



UNIVERSIDAD TÉCNICA DE AMBATO

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

CARRERA DE SOFTWARE

Tema:

**MÓDULO DE FACTURACIÓN ELECTRÓNICA ADAPTABLE AL
SISTEMA “ATIENDO” Y EVALÚO DE RENDIMIENTO BAJO LA
NORMATIVA ISO/IEC 25010 PARA LA EMPRESA PLASTICAUCHO
INDUSTRIAL S.A.**

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero de Software

AREA: Software

LINEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Daniel Eduardo Montero López

TUTOR: Ing. Oscar Fernando Ibarra Mg.

Ambato – Ecuador

agosto – 2024

APROBACIÓN DEL TUTOR

En calidad de tutor del trabajo de titulación con el tema: MÓDULO DE FACTURACIÓN ELECTRÓNICA ADAPTABLE AL SISTEMA “ATIENDO” Y EVALÚO DE RENDIMIENTO BAJO LA NORMATIVA ISO/IEC 25010 PARA LA EMPRESA PLASTICAUCHO INDUSTRIAL S.A., desarrollado bajo la modalidad Proyecto de Investigación por el señor Daniel Eduardo Montero López, estudiante de la Carrera de Ingeniería de Software, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, agosto 2024.

Ing. Oscar Fernando Ibarra Torres, Mg.

TUTOR

AUTORÍA

El presente trabajo de titulación con el tema: MÓDULO DE FACTURACIÓN ELECTRÓNICA ADAPTABLE AL SISTEMA “ATIENDO” Y EVALÚO DE RENDIMIENTO BAJO LA NORMATIVA ISO/IEC 25010 PARA LA EMPRESA PLASTICAUCHO INDUSTRIAL S.A., es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2024.



Daniel Eduardo Montero López

C.C. 1850294875

AUTOR

DERECHOS DE AUTOR

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

Ambato, agosto 2024.



Daniel Eduardo Montero López

C.C. 1850294875

AUTOR

APROBACIÓN DEL TRIBUNAL DE GRADO

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Daniel Eduardo Montero López, estudiante de la Carrera de Software de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado MÓDULO DE FACTURACIÓN ELECTRÓNICA ADAPTABLE AL SISTEMA “ATIENDO” Y EVALÚO DE RENDIMIENTO BAJO LA NORMATIVA ISO/IEC 25010 PARA LA EMPRESA PLASTICAUCHO INDUSTRIAL S.A., nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, agosto 2024.

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

Ing. Marco Guachimboza Villalba Mg.
PROFESOR CALIFICADOR

Ing. Carlos Núñez Miranda Mg.
PROFESOR CALIFICADOR

DEDICATORIA

Dedico este trabajo a mis padres, Milton y Elena, los cuales me brindaron todo el apoyo necesario a lo largo de mi carrera universitaria, y cuyo soporte fue indispensable para seguir adelante día tras día. A mis hermanos David y Pablo quienes con sus palabras fortalecieron mi trabajo. De igual manera a mi abuelo José que con su amor incondicional me ayudó a sobrellevar cada etapa de mi vida.

También dedico este trabajo a mi novia Ariana, quien estuvo siempre a mi lado incluso en los momentos más difíciles, y que con su amor y cariño supo guiarme en todos los pasos que daba en mi camino.

Finalmente, dedico este y todos mis trabajos a la memoria de mis dos mascotas Perlita y Candy quienes fueron mis compañeras fieles en la vida y que me supieron enseñar lo que significa el verdadero amor.

Daniel Eduardo Montero López

AGRADECIMIENTO

Primero y, ante todo, agradezco a Dios por haberme brindado la fortaleza y sabiduría para alcanzar mis metas y superar los desafíos en el transcurso de ellas. A mis padres Milton, Elena y hermanos David y Pablo, quienes han sido una fuente inagotable de inspiración. Gracias por creer en mí y por enseñarme el valor del esfuerzo y la dedicación.

Agradezco infinitamente a mi novia Ari, por su paciencia, cariño y por estar siempre a mi lado, brindándome ánimo en los momentos difíciles y celebrando conmigo cada logro.

A mi tutor académico, el Ing. Fernando Ibarra quien con su orientación y vocación me ayudó a desarrollar este trabajo. También agradezco a Plasticaucho Industrial S.A. y especialmente a Vane por darme la oportunidad y confianza en el transcurso de mi vida profesional.

Finalmente, agradezco a todos mis fieles amigos, compañeros de la empresa y personas que estuvieron junto a mí en todo este proceso, sin ustedes esto no fuera posible.

Daniel Eduardo Montero López

ÍNDICE GENERAL DE CONTENIDOS

PORTADA.....	i
APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
DERECHOS DE AUTOR	iv
APROBACIÓN DEL TRIBUNAL DE GRADO.....	v
DEDICATORIA.....	vi
AGRADECIMIENTO	vii
ÍNDICE GENERAL DE CONTENIDOS	viii
ÍNDICE DE TABLAS.....	xiii
ÍNDICE DE FIGURAS.....	xiv
ÍNDICE DE ANEXOS.....	xx
RESUMEN EJECUTIVO	xxi
ABSTRACT	xxii
CAPÍTULO I. MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos	2
1.3 Fundamentación teórica	4
1.3.1 Facturación Electrónica.....	4

1.3.2 Servicios Web	5
1.3.3 Base de Datos	5
1.3.4 SQL	5
1.3.5 Firma Digital	6
1.3.6 Estandar Xades_BES	6
1.3.7 Apache JMeter	6
1.3.8 Modelo de Desarrollo Incremental.....	7
1.3.9 Angular.....	8
1.3.10 C# y ASP.NET	10
1.3.11 ISO/IEC 25010.....	12
1.3.12 Eficiencia de Desempeño	15
1.3.13 MySQL.....	16
1.4 Objetivos	18
1.4.1 Objetivo general	18
1.4.2 Objetivos específicos	18
CAPÍTULO II. METODOLOGÍA.....	19
2.1 Materiales.....	19
2.2 Métodos.....	21
2.2.1 Modalidad de la investigación	21
2.2.1.1 Modalidad bibliográfica documental	21
2.2.1.2 Modalidad aplicada	22

2.2.1.3 Modalidad de campo	22
2.2.2 Población y muestra	22
2.2.2.1 Población.....	22
2.2.2.2 Muestra.....	22
2.2.3 Recolección de información.....	23
2.2.3.1 Procedimiento	25
2.2.4 Procesamiento y análisis de datos	38
CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....	39
3.1 Análisis y discusión de los resultados.....	39
3.1.1 Diagrama del proceso de gestión actual.....	39
3.1.2 Work Breakdown Structure (WBS)	41
3.1.3 Análisis y selección de la metodología de desarrollo.	43
3.1.4 Análisis y selección del framework web frontend.	44
3.1.5 Análisis y selección del framework web backend.	46
3.2 Desarrollo de la propuesta.....	47
3.2.1. Recopilación de requerimientos.....	47
3.2.1.1. Definición de roles	47
3.2.1.2 Requerimientos Técnicos	48
3.2.1.3 Listado de Requerimientos.....	48
3.2.1.4 Plan de Entrega	51
3.2.1.5 Arquitectura del sistema.....	52

3.2.2 Aplicación del primer incremento (Incremento 1).....	55
3.2.2.1 Comunicación	55
3.2.2.2 Planeación	55
3.2.2.3 Modelado	56
3.2.2.4. Construcción	63
3.2.2.5. Despliegue.....	79
3.2.3 Aplicación del segundo incremento (Incremento 2)	81
3.2.3.1 Comunicación	81
3.2.3.2 Planeación	81
3.2.2.3 Modelado	82
3.2.3.4. Construcción	84
3.2.3.5. Despliegue.....	94
3.2.4 Aplicación del tercer incremento (Incremento 3)	96
3.2.4.1 Comunicación	96
3.2.4.2 Planeación	96
3.2.4.3 Modelado	97
3.2.4.4. Construcción	99
3.2.4.5. Despliegue.....	108
3.3. Evaluación aplicando la norma ISO/IEC 25010	110
3.3.1 Norma ISO/IEC 25010.....	110
3.3.2 Pruebas de Carga.....	114

3.3.3 Pruebas de Estrés.....	119
3.3.4. Análisis de los resultados de la evaluación.	124
CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES	126
4.1 Conclusiones	126
4.2 Recomendaciones.....	128
MATERIALES DE REFERENCIA.....	130
ANEXOS	132

ÍNDICE DE TABLAS

Tabla 1. Estructura y preguntas de la encuesta aplicada.	24
Tabla 2. Resultado de varianzas aplicando Alfa de Cronbach.	27
Tabla 3. Rangos de confiabilidad del Alfa de Cronbach.	28
Tabla 4. Cuadro comparativo de metodologías de desarrollo de software.	43
Tabla 5. Tabla comparativa de frameworks frontend.	44
Tabla 6. Cuadro comparativo de frameworks backend.	46
Tabla 7. Definición de roles.	47
Tabla 8. Requerimientos técnicos.	48
Tabla 9. Lista de requerimientos.	49
Tabla 10. Planificación de la entrega de cada incremento.	51
Tabla 11. Cronograma Incremento 1.	56
Tabla 12. Cronograma Imcremento 2.	82
Tabla 13. Cronograma INcremento 3.	97
Tabla 14. Servicios web aplicables para las pruebas de rendimiento.	112
Tabla 15. Criterios de aceptación ISO/IEC 25010.	113
Tabla 16. Evaluación de resultados.	125

