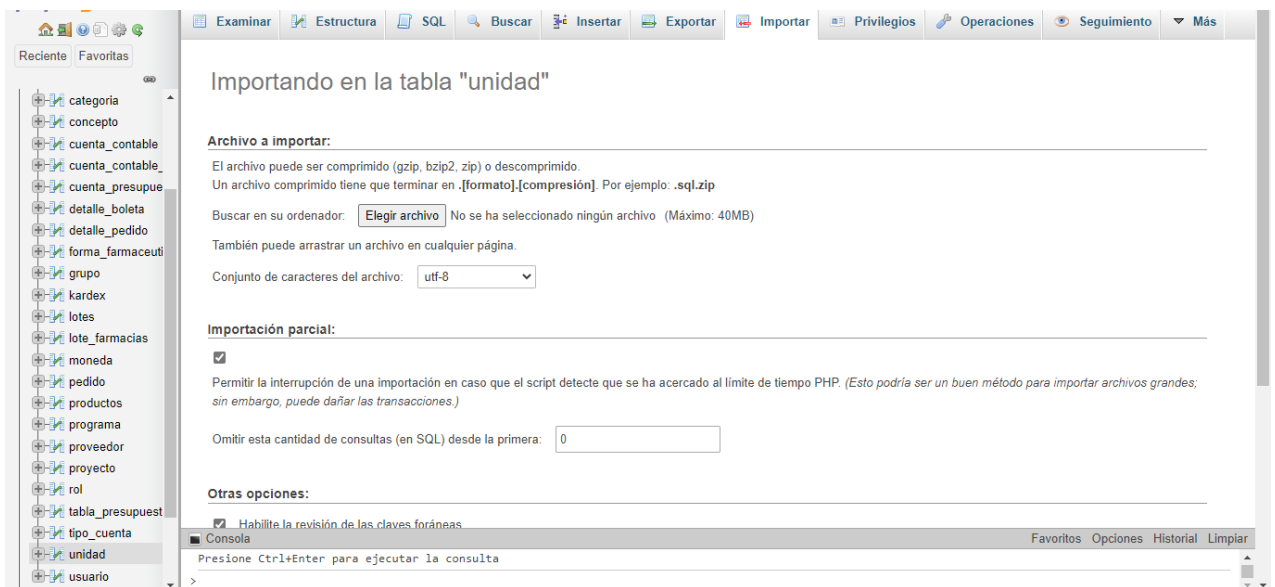
En la tabla “unidad”, dar clic en  .

Figura 104. Página para importar datos a la tabla “Unidad”



Importando en la tabla "unidad"

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.
Un archivo comprimido tiene que terminar en .[formato].[compresión]. Por ejemplo: .sql.zip

Buscar en su ordenador: No se ha seleccionado ningún archivo (Máximo: 40MB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

Importación parcial:

Permitir la interrupción de una importación en caso que el script detecte que se ha acercado al límite de tiempo PHP. (Esto podría ser un buen método para importar archivos grandes; sin embargo, puede dañar las transacciones.)

Omitir esta cantidad de consultas (en SQL) desde la primera:

Otras opciones:

Habilita la revisión de las claves foráneas.

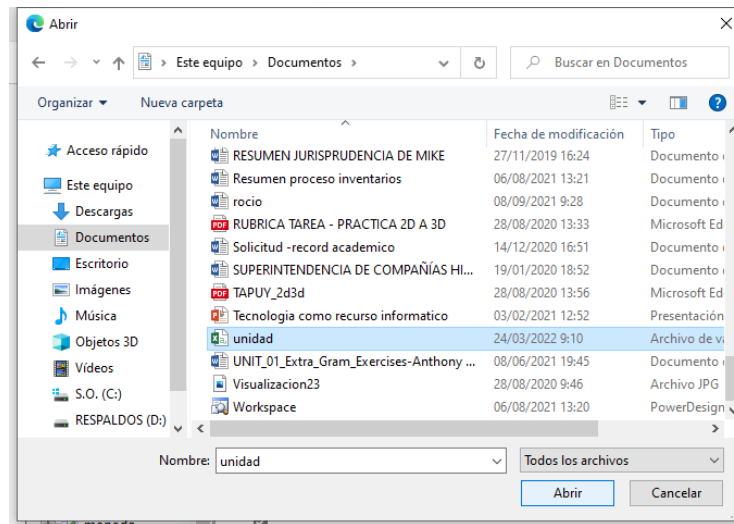
Consola Favoritos Opciones Historial Limpiar