ÍNDICE DE FIGURAS

Figura 1. El modelo de desarrollo de software incremental [5].	8
Figura 2. Resultados de la encuesta aplicada. Item 1.....	28
Figura 3. Resultados de la encuesta aplicada. Item 2.....	29
Figura 4. Resultados de la encuesta aplicada. Item 3.....	30
Figura 5. Resultados de la encuesta aplicada. Item 4.....	30
Figura 6. Resultados de la encuesta aplicada. Item 5.....	31
Figura 7. Resultados de la encuesta aplicada. Item 6.....	32
Figura 8. Resultados de la encuesta aplicada. Item 7.....	33
Figura 9. Resultados de la encuesta aplicada. Item 8.....	34
Figura 10. Resultados de la encuesta aplicada. Item 9.....	35
Figura 11. Resultados de la encuesta aplicada. Item 10.....	35
Figura 12. Resultados de la encuesta aplicada. Item 11.....	36
Figura 13. Diagrama de flujo del primer escenario del usuario.	40
Figura 14. Diagrama de flujo del segundo escenario del usuario.	41
Figura 15. WBS del módulo de facturación electrónica.	42
Figura 16. Arquitectura del sistema y módulo de facturación electrónica.....	53
Figura 17. Tablas en la base de datos.....	54
Figura 18. Wireframe del login del sistema	57
Figura 19. Wireframe del dashboard.....	57

Figura 20. Wireframe del listado de almacenes en la venta.....	58
Figura 21. Wireframe del informe de ventas en el almacén.	59
Figura 22. Wireframe generación de nueva venta.	59
Figura 23. Wireframe de la vista de adición de datos del cliente.	60
Figura 24. Wireframe de visualización y edición de datos del cliente.....	61
Figura 25. Wireframe de la adición de productos a la lista de venta.	62
Figura 26. Wireframe de la vista, edición y eliminación de la lista de venta.	63
Figura 27. Estructura del módulo con servicios y modelos en Angular.	64
Figura 28. Estructura del módulo de facturación en la aplicación ASP. Net.....	65
Figura 29. Ingreso al sistema "aTiendo".	66
Figura 30. Método de encriptación de contraseña.	66
Figura 31. Método de validación de credencial y rol.....	67
Figura 32. Módulo de rutas en Angular.	68
Figura 33. Dashboard del sistema "aTiendo" con módulo ventas.	69
Figura 34. Lógica del dashboard del cliente.	69
Figura 35. Lista de almacenes del usuario.	70
Figura 36. Método para listar almacenes.	70
Figura 37. Vista de informe de ventas con filtros.	71
Figura 38. Método para listar las ventas por almacén.....	72
Figura 39. Método para filtrar por fechas.	72

Figura 40. Generación de nueva venta.	73
Figura 41. Popup de ingreso de datos del cliente.	73
Figura 42. Método para agregar datos de facturación del cliente.	74
Figura 43. Vista de edición de datos del cliente.	75
Figura 44. Método de actualización de datos del cliente.	75
Figura 45. Popup para agregar productos al carrito o lista de venta.	76
Figura 46. Método para agregar productos al detalle de la venta.	77
Figura 47. Botones para editar o eliminar el detalle de la venta.	78
Figura 48. Método para eliminar o modificar el detalle de la venta.	78
Figura 49. Comando para generar carpetas de despliegue en Angular.	79
Figura 50. Despliegue de proyecto en cPanel.	80
Figura 51. Generación de archivos de despliegue en ASP .Net.	80
Figura 52. Wireframe de la aplicación de descuentos y vista de importes.	82
Figura 53. Wireframe para agregar la forma de pago y monto de la factura.	83
Figura 54. Wireframe de la vista para generar y enviar comprobante por correo.	84
Figura 55. Archivo para consumir servicio web del SRI.	85
Figura 56. Estructura de consumo de servicio web de SRI en ASP .Net.	85
Figura 57. Vista para agregar descuentos nominales y porcentuales a la factura.	86
Figura 58. Método para aplicar descuentos y visualizar importes de la factura.	87
Figura 59. Manera de agregar las formas de pago a la factura y abonar.	88

Figura 60. Lógica para agregar las formas de pago y abonar.	89
Figura 61. Método para enviar la factura al SRI.	90
Figura 62. Ejemplo de factura xml.	91
Figura 63. Vista de generación de PDF y envío por correo.	92
Figura 64. Método con la generación y envío de la factura por correo.	93
Figura 65. Ejemplo de envío de factura por correo electrónico.	93
Figura 66. Comando para generar carpetas de despliegue en Angular.	94
Figura 67. Despliegue de proyecto en cPanel.	95
Figura 68. Generación de archivos de despliegue en ASP .Net.	95
Figura 69. Wireframe de la vista de configuración de datos de firma electrónica. ...	97
Figura 70. Wireframe del módulo de Cuadre de Caja.	98
Figura 71. Wireframe del detalle del cuadro de caja.	98
Figura 72. Wireframe del módulo de Movimientos.	99
Figura 73. Estructura de los tres componentes en Angular.	100
Figura 74. Vista de configuración de datos del certificado, correo e imagen de la factura.	101
Figura 75. Método para modificar los datos del certificado digital.	102
Figura 76. Método de encriptación de la contraseña en el backend.	103
Figura 77. Método para modificar los datos del mail de envío de comprobantes. ..	103
Figura 78. Vista detallada del cuadro de caja con filtros.	104

Figura 79. Lógica para visualizar los detalles para el cuadro de caja.	105
Figura 80. Exportar archivo a Excel del cuadro de caja.....	106
Figura 81. Vista del módulo de movimientos	107
Figura 82. Método para listar los movimientos.	107
Figura 83. Despliegue de proyecto en cPanel.	108
Figura 84. Generación de archivos de despliegue del backend.	109
Figura 85. Modelo de pruebas de rendimiento de CPU.	111
Figura 86. Resultados de la prueba del primer servicio web.	114
Figura 87. Tiempo de respuesta del primer servicio en un lapso de tiempo.	114
Figura 88. Uso de CPU con el primer servicio en ejecución.	115
Figura 89. Resultados de la prueba del segundo servicio web.....	115
Figura 90. Tiempo de respuesta del segundo servicio en un lapso de tiempo.	115
Figura 91. Uso de CPU con el segundo servicio en ejecución.....	116
Figura 92. Resultados de la prueba del tercer servicio web.	116
Figura 93. Tiempo de respuesta del tercer servicio en un lapso de tiempo.	116
Figura 94. Uso de CPU con el tercer servicio en ejecución.....	117
Figura 95. Resultados de la prueba del cuarto servicio web.	117
Figura 96. Tiempo de respuesta del cuarto servicio en un lapso de tiempo.....	117
Figura 97. Uso de CPU con el cuarto servicio en ejecución.....	118
Figura 98. Resultados de la prueba del quinto servicio web.....	118

Figura 99. Tiempo de respuesta del quinto servicio en un lapso de tiempo.	118
Figura 100. Uso de CPU con el quinto servicio en ejecución.....	119
Figura 101. Resultados de la prueba del primer servicio web.	119
Figura 102. Tiempo de respuesta del primer servicio en un lapso de tiempo.....	120
Figura 103. Uso de CPU con el primer servicio en ejecución.	120
Figura 104. Resultados de la prueba del segundo servicio web.....	120
Figura 105. Tiempo de respuesta del segundo servicio en un lapso de tiempo.	121
Figura 106. Uso de CPU con el segundo servicio en ejecución.....	121
Figura 107. Resultados de la prueba del tercer servicio web.....	121
Figura 108. Tiempo de respuesta del tercer servicio en un lapso de tiempo.	122
Figura 109. Uso de CPU con el tercer servicio en ejecución.....	122
Figura 110. Resultados de la prueba del cuarto servicio web.....	123
Figura 111. Tiempo de respuesta del cuarto servicio en un lapso de tiempo.....	123
Figura 112. Uso de CPU con el cuarto servicio en ejecución.....	123
Figura 113. Resultados de la prueba del quinto servicio web.....	124
Figura 114. Tiempo de respuesta del quinto servicio en un lapso de tiempo.	124
Figura 115. Uso de CPU con el quinto servicio en ejecución.....	124

ÍNDICE DE ANEXOS

Anexo A. Encuesta para obtener nuevos requerimientos aplicada a los usuarios del sistema “aTiendo”	132
Anexo B. Referencias Asp. NET	134
Anexo C. Entidades del módulo de facturación electrónica.	136
Anexo D. Controladores y servicios web.....	143
Anexo E. Funcionalidades para servicios web del SRI.....	155
Anexo F. Manual de usuario.	158

RESUMEN EJECUTIVO

En la gestión de ventas y la facturación electrónica, es crucial contar con una solución integrada que evite errores, retrasos y problemas de cumplimiento, los cuales pueden afectar negativamente la competitividad de la empresa. La ausencia de un sistema unificado en estos procesos puede resultar en ineficiencias significativas. El proyecto tiene como meta principal desarrollar un módulo de facturación electrónica que se integre perfectamente con el sistema "aTiendo". Además, se evaluó el desempeño del módulo utilizando la normativa ISO/IEC 25010 para asegurar su eficiencia. El desarrollo del módulo involucró el uso de tecnologías como Angular, ASP .Net, phpMyAdmin y Apache JMeter. A través de encuestas y reuniones, se recopilaron los requerimientos del cliente para asegurar que el módulo cumpla con las necesidades específicas de la empresa. Entre las funcionalidades desarrolladas se incluyen la generación y envío de facturas electrónicas, manejo de datos de venta y generación de informes detallados. El proyecto se gestionó con una metodología de desarrollo de software incremental, mediante la definición clara de roles y la creación de un plan de entrega dividido en incrementos. Esto permitió abordar de manera organizada y eficiente las distintas etapas del desarrollo, asegurando la culminación exitosa del módulo. La importancia del desarrollo es que el módulo permita una integración efectiva en el sistema "aTiendo", ofreciendo capacidades avanzadas de listado, gestión y reporte de ventas. Adicionalmente, el módulo facilitará la generación y envío de facturas en formato PDF por correo electrónico, así como la implementación de un sistema de firmas electrónicas para la validación de las facturas.

Palabras clave: Metodología de desarrollo incremental, ISO/IEC 25010, Angular, Asp .NET, Apache JMeter, facturación electrónica, firma electrónica, gestión de ventas.

ABSTRACT

In sales management and electronic invoicing, it is crucial to have an integrated solution that avoids errors, delays and compliance issues, which can negatively affect a company's competitiveness. The absence of a unified system in these processes can result in significant inefficiencies. The main goal of the project is to develop an electronic invoicing module that seamlessly integrates with the "aTiendo" system. In addition, the module's performance was evaluated using ISO/IEC 25010 standards to ensure its efficiency. The development of the module involved the use of technologies such as Angular, ASP .Net, phpMyAdmin and Apache JMeter. Through surveys and meetings, client requirements were gathered to ensure that the module met the specific needs of the company. Functionalities developed included the generation and sending of electronic invoices, sales data management and detailed reporting. The project was managed with an incremental software development methodology, by clearly defining roles and creating a delivery plan divided into increments. This allowed to address in an organized and efficient way the different stages of the development, ensuring the successful completion of the module. The importance of the development is that the module will allow an effective integration with the "aTiendo" system, offering advanced sales listing, management and reporting capabilities. Additionally, the module will facilitate the generation and sending of invoices in PDF format by e-mail, as well as the implementation of an electronic signature system for invoice validation.

Keywords: Incremental development, ISO/IEC 25010, Angular, Asp .NET, Apache JMeter, electronic invoicing, electronic signature, sales management.

CAPÍTULO I. MARCO TEÓRICO

1.1 Tema de investigación

MÓDULO DE FACTURACIÓN ELECTRÓNICA ADAPTABLE AL SISTEMA “ATIENDO” Y EVALÚO DE RENDIMIENTO BAJO LA NORMATIVA ISO/IEC 25010 PARA LA EMPRESA PLASTICAUCHO INDUSTRIAL S.A.

1.1.1 Planteamiento del problema

En el contexto actual del entorno empresarial en el Ecuador, la gestión de ventas y la implementación de la facturación electrónica se han vuelto elementos cruciales para garantizar tanto la eficiencia operativa como el cumplimiento normativo en las empresas. Particularmente, la facturación electrónica se ha convertido en un requisito esencial, impulsado por la necesidad de una mayor transparencia fiscal y la modernización de los procesos comerciales.

No obstante, pese a la creciente importancia de los aspectos mencionados, a menudo las empresas se encuentran con un desafío significativo: la falta de una solución integral que incorpore de manera efectiva y eficiente un módulo de ventas con un sistema de facturación electrónica en un solo ambiente de gestión empresarial. La ausencia de esta integración efectiva se convierte en un obstáculo para obtener los objetivos de eficiencia operativa y cumplimiento normativo que toda empresa debe alcanzar.

Consecuentemente, la carencia de unificación entre la gestión de ventas y la facturación electrónica puede generar errores en los registros, retrasos en la emisión de facturas y dificultades en el seguimiento de las transacciones comerciales. Esto no solo aumenta el riesgo de incumplimiento de las regulaciones fiscales, sino que también puede afectar negativamente la reputación de la empresa y su capacidad para competir en un mercado cada vez más exigente.

En Plasticaucho Industrial S.A., la causa principal a abordar es la falta de una solución integral que integre un módulo de ventas con un sistema de facturación electrónica dentro de un solo ambiente de gestión empresarial. Este problema surge en un contexto donde la gestión de ventas y la implementación de la facturación electrónica son cruciales para la eficiencia operativa y el cumplimiento normativo, además de tener efectos como: errores en los registros, dificultades en el seguimiento de transacciones comerciales, riesgo de incumplimiento de regulaciones, y reducción de la competitividad.

1.2 Antecedentes investigativos

La facturación electrónica se ha convertido en un término ampliamente referido, además de ser un aspecto obligatorio actualmente en distintos países para muchas empresas; en donde estas últimas han tenido que ajustarse a esta norma dando diferentes puntos de vista con respecto a aquello. Según Bardales y Rojas se indica que en los últimos años se ha nombrado más a menudo la facturación electrónica en América Latina debido a los avances tecnológicos que día a día suceden, y en el ámbito contable y tributario aquello no se podía quedar por fuera [1].

Ecuador implantó el sistema de facturación electrónica como parte de sus esfuerzos por mejorar el cumplimiento de las obligaciones tributarias y hacer frente al déficit fiscal. El gobierno reconoció la necesidad de un sistema administrativo moderno que promueva la eficiencia, la economía y los avances tecnológicos. Como resultado, se introdujo el sistema de facturación electrónica para agilizar el proceso de recaudación de impuestos y animar a los contribuyentes a cumplir honestamente con sus obligaciones. Este sistema se ha convertido en una práctica habitual no sólo en Ecuador sino en todo el mundo, ya que ofrece numerosas ventajas como la reducción de costes y el ahorro de tiempo en comparación con los métodos tradicionales basados en papel.

En el Ecuador se ha acoplado la facturación electrónica para un sinnúmero de empresas y personas como un tema obligatorio, siendo el Servicio de Rentas Internas (SRI) el ente regulador de este sistema. En Portoviejo se realizó un artículo con referencia a la obligación de la facturación electrónica, los autores mencionan que “la facturación

electrónica es la norma donde el estado implementa una herramienta para poder controlar al contribuyente, no necesariamente aquellos que están obligados a llevar contabilidad, simplemente es que, al elaborar la factura electrónica, el ente regulador tributario obliga y autoriza al contribuyente a realizar dicha factura.” [2].

Por otra parte, la implementación de la facturación electrónica en Ecuador ha sido un proceso gradual, que ha permitido a los contribuyentes adaptarse a esta nueva forma de emitir comprobantes de venta. En este contexto, el Servicio de Rentas Internas (SRI) ofrece sistemas de facturación electrónica gratuitos para los pequeños y medianos contribuyentes. Estos sistemas son una alternativa a los facturadores de pago creados por empresas, que ofrecen una gama más amplia de funcionalidades y servicios [3]. Cabe recalcar que el SRI también pone a disposición los servicios web necesarios para realizar la facturación electrónica, lo que significa una solución económica frente a los facturadores de pago, los cuales son usados por las empresas que deciden utilizar su propio sistema, el cual tiene sus instrumentos bien definidos.

Consecuentemente, muchas empresas han incorporado a su modelo y gestión de ventas un proveedor de facturación electrónica, el mismo que se encarga de realizar todo el proceso necesario, desde la creación del XML hasta el envío y autorización del comprobante por parte del SRI. En el Ecuador este tipo de sistemas son de paga debido a la agilidad con la que se realiza el proceso, una de estas empresas es FacturaSoft, la cual explica que el proceso que realizan es darle al usuario el sistema, luego se registra este último y la subida de una firma electrónica [4]. Este proceso es ágil y rápido, pero de pago, lo que representa un gasto para las personas o empresas que deciden implementar este tipo de software.

El sistema “aTiendo” es un sistema web que actualmente sirve como aplicación de gestión de inventarios para usuarios que buscan controlar su negocio y, por consiguiente, el inventario de sus productos dentro de su negocio independientemente de que cuente con un sistema propio o no. Por consecuencia, los requerimientos principales de este sistema se enfocan en únicamente la gestión ágil que se brinda a los usuarios para manejar los productos existentes en su tienda o negocio.

Finalmente, la incorporación de los módulos existentes dentro del sistema como, compras, gestión de almacenes, productos e inventario, se ha acoplado en el transcurso del plan y cronograma de entrega; los mismos que se han basado en una metodología de desarrollo de software incremental y utilizando una arquitectura MVC, manteniendo un entrono frontend con la tecnología de Angular, y otro entorno backend con la tecnología de C# Asp.NET. Los datos del sistema se alojan en una base de datos relacional MySQL lo que permite una ágil conexión de datos en los procesos embebidos del sistema.