Presione Ctrl+Enter para ejecutar la consulta

Fuente: Elaboración propia

Ahora, dar clic en y seleccionar el archivo Excel y hacer clic en “Abrir”.

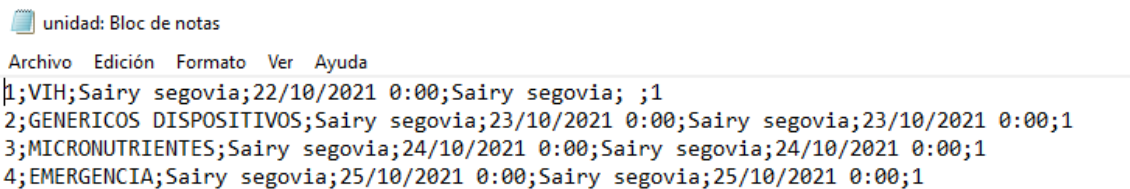
Figura 105. Selección del archivo a importar



Fuente: Elaboración propia

- Al abrir el archivo Excel con un blog de notas, se observa el formato que está utilizando para guardar los datos.

Figura 106. Blog de notas de los datos del archivo Excel



Fuente: Elaboración propia


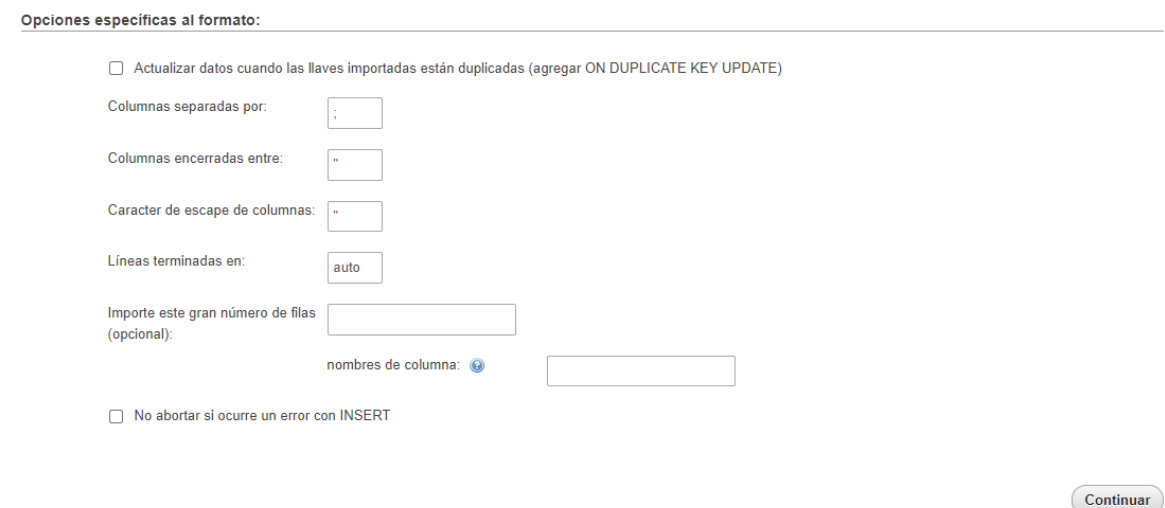
Ahora, en “Columnas separadas por”, ingresar “;” y dar clic en .