1.3 Fundamentación teórica

1.3.1 Facturación Electrónica

La factura electrónica ha sido definida de muchas maneras por distintos autores. Según Beltrán, “una factura electrónica es un documento digital legalmente válido y sirve como respaldo de las operaciones comerciales entre contribuyentes, por lo cual reemplaza a las facturas tradicionales de papel” [5]. Lo que quiere exponer el autor con esta definición es que la factura electrónica es igual que una física ya que tiene toda la validez y legislación necesaria.

El Servicio de Rentas Internas explica que la factura electrónica es un documento digital que tiene la misma validez que una factura en papel. Se genera y se envía a través de Internet, y cumple con los mismos requisitos legales que una factura tradicional [3].

La empresa Contifico también interpreta su propia definición sobre la factura electrónica y menciona que “es el documento que da soporte a las ventas y movimientos financieros hechos en tu empresa y/o negocio” [6]. Esta definición se centra en el soporte a la empresa. Añadiendo más información a las definiciones anteriores.

Uniando las anteriores definiciones, se puede llegar a una general, y es que la facturación electrónica es un sistema de emisión y recepción de comprobantes de venta

electrónicos que reemplaza a los documentos físicos tradicionales como las facturas en papel. Este sistema busca mejorar la eficiencia, transparencia y control fiscal en las transacciones comerciales, además de agilizar los procesos contables y la gestión de documentos.

1.3.2 Servicios Web

Los servicios web son un software intermedio que permite la interacción entre las aplicaciones escritas por diferentes lenguajes, que por lo general se implementan en la nube o internet para su consumo [1]. Estos servicios web son comúnmente conocidos como API's web ya que permiten la comunicación entre aplicaciones o funcionalidades.

Estos servicios web tienen la posibilidad de ser consumidos siempre y cuando se encuentren activos y subidos a la red, ya que utilizar este medio como intercomunicador para realizar una gestión eficiente entre estos componentes.

1.3.3 Base de Datos

Una base de datos permite almacenar información relevante que servirá de gran ayuda a la aplicación móvil. Esta base de datos por lo general se encuentra creada con lenguaje SQL, y para esta ocasión se la creará en un gestor de base de datos relacional.

Según Valverde, Portalanza y Mora, las bases de datos son grandes cantidades de información almacenadas en registros para lograr una mejor eficiencia al momento de ingresar, buscar, actualizar o eliminar la información. En algunos casos la información debe estar interrelacionada para evitar la duplicidad de información y mejor organización de esta, en otros para mejorar el rendimiento algunos desarrolladores no realizan relaciones de datos [8].

1.3.4 SQL

Según Luis Valencia, SQL (Structured Query Language) es el lenguaje estándar de las bases de datos relacionales. Es un lenguaje declarativo que permite especificar diversos tipos de operaciones sobre estas. Se puede combinar operaciones algebraicas y relacionales con distintos operadores adicionales a los existentes, y así definir consultas para recuperar o modificar información de la base de datos

Cabe recalcar que no se puede realizar solamente consultas con SQL, también se incluye comandos, cláusulas, declaraciones y funciones agregadas. Existen pautas para definir, mantener (insertar, actualizar, eliminar) y acceder a bases de datos relacionales [9].

1.3.5 Firma Digital

Una firma digital es un mecanismo criptográfico que permite al emisor de un documento electrónico identificarse y garantizar la integridad del documento [2]. Esta se genera a partir de un certificado digital el cual se puede obtener a través de distintas instituciones públicas y privadas en el Ecuador.

1.3.6 Estandar Xades_BES

Xades es un grupo de firmas avanzadas, basadas en formatos XML; y Xades_BES es una “firma básica que simplemente cumple los requisitos legales de la Directiva para firma electrónica avanzada” [3]. Es decir, este estándar sirve para firmar un documento XML, el cual tiene todos los requerimientos necesarios embebidos en la misma firma, y por consiguiente en el documento.

1.3.7 Apache JMeter

Según el Centro de Pruebas de Excelente “MyTCOE”, Apache JMeterTM es un software de código abierto diseñado puramente en Java, desarrollado por primera vez por Stefano Mazzocchi, integrante de la Apache Software Foundation, diseñado para probar el comportamiento funcional y medir el rendimiento, comúnmente de servicios o comportamientos de páginas web [4].

1.3.8 Modelo de Desarrollo Incremental

Este modelo es una combinación entre el modelo en cascada y el modelo iterativo. Pressman consolida que “el modelo incremental aplica secuencias lineales en forma escalonada a medida que avanza el calendario de actividades. Cada secuencia lineal produce “incrementos” de software susceptibles de entregarse de manera parecida a los incrementos producidos en un flujo de proceso evolutivo” [5].

El modelo de desarrollo de software incremental es un proceso de desarrollo de software en el que el sistema se construye en incrementos o fases. Cada incremento es una versión completa del sistema con un conjunto de características y funcionalidades.

Este modelo es una variante del modelo de desarrollo iterativo y creciente, en el que el sistema se construye a través de un proceso iterativo de desarrollo y entrega. Sin embargo, en el modelo incremental, los incrementos se entregan al cliente de forma independiente, mientras que, en el modelo iterativo y creciente, los incrementos se entregan al cliente como un todo.

El modelo incremental tiene una serie de ventajas, entre las que se incluyen:

- **Riesgo reducido:** El riesgo de fracaso del proyecto se reduce porque el sistema se construye en incrementos más pequeños y manejables.
- **Flexibilidad:** El modelo incremental permite adaptarse a los cambios en los requisitos del cliente durante el desarrollo.
- **Rentabilidad:** El modelo incremental permite entregar el sistema al cliente de forma más rápida y con un menor costo.

El modelo incremental es adecuado para proyectos de software de gran tamaño o complejos. También es adecuado para proyectos en los que los requisitos del cliente son inciertos o están sujetos a cambios.

Un ejemplo de aplicación del modelo incremental es el desarrollo de un sistema de gestión de clientes. El sistema podría construirse en incrementos que incluyan las siguientes características:

- Incremento 1: Creación de un registro de clientes
- Incremento 2: Seguimiento de las ventas
- Incremento 3: Generación de informes

Cada incremento se entregaría al cliente para su revisión y aprobación antes de comenzar el desarrollo del siguiente incremento.

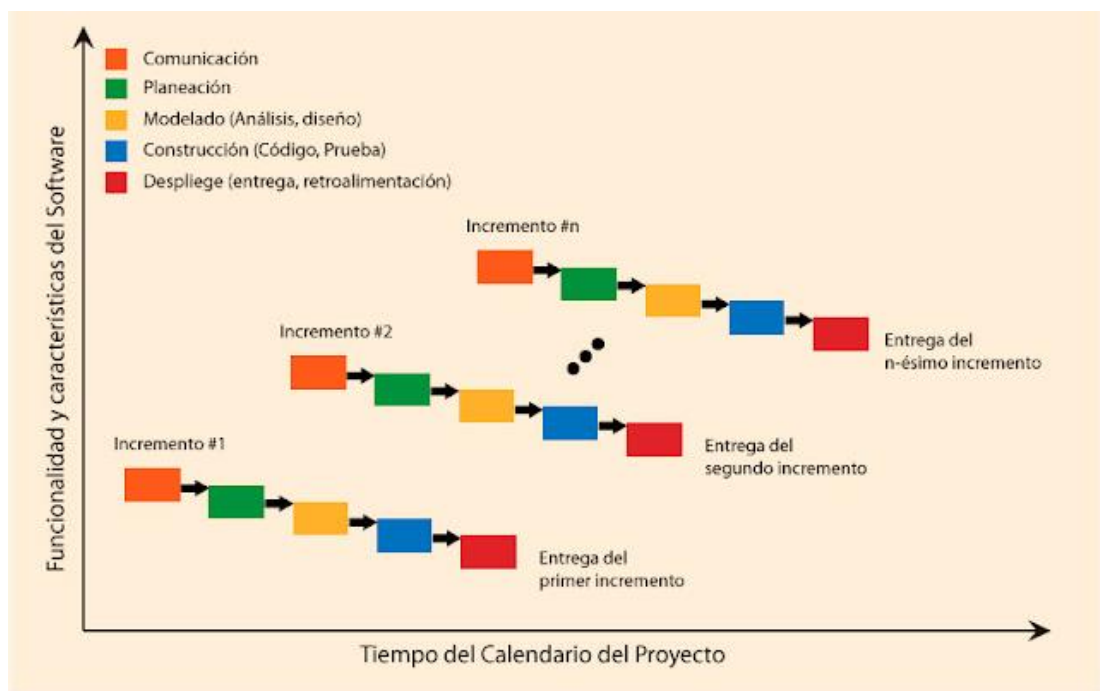


Figura 1. El modelo de desarrollo de software incremental [5].

1.3.9 Angular

Actualmente, para la creación de sitios web se utilizan frameworks para facilitar el proceso; uno de estos es Angular. López y Cáceres aseguran que “Angular es un

framework para el desarrollo de aplicaciones web y de escritorio basado en JavaScript. [6]”. Este ha estado en auge para el mundo del desarrollo web, siendo uno de los más utilizados por los expertos debido a su utilidad y escalabilidad que ofrece.

Angular es un popular framework de desarrollo de aplicaciones web que se utiliza para crear aplicaciones web dinámicas y de una sola página (SPA, por sus siglas en inglés). Angular se basa en TypeScript, un superset de JavaScript, lo que significa que TypeScript es el lenguaje principal utilizado para desarrollar aplicaciones en Angular. A continuación, te proporcionaré una explicación más detallada acerca de Angular y TypeScript:

Angular es un framework desarrollado por Google que simplifica la creación de aplicaciones web complejas y de alta calidad. Con Angular, puedes construir aplicaciones web escalables y mantenibles utilizando componentes reutilizables y una estructura modular. Algunas de las características clave de Angular incluyen enrutamiento, manejo de formularios, comunicación con servidores a través de HTTP, y una amplia comunidad de desarrollo.

TypeScript es un lenguaje de programación desarrollado por Microsoft que se basa en JavaScript pero añade funcionalidades adicionales, como tipado estático. TypeScript es un superset de JavaScript, lo que significa que todo código JavaScript válido también es válido en TypeScript. Sin embargo, TypeScript agrega un sistema de tipos estáticos que ayuda a atrapar errores en tiempo de desarrollo y proporciona un mejor soporte para herramientas de desarrollo como editores de código y IDEs.

Angular utiliza TypeScript como su lenguaje principal por varias razones:

- **Tipado estático:** TypeScript permite definir tipos de datos para variables, parámetros de función y más. Esto ayuda a detectar errores de tipo en tiempo de compilación, lo que conduce a aplicaciones más robustas y menos propensas a errores en tiempo de ejecución.
- **Mejor mantenimiento del código:** El sistema de tipos de TypeScript facilita la lectura y el mantenimiento del código a medida que las aplicaciones crecen

en complejidad. Proporciona una documentación más clara del código y facilita la refactorización.

- **Integración con herramientas:** TypeScript se integra bien con herramientas de desarrollo como Visual Studio Code, proporcionando sugerencias de código, autocompletado y otras características que aumentan la productividad del desarrollador.
- **Mejoras en el rendimiento:** TypeScript ofrece optimizaciones de rendimiento en comparación con JavaScript estándar, lo que puede resultar en aplicaciones más rápidas y eficientes.

1.3.10 C# y ASP.NET

Visual C# .NET es un lenguaje de programación creado por Microsoft para el desarrollo de aplicaciones en la plataforma .NET. Esta plataforma y el lenguaje integrado tienen el potencial de cambiar la forma en que se desarrollan las aplicaciones para Windows e Internet. C# es un lenguaje de programación eficiente y seguro que se basa en la plataforma .NET Framework. Esta plataforma proporciona una amplia gama de herramientas y bibliotecas que facilitan el desarrollo de aplicaciones [7].

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft. Es parte del conjunto de tecnologías de desarrollo de software de Microsoft y se utiliza en una variedad de aplicaciones, desde aplicaciones de escritorio hasta aplicaciones web y móviles. Algunas de las características clave de C# incluyen:

- **Orientación a objetos:** C# es un lenguaje de programación orientado a objetos, lo que significa que se basa en conceptos como clases, objetos, herencia, encapsulamiento y polimorfismo para organizar y estructurar el código.
- **Tipado estático:** C# es un lenguaje de tipado estático, lo que significa que las variables deben declararse con un tipo de datos específico y el tipo de datos de una variable no cambia una vez que se ha definido.

- **Plataforma cruzada:** A partir de .NET Core (y posteriormente .NET 5 y .NET 6), C# es compatible con múltiples plataformas, incluidas Windows, macOS y Linux. Esto permite el desarrollo de aplicaciones multiplataforma.
- **Rico ecosistema de bibliotecas:** C# se beneficia de un extenso ecosistema de bibliotecas y frameworks que simplifican el desarrollo de una amplia variedad de aplicaciones.

ASP.NET es un conjunto de tecnologías de desarrollo web desarrolladas por Microsoft. Se utiliza para crear aplicaciones web dinámicas y robustas. ASP.NET permite el desarrollo de aplicaciones web utilizando varios enfoques, siendo los más comunes ASP.NET Web Forms y ASP.NET MVC (Model-View-Controller). Algunas características importantes de ASP.NET son:

- **Modelo de programación:** ASP.NET ofrece un modelo de programación basado en eventos que se utiliza en ASP.NET Web Forms, así como un enfoque basado en patrones MVC en ASP.NET MVC. Esto proporciona flexibilidad para elegir el enfoque que mejor se adapte a los requisitos del proyecto.
- **Controles y componentes:** ASP.NET proporciona una amplia variedad de controles y componentes que facilitan la creación de aplicaciones web interactivas y ricas en funcionalidad. Esto incluye controles de interfaz de usuario, controles de validación y más.
- **Seguridad:** ASP.NET ofrece características integradas de seguridad, como autenticación y autorización, para proteger las aplicaciones web contra amenazas de seguridad.
- **Integración con C#:** ASP.NET se integra estrechamente con C#, lo que significa que puedes escribir la lógica de servidor de tus aplicaciones web utilizando C#.
- **Rendimiento y escalabilidad:** ASP.NET está diseñado para ser escalable y eficiente, lo que lo hace adecuado para aplicaciones web de alto tráfico y carga de trabajo intensiva.
- **ASP.NET Core:** A partir de ASP.NET Core, Microsoft ha desarrollado una versión más ligera y de código abierto de ASP.NET que es adecuada para aplicaciones web modernas y multiplataforma.

Esta herramienta será utilizada debido a que el sistema se encuentra desarrollado con el lenguaje especificado.

1.3.11 ISO/IEC 25010

La norma ISO/IEC 25010, que se enfoca en la calidad del producto de software y sistemas, es como el "estándar de oro" para evaluar y mejorar la calidad de los productos de software en todo el mundo. A través de sus distintos atributos, esta norma proporciona un marco integral que permite a las organizaciones medir y gestionar la calidad de su software de manera efectiva. [15].

La norma ISO/IEC 25010, también conocida como SQuaRE (Software product Quality Requirements and Evaluation), es una norma internacional que se utiliza para definir y evaluar la calidad del producto de software. Esta norma es parte de la serie ISO/IEC 25000 y proporciona un marco para la gestión de la calidad del software, centrándose específicamente en los requisitos y la evaluación de la calidad del producto de software.

A continuación, se describen los principales aspectos de la norma ISO/IEC 25010:

1. Modelo de calidad: La norma ISO/IEC 25010 define un modelo de calidad que se compone de ocho características de calidad, divididas en dos grupos: características de calidad interna y características de calidad externa. Estas características son:

Características de calidad interna:

- **Mantenibilidad:** La facilidad con la que el software puede modificarse.
- **Eficiencia:** El uso eficiente de recursos como memoria y CPU.
- **Portabilidad:** La capacidad de ejecutarse en diferentes entornos.

Características de calidad externa:

- **Funcionalidad:** El cumplimiento de las funciones requeridas.

- **Fiabilidad:** La capacidad de funcionar sin fallos.
- **Usabilidad:** La facilidad de uso y la satisfacción del usuario.
- **Eficiencia en el uso:** La eficiencia percibida por el usuario.
- **Mantenibilidad:** La facilidad de corrección de errores y actualización.

2. Subcaracterísticas: Cada una de estas características de calidad se desglosa en subcaracterísticas que permiten una evaluación más detallada. Por ejemplo, la característica de "Usabilidad" se descompone en subcaracterísticas como la comprensibilidad, la capacidad de aprendizaje, la eficiencia y la satisfacción del usuario.

3. Criterios de calidad: La norma ISO/IEC 25010 también establece criterios específicos para cada subcaracterística de calidad. Estos criterios proporcionan pautas para evaluar si el producto de software cumple con los estándares de calidad esperados.

4. Proceso de evaluación: La norma ISO/IEC 25010 proporciona directrices para evaluar la calidad del producto de software. Esto implica la recopilación de datos relevantes, la realización de pruebas y evaluaciones, y la comparación de los resultados con los criterios de calidad definidos.

5. Requisitos de calidad: La norma ISO/IEC 25010 también se utiliza para establecer requisitos de calidad para un producto de software. Estos requisitos se derivan de las características y subcaracterísticas de calidad definidas en el modelo.

6. Beneficios: Utilizar la norma ISO/IEC 25010 beneficia a las organizaciones de desarrollo de software al proporcionar una estructura sólida para definir, evaluar y mejorar la calidad de sus productos de software. Ayuda a identificar áreas de mejora y a garantizar que el software cumpla con los estándares de calidad deseados.

Para evaluar el rendimiento de un producto software, se pueden utilizar una serie de métricas que se centran en estas subcaracterísticas. Estas métricas pueden clasificarse en dos categorías principales:

- Métricas de rendimiento absoluto: se centran en el rendimiento del producto en sí mismo, sin tener en cuenta las condiciones ambientales o de carga.
- Métricas de rendimiento relativo: se centran en el rendimiento del producto en relación con otros productos o con los requisitos especificados.

Las métricas de rendimiento absoluto se utilizan para cuantificar el rendimiento del producto en sí mismo, sin tener en cuenta las condiciones ambientales o de carga. Estas métricas pueden utilizarse para comparar el rendimiento de diferentes productos o para identificar áreas de mejora en el rendimiento de un producto concreto.