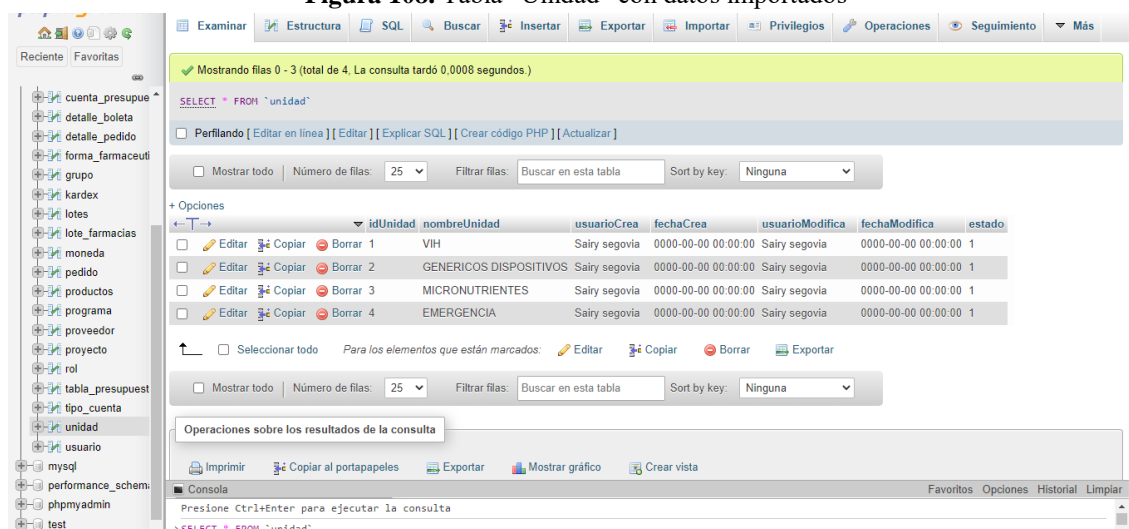
Figura 107. Opciones específicas al formato



Fuente: Elaboración propia

Se exportaron los datos correctamente a la base de datos.

Figura 108. Tabla “Unidad” con datos importados



The screenshot shows a MySQL database management tool interface. The main window displays the results of a SQL query: `SELECT * FROM 'unidad'`. The query returned 4 rows of data. The table has the following columns: `idUnidad`, `nombreUnidad`, `usuarioCrea`, `fechaCrea`, `usuarioModifica`, `fechaModifica`, and `estado`. The data rows are:

| idUnidad | nombreUnidad | usuarioCrea | fechaCrea | usuarioModifica | fechaModifica | estado |
|----------|------------------------|---------------|---------------------|-----------------|---------------------|--------|
| 1 | VIH | Sairy segovia | 0000-00-00 00:00:00 | Sairy segovia | 0000-00-00 00:00:00 | 1 |
| 2 | GENERICOS DISPOSITIVOS | Sairy segovia | 0000-00-00 00:00:00 | Sairy segovia | 0000-00-00 00:00:00 | 1 |
| 3 | MICRONUTRIENTES | Sairy segovia | 0000-00-00 00:00:00 | Sairy segovia | 0000-00-00 00:00:00 | 1 |
| 4 | EMERGENCIA | Sairy segovia | 0000-00-00 00:00:00 | Sairy segovia | 0000-00-00 00:00:00 | 1 |

The interface also shows a sidebar with a tree view of database tables, including `unidad`. The console at the bottom shows the executed query: `>SELECT * FROM 'unidad'`.

Fuente: Elaboración propia

Anexo 2: Manual técnico

MANUAL TÉCNICO

SOFTWARE DE CONTROL DE INVENTARIO PARA LA FARMACIA DE LA COORDINACIÓN DE SALUD ZONA 3

Versión 1



I. REQUISITOS DEL SISTEMA

1.1. Requerimientos del hardware

- Laptop, ordenador de escritorio o dispositivo móvil.
- Memoria RAM 2 GB en adelante.
- Procesador 1.4 GHz.
- Tarjeta de red LAN o WIFI.

1.2. Requerimientos del software

- Sistema operativo (Windows, Android, IOS o Linux).
- Navegadores (Chrome, Firefox, Opera, Edge, Safari, QQ, Baidu, UC Browser, Samsung internet, Mint Browser y Wechat).

II. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

2.1. Node Js

Es una plataforma de software de ejecución de JavaScript, para construir una red escalable aplicaciones sin esfuerzo, controlado por eventos que lo hace ligero y eficiente, perfecto para datos intensivos en tiempo real aplicaciones que se ejecutan en distribuidos dispositivos (Hota & Prabhu, 2016).

2.2. MySql

Es un sistema gestor de bases de datos, usado por su simplicidad y notable rendimiento, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha, ofreciendo un alto grado de estabilidad y un rápido desarrollo (Santillan, 2014).

2.3. Visual Studio 2017 Community

Es una interfaz de desarrollo de Microsoft, que se compone herramientas que permiten a los desarrolladores crear aplicaciones para plataformas .NET (Hugon, 2018).

2.4. Visual Studio Code

Es un editor de texto o código fuente independiente.

III. EJECUCION DEL PROYECTO

3.1. Ejecución del Front End

- Ingresar al proyecto vue desde cualquier terminal y colocar los siguientes comandos:
 - npm install
 - npm run serve

3.2. Ejecución del Back End

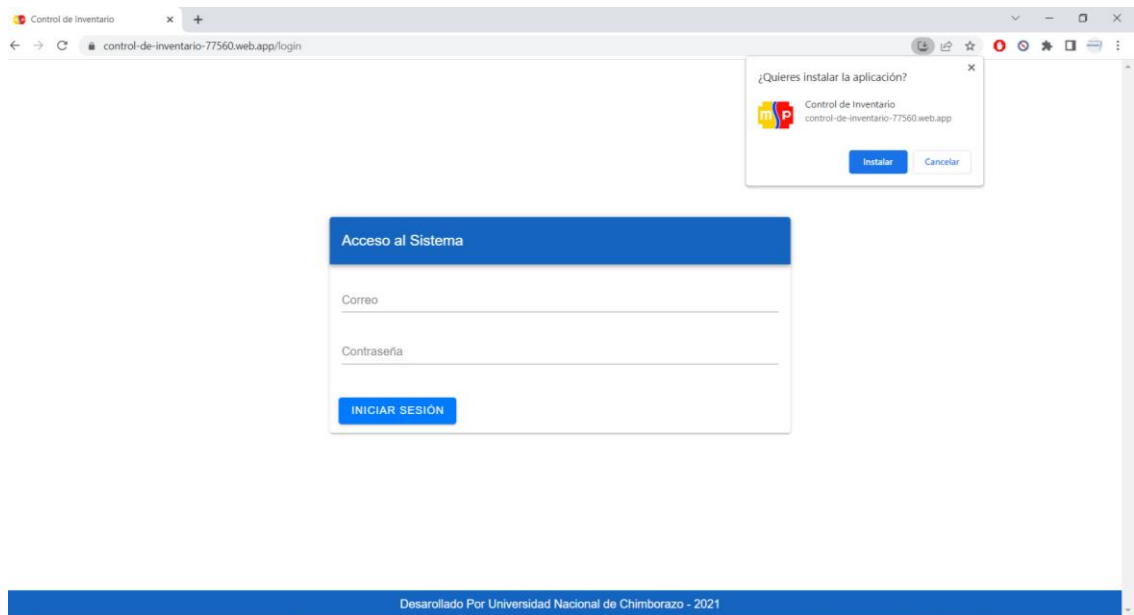
- Correr el servidor de base de datos MySQL.
- Ingresar al proyecto apilaravel desde cualquier terminal y colocar el siguiente comando:
 - php artisan serve
- Abrir el proyecto inventario desde Visual Studio y seleccionar Inventario.Web para ejecutar.

IV. Instalación del aplicativo web progresivo

4.1. Desde un ordenador

- Acceder al aplicativo web, en la barra de direcciones dar clic en el botón instalar, que se encuentra en la parte derecha de la barra de direcciones.

Figura 109. Instalación desde un navegador.



Fuente: Elaboración propia.

4.2. Desde un dispositivo móvil

- Acceder con la url al aplicativo e inmediatamente aparecerá la opción de instalar en la pantalla del principal e instalar.