Algunos ejemplos de métricas de rendimiento absoluto son:

- Tiempo de respuesta: tiempo que tarda el producto en responder a una solicitud.
- Tasa de transferencia: cantidad de datos que el producto puede procesar por unidad de tiempo.
- Uso de la CPU: porcentaje de tiempo que la CPU está ocupada procesando tareas del producto.
- Uso de la memoria: cantidad de memoria que el producto utiliza.
- Uso del almacenamiento: cantidad de almacenamiento que el producto utiliza.

Las métricas de rendimiento relativo se utilizan para cuantificar el rendimiento del producto en relación con otros productos o con los requisitos especificados. Estas métricas pueden utilizarse para evaluar el cumplimiento de los requisitos de rendimiento o para comparar el rendimiento del producto con el de otros productos similares.

Algunos ejemplos de métricas de rendimiento relativo son:

- Relación de rendimiento: relación entre el rendimiento del producto y el rendimiento de un producto de referencia.
- Índice de rendimiento: puntuación que indica el rendimiento del producto en comparación con los requisitos especificados.
- Porcentaje de utilización de los recursos: porcentaje de los recursos disponibles que el producto utiliza.

1.3.12 Eficiencia de Desempeño

La eficiencia de desempeño es una de las características clave de calidad que se abordan en la norma ISO/IEC 25010. Esta característica se centra en la capacidad del software para utilizar recursos de manera eficiente y proporcionar una respuesta rápida y adecuada a las solicitudes del usuario. Aquí se detalla la eficiencia de desempeño según la norma ISO/IEC 25010:

La eficiencia de desempeño se refiere a la capacidad del software para realizar sus funciones de manera eficiente, utilizando la cantidad mínima de recursos, como memoria, CPU y ancho de banda de red. Implica que el software debería ejecutarse sin agotar excesivamente los recursos del sistema y proporcionar una respuesta rápida a las acciones del usuario.

Subcaracterísticas de Eficiencia de Desempeño: Dentro de la característica de Eficiencia de Desempeño, la norma ISO/IEC 25010 identifica varias subcaracterísticas que son esenciales para evaluar esta calidad, algunas de las cuales incluyen:

1. Utilización de recursos: Evalúa si el software utiliza eficientemente los recursos del sistema, como la memoria y la CPU, minimizando el desperdicio.
2. Tiempo de respuesta: Mide el tiempo que tarda el software en responder a las solicitudes del usuario. Un software con buena eficiencia de desempeño debe proporcionar respuestas rápidas.
3. Capacidad de carga: Evalúa la capacidad del software para manejar una carga significativa de trabajo sin degradación significativa del desempeño.
4. Escalabilidad: Se refiere a la capacidad del software para escalar y gestionar un aumento en la carga de trabajo sin problemas.

Criterios de Calidad de Eficiencia de Desempeño: La norma ISO/IEC 25010 también establece criterios específicos para cada una de estas subcaracterísticas. Por ejemplo, en el caso de la utilización de recursos, los criterios podrían incluir limitar el uso de memoria y CPU dentro de ciertos límites aceptables.

Importancia de la Eficiencia de Desempeño: La eficiencia de desempeño es esencial en aplicaciones y sistemas que deben manejar una alta concurrencia de usuarios, como sitios web, aplicaciones en la nube y sistemas de tiempo real. Un software eficiente de desempeño garantiza que los usuarios experimenten tiempos de respuesta rápidos y que los recursos se utilicen de manera efectiva, lo que puede reducir los costos operativos y mejorar la satisfacción del usuario.

Evaluación de Eficiencia de Desempeño: Para evaluar la eficiencia de desempeño de un software, se pueden realizar pruebas de rendimiento y escalabilidad, analizar métricas de recursos como el uso de CPU y memoria, y realizar perfiles de rendimiento para identificar cuellos de botella y áreas de mejora.

1.3.13 MySQL

MySQL es un sistema de gestión de bases de datos (DBMS) ampliamente utilizado en todo el mundo. Se caracteriza por ser un sistema de código abierto, lo que significa que su código fuente está disponible de forma gratuita y se puede modificar y personalizar según las necesidades específicas de cada usuario. A continuación, se proporciona un resumen sobre MySQL:

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) que se originó en Suecia en la década de 1990. Desde entonces, ha sido ampliamente adoptado en la comunidad de desarrollo de software debido a su rendimiento confiable, escalabilidad y facilidad de uso.

Características clave de MySQL:

- **Rendimiento:** MySQL está diseñado para proporcionar un rendimiento rápido y eficiente. Puede manejar grandes cantidades de datos y consultas complejas de manera efectiva.
- **Escalabilidad:** MySQL es altamente escalable y puede utilizarse tanto en aplicaciones pequeñas como en sistemas empresariales de alto tráfico.

- **Confiabilidad:** Es conocido por su estabilidad y capacidad para manejar cargas de trabajo críticas. Ofrece funciones de recuperación y copia de seguridad.
- **Seguridad:** MySQL proporciona opciones avanzadas de seguridad, como autenticación basada en roles y cifrado de datos.
- **Soporte multiplataforma:** Está disponible en varias plataformas, incluyendo Windows, Linux y macOS, lo que permite una amplia adopción.
- **Soporte para múltiples lenguajes de programación:** Puede integrarse fácilmente con muchos lenguajes de programación, como PHP, Python, Java y más.

MySQL utiliza el lenguaje SQL (Structured Query Language) para administrar y manipular datos. Los usuarios pueden realizar consultas SQL para recuperar, insertar, actualizar y eliminar datos de la base de datos.

MySQL proporciona varias herramientas y clientes, como MySQL Workbench y phpMyAdmin, que facilitan la administración de bases de datos y la creación de esquemas.

Este gestor de base de datos posee muchas ventajas, además que el sistema se encuentra desarrollado en un servidor cPanel, por lo cual se utiliza MySQL para almacenar la base de datos y manejar información.

Modelado de Datos: MySQL admite la creación de esquemas de bases de datos complejos, relaciones entre tablas y la definición de claves primarias y foráneas para garantizar la integridad de los datos.

Comunidad y Versión Comercial: Además de la versión de código abierto, MySQL también ofrece una versión comercial llamada MySQL Enterprise Edition que incluye características adicionales y soporte premium.

Compatibilidad con Estándares: MySQL cumple con los estándares SQL definidos por ANSI y proporciona extensiones propias para mejorar la funcionalidad.

1.4 Objetivos

1.4.1 Objetivo general

- Implantar el módulo de Facturación Electrónica al sistema “aTiendo” utilizando un modelo de desarrollo de software incremental y evaluando el rendimiento bajo la normativa ISO/IEC 25010 en la empresa Plasticaucho Industrial S.A.

1.4.2 Objetivos específicos

- Indagar acerca del procedimiento de la facturación electrónica en el Ecuador.
- Desarrollar un módulo de facturación electrónica bajo la metodología de desarrollo de software incremental que cumpla con la ficha técnica del Servicio de Rentas Internas del Ecuador.
- Evaluar el rendimiento del módulo de facturación electrónica bajo la normativa ISO/IEC 25010 utilizando la herramienta JMeter.

CAPÍTULO II. METODOLOGÍA.

2.1 Materiales

En el marco de una investigación sobre la implementación de la facturación electrónica en el sistema web “aTiendo”, se desarrollará un módulo de facturación electrónica adaptable a dicho sistema. Este módulo es el paso para la implementación de la facturación electrónica en el sistema web “aTiendo”, pero es importante evaluar los nuevos requerimientos que los usuarios puedan tener, y para ello, se ha diseñado una encuesta dirigida a usuarios del sistema web “aTiendo”.

El instrumento para la recolección de información mediante la encuesta a usuarios será una encuesta en línea. La encuesta incluirá preguntas sobre las áreas mencionadas anteriormente. Las preguntas serán abiertas y cerradas para permitir a los usuarios expresar sus opiniones y comentarios de manera libre y detallada. Los resultados de la encuesta se analizarán para identificar los nuevos requerimientos con respecto al módulo de facturación electrónica. La información obtenida de esta encuesta será utilizada como justificación a la construcción del módulo.

Por otra parte, para la elaboración del módulo se utilizó la herramienta ASP.Net de Visual Studio como backend, en donde se fabricaron las API. El desarrollo del sistema web se llevó a cabo mediante la metodología incremental, que permite agregar cambios en incrementos sucesivos con el fin de evaluar el retraso y reprogramar las tareas. Para el desarrollo del módulo de facturación electrónica, se utilizó el framework Angular con lenguaje Typescript, la base de datos de MySQL alojada en un servidor cPanel, y librerías para la firma electrónica como Firma_XADES para Visual Studio.

Además, se utilizó un enfoque cuantitativo para medir el rendimiento del sistema. Para ello, se utilizaron las siguientes métricas:

- Tiempo de respuesta: El tiempo que tarda el sistema en responder a cierto número de solicitudes.

- Tiempo de espera: El tiempo que tarda el usuario en obtener una respuesta del sistema.
- Rendimiento: La capacidad del sistema para procesar una cantidad determinada de solicitudes en un período de tiempo determinado.

Para la obtención de datos se utilizó la herramienta apache JMeter, que permite simular peticiones de carga al servidor a través de un navegador Firefox. JMeter se utilizó para generar un conjunto de peticiones de carga, que se enviaron al sistema web a intervalos regulares. Los resultados de JMeter se utilizaron para analizar el rendimiento del sistema.

De acuerdo con la norma ISO/IEC 25010, la eficiencia de desempeño es la capacidad de un sistema para cumplir con sus requisitos de desempeño con un uso eficiente de los recursos. En el caso del proyecto de titulación sobre la implementación de la facturación electrónica en el sistema web “aTiendo”, el objetivo es evaluar el rendimiento del módulo de facturación electrónica para garantizar que cumple con los requisitos de los usuarios y las normas de la industria.

Para ello, se utilizarán las siguientes métricas de rendimiento:

- Tiempo de respuesta
- Tiempo de espera
- Rendimiento

Estas métricas se han seleccionado porque son relevantes para evaluar la eficiencia de desempeño del módulo de facturación electrónica. El tiempo de respuesta es importante para garantizar que el sistema sea responsive y que los usuarios puedan interactuar con él de manera eficiente. El tiempo de espera es importante para garantizar que los usuarios no tengan que esperar demasiado tiempo para obtener una respuesta del sistema. El rendimiento es importante para garantizar que el sistema pueda manejar una carga de usuarios significativa.

Las pruebas de rendimiento se llevarán a cabo utilizando la herramienta JMeter. JMeter es una herramienta de código abierto que permite simular peticiones de carga

al servidor a través de un navegador Firefox. JMeter se utilizará para generar un conjunto de peticiones de carga, que se enviarán al sistema web a intervalos regulares.

Los resultados de JMeter se utilizarán para analizar el rendimiento del sistema.

El análisis de los resultados de las pruebas de rendimiento se llevará a cabo utilizando la norma ISO/IEC 25010. La norma ISO/IEC 25010 proporciona una serie de pautas para evaluar la eficiencia de desempeño de los sistemas. Estas pautas se utilizarán para determinar si el módulo de facturación electrónica cumple con los requisitos de rendimiento establecidos.

2.2 Métodos

La metodología aplicada en el proyecto de titulación sobre la implementación de la facturación electrónica en el sistema web “aTiendo” se basa en un enfoque cuantitativo, que utiliza datos numéricos para probar hipótesis y analizar resultados.

2.2.1 Modalidad de la investigación

La investigación se basará en un estudio bibliográfico documental, aplicado y experimental.

2.2.1.1 Modalidad bibliográfica documental

Esta modalidad implica la recopilación y análisis de información existente en fuentes bibliográficas y documentos relevantes. Se llevará a cabo un análisis comprensivo de libros, artículos científicos, informes técnicos y otros recursos documentales relacionados con el tema de la facturación y firma electrónica, así como la implementación de tecnologías como la herramienta JMeter, el framework Angular, servicios web en C# .Net, modelo de desarrollo de software incremental, etc. La investigación bibliográfica documental permitirá obtener una base sólida de conocimientos teóricos y antecedentes relacionados con el proyecto propuesto.

2.2.1.2 Modalidad aplicada

Esta modalidad se enfoca en la aplicación práctica de los conocimientos y conceptos teóricos obtenidos a través de la investigación bibliográfica documental. En este caso, se buscará aplicar esos conocimientos al desarrollo del módulo la facturación electrónica implementado a las ventas. Diseñar, poner en práctica y evaluar el módulo son ejemplos de investigación aplicada, cuyo objetivo es proporcionar soluciones reales a los problemas de unificación entre la gestión de ventas y la facturación electrónica.

2.2.1.3 Modalidad de campo

La investigación será de campo ya que se recabará datos dentro de la empresa Plasticaucho Industrial, con el objetivo de cumplir con la etapa de comunicación de los distintos incrementos de la metodología de desarrollo existirá contacto directo con distintos involucrados.

2.2.2 Población y muestra

2.2.2.1 Población

La población involucrada en el proyecto propuesto está compuesta por un grupo finito de 40, el cual se compone por clientes directos o detallistas, de la empresa Plasticaucho Industrial S.A. que usarán el módulo. Esta población se definió a partir del plan inicial del sistema en el cual se propuso llegar a esta cantidad de clientes y mediante el equipo principal de desarrollo se seleccionaron los clientes. Estos serán utilizados como población virtual, es decir, una población simulada por la herramienta JMeter.

2.2.2.2 Muestra

Como se mencionó anteriormente, se tendrá una población conformada por un grupo; a partir de esta se obtendrá la muestra

Para obtener la muestra de la población se utilizará la siguiente fórmula:

$$\text{Tamaño de la muestra} = \frac{\frac{z^2 * p (1 - p)}{e^2}}{1 + \left(\frac{z^2 * p (1 - p)}{e^2 * N}\right)}$$

Teniendo en cuenta que: N = tamaño de la población, e = margen de error expresado con decimales, z = puntuación z, p = desviación estándar.

Por lo tanto, para obtener la muestra se tiene que:

Z = 1,96 (valor que se obtiene a partir del nivel de confianza requerido)

p = 0.5

e = 0.05

N = 40

El valor de la muestra reemplazando en la fórmula es igual a: m=37

2.2.3 Recolección de información

La recolección de información en el proyecto de titulación sobre la implementación de la facturación electrónica en el sistema web “aTiendo” se llevará a cabo de dos maneras:

Encuesta a usuarios

Se utilizará una encuesta a usuarios para identificar las necesidades de los usuarios, con el fin de obtener nuevos requerimientos al proyecto. La encuesta se enviará a una

muestra de 37 clientes que utilizan el sistema web “aTiendo”. La encuesta incluirá preguntas sobre las siguientes áreas:

* Experiencia actual con la facturación electrónica

* Necesidades de facturación electrónica

La encuesta posee la siguiente estructura:

Tabla 1. Estructura y preguntas de la encuesta aplicada.

1.- ¿Qué tan importante es para usted que un sistema de facturación electrónica y gestión de inventario tenga las siguientes características o funcionalidades?					
Generar facturas electrónicas válidas	Muy poco importante	Poco importante	Neutral	Importante	Muy Importante
Gestionar el inventario de manera eficiente	Muy poco importante	Poco importante	Neutral	Importante	Muy Importante
Proporcionar informes útiles para la toma de decisiones	Muy poco importante	Poco importante	Neutral	Importante	Muy Importante
Ser confiable	Muy poco importante	Poco importante	Neutral	Importante	Muy Importante
2.- ¿Cómo utiliza actualmente la facturación y la gestión de inventario?					
¿La realiza manualmente?	Nunca	Casi nunca	A veces	Casi siempre	Siempre
¿Qué sistema de software utiliza actualmente?	Pregunta abierta.				
3.- ¿Qué desafíos o problemas enfrenta con la facturación y la gestión de inventario actualmente?					
¿Cuánto tiempo lleva completar la tarea de facturación?	0 – 1 min	1 – 5 min	5 – 10 min	10 – 20 min	Más de 20 min
¿Cuál es el mayor problema que enfrenta con respecto a la facturación?	Pregunta abierta				
4.- ¿Qué tipo de empresa tiene?	Pequeña		Mediana		Grande

5.- ¿Cuál es su volumen de facturación diario?	0 – 10 facturas	10 – 50 facturas	50 – 100 facturas	100 – 200 facturas	Más de 200 facturas
6.- ¿Qué tipo de productos vende? Seleccione una o más:	Primera necesidad	Tecnología.	Ropa	Calzado	Otro:
7.- ¿Tiene algún comentario o sugerencia adicional?	Pregunta abierta.				

2.2.3.1 Procedimiento

Encuesta a usuarios

El procedimiento para la recolección de información mediante la encuesta a usuarios se llevó a cabo de la siguiente manera:

1. Se diseñó una encuesta que incluya preguntas sobre las áreas mencionadas anteriormente.
2. Se envió la encuesta a una muestra de 37 clientes que utilizan el sistema web “aTiendo”.
3. Se recopiló las respuestas de la encuesta.
4. Se analizó las respuestas de la encuesta para identificar las necesidades de los usuarios con respecto a la facturación electrónica.

Análisis de datos

Se requiere recolectar información para analizar principalmente las respuestas del usuario con respecto a la encuesta, estas respuestas serán tabuladas e interpretadas; además, se debe validar el método de evaluación escogido, el cual es la realización del

Alfa de Cronbach para verificar la confiabilidad que representa la aplicación de la encuesta y sus resultados.

Los datos recopilados mediante la encuesta a usuarios se analizarán estadísticamente. El análisis de datos se llevará a cabo para identificar las necesidades de los usuarios con respecto al módulo de facturación electrónica.

Los datos obtenidos a partir de la encuesta fueron tabulados para visualizar de una mejor manera los nuevos requerimientos, con el fin de conseguir un análisis crítico acerca del nuevo módulo.

Cálculo del Alfa de Cronbach

El Alfa de Cronbach es una herramienta que evalúa la confiabilidad de un conjunto de preguntas (ítems) al medir un mismo constructo psicológico. Indica qué tan consistentes son las respuestas de las personas a estas preguntas, es decir, qué tan estables son en relación con el rasgo o característica que se está midiendo [8].

El Alfa de Cronbach es un método que se utilizó con el fin de analizar y evaluar los resultados de la encuesta aplicada; y en función de los resultados obtenidos se medirá que si es mayor a 0.7 la encuesta es fiable, caso contrario no lo es.