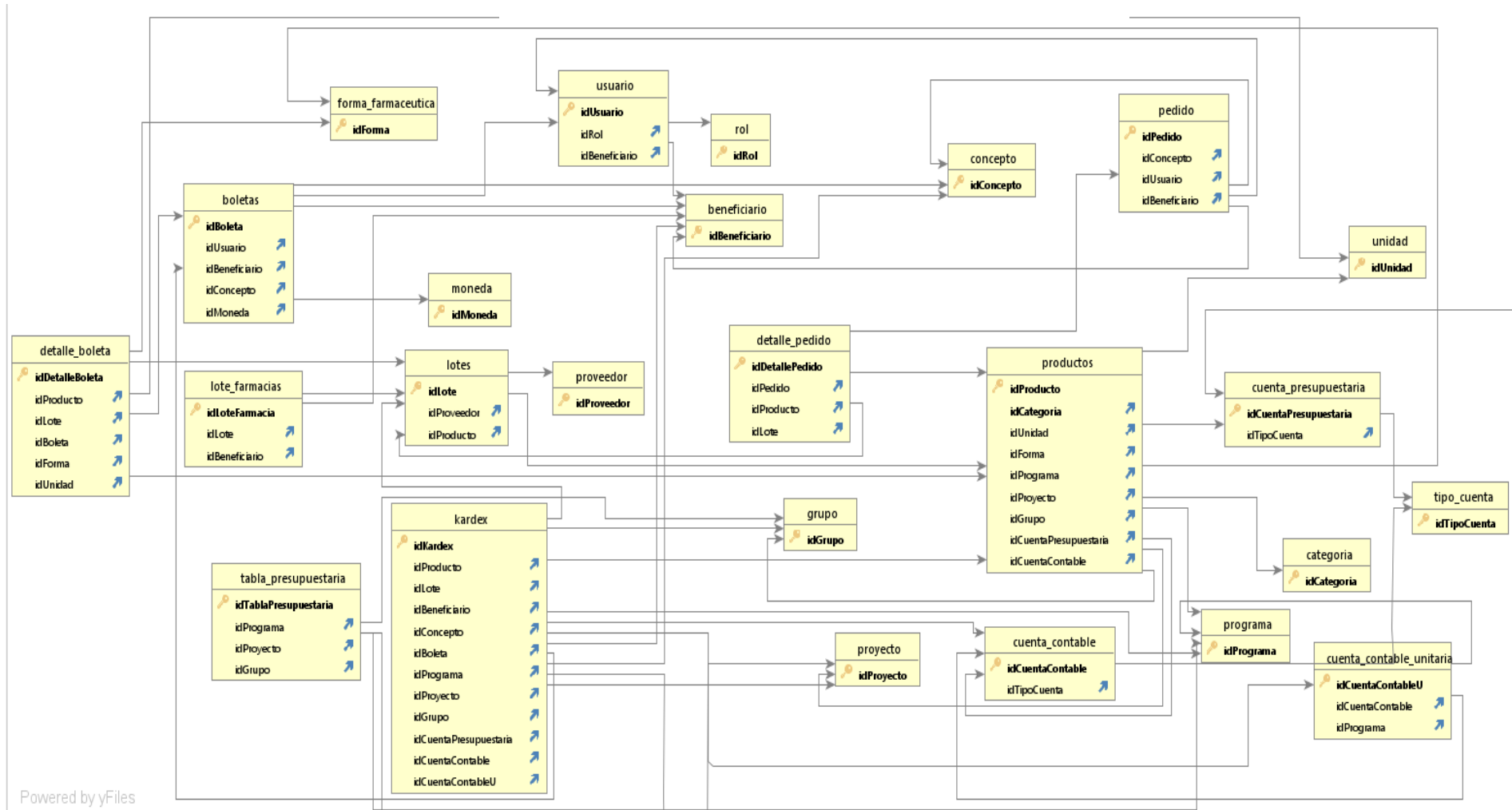
Figura 110. Instalación desde un dispositivo móvil



Fuente: Elaboración propia

Anexo 3: Modelo Físico

Figura 111. Modelo físico



Powered by yFiles

Fuente: Elaboración propia