Para calcular el Alfa de Cronbach se procede a obtener las varianzas de cada pregunta esquematizada con la siguiente fórmula:

$$Var = \frac{1}{N-1} \sum (x_i - \bar{x})^2$$

Donde:

N = Número total de la muestra.

x_i = Valor individual del conjunto de datos.

$\bar{x}$ = Valor de la media del conjunto de datos.

Tabla 2. Resultado de varianzas aplicando Alfa de Cronbach.

Pregunta	Varianza
¿Qué tan importante considera que un sistema permita generar facturas electrónicas válidas?	0.46996997
¿Qué tan importante considera que un sistema permita gestionar el inventario de manera eficiente?	1.496996997
¿Qué tan importante considera que un sistema proporcione informes útiles para la toma de decisiones?	1.231231231
¿Qué tan importante considera que un sistema sea confiable?	1.082582583
¿Qué tan seguido realiza la facturación manualmente?	1.774774775
¿Cuánto tiempo lleva completar la tarea de facturación?	0.391891892

Una vez se ha obtenido la varianza de cada pregunta, se procede con el cálculo del Alfa de Cronbach aplicando la siguiente fórmula:

$$\alpha = \frac{K}{K-1} * (1 - \frac{\sum V_i}{V_t})$$

Donde:

K = número total de preguntas o ítems.

$\sum V_i$ = Sumatoria total de la varianza de cada pregunta o ítem.

V_t = Varianza total

$$\alpha = \frac{6}{6-1} * (1 - \frac{6.447447}{16.92492})$$

$$\alpha = 0.7428671805$$

Este cálculo se empata con una tabla de rangos en los que se mide el nivel de confiabilidad de la encuesta:

Tabla 3. Rangos de confiabilidad del Alfa de Cronbach.

Índice	Rango	Nivel de Confiabilidad
1	0.30 o menos	Deficiente
2	0.31 a 0.50	Regular
3	0.51 a 0.70	Bueno
4	0.71 a 0.90	Muy Bueno
5	0.90 a 1	Excelente

Cómo se puede observar en la tabla anterior, el alfa de Cronbach obtenido de la encuesta entra en el cuarto rango de confiabilidad del 0.71 a 0.90 por lo que se infiere que los ítems mantienen un buen nivel de correlación, concluyendo con un nivel muy bueno de confiabilidad.

Resultados de la encuesta aplicada a clientes de Plasticaucho Industrial S.A. que hacen uso del sistema aTiendo

1.- ¿Qué tan importante es para usted que un sistema de facturación electrónica y gestión de inventario tenga las siguientes características o funcionalidades?

Item 1. Generar facturas electrónicas válidas

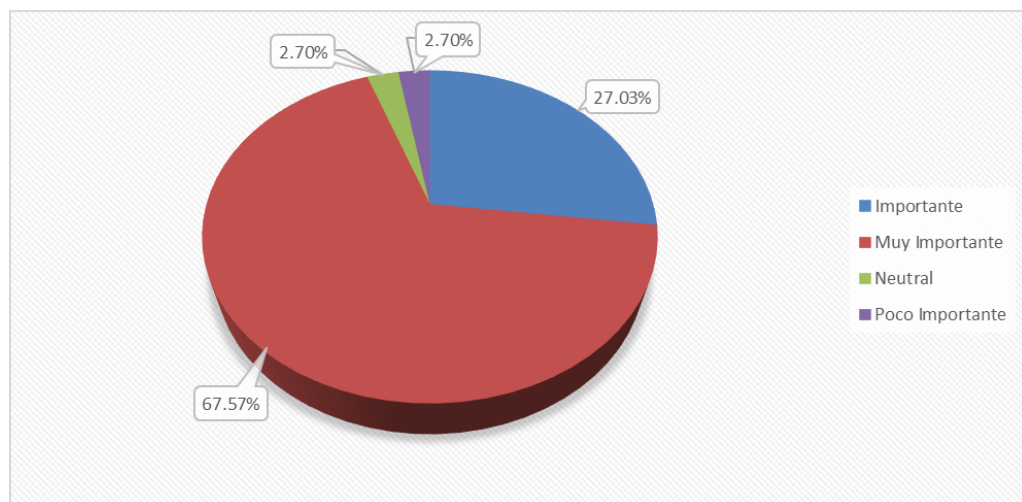


Figura 2. Resultados de la encuesta aplicada. Item 1.

Análisis e interpretación de datos.

Según la figura 2 se puede apreciar que existe un 67.57% que considera muy importante que un sistema de facturación debe emitir facturas electrónicas válidas, el 27.03% lo considera importante, el 2.70% se encuentra en posición neutral, y el 2.70% restante lo considera poco importante; lo que refleja que para el usuario es crucial que el sistema brinde facturas electrónicas válidas.

Item 2. Gestionar el inventario de manera eficiente

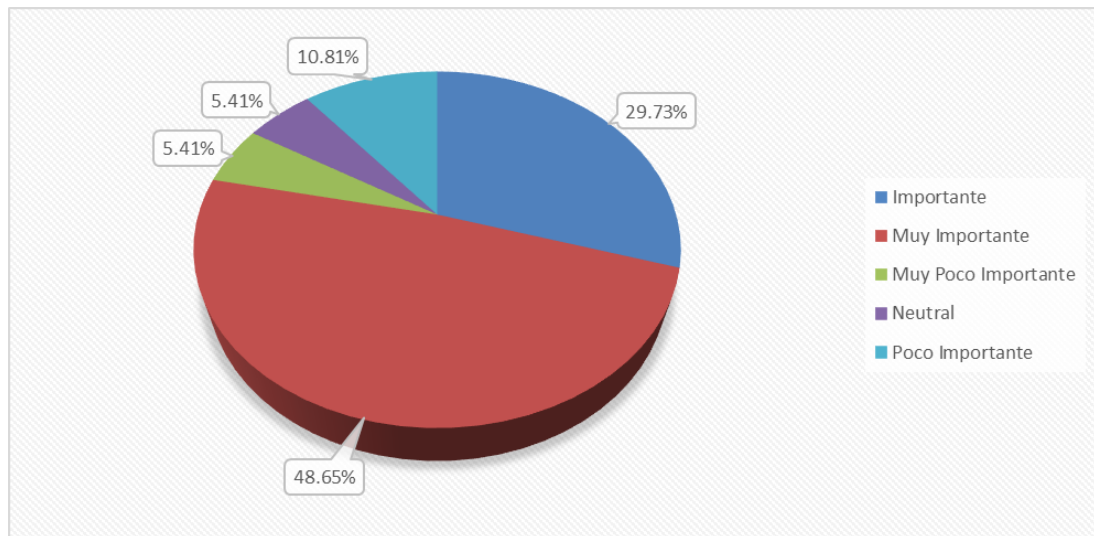


Figura 3. Resultados de la encuesta aplicada. Item 2.

Análisis e interpretación de datos.

Según la figura 3, el porcentaje de usuarios que consideran muy importante que un sistema permita gestionar el inventario de manera eficiente es del 48.65%, por otro lado, el 29.73% restante lo considera importante; mientras que el 5.41% responde de manera neutral, el 10.81% respondió que es poco importante y finalmente el 5.41% lo considera muy poco importante. Estas respuestas reflejan el nivel de importancia alta que representa para el usuario el gestionar el inventario eficientemente.

Item 3. Proporcionar informes útiles para la toma de decisiones

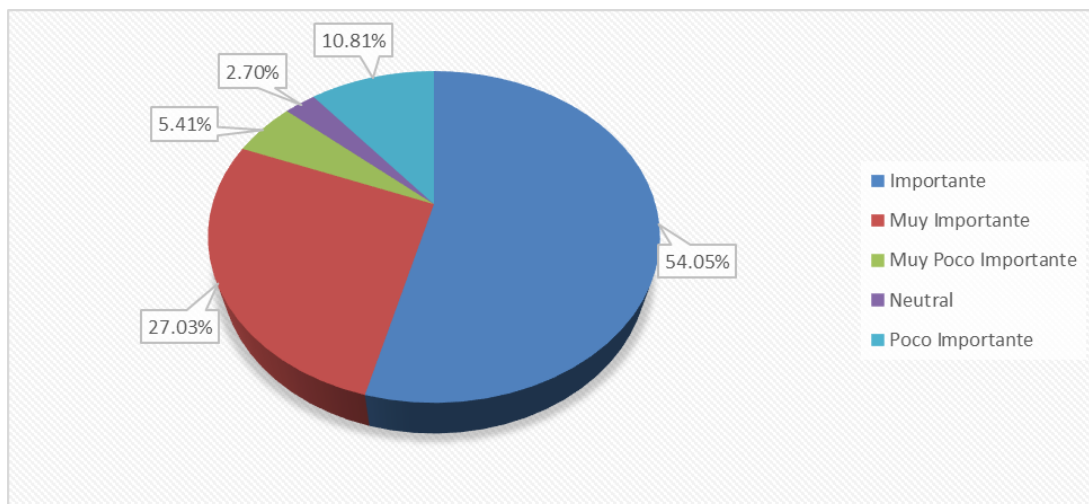


Figura 4. Resultados de la encuesta aplicada. Item 3.

Análisis e interpretación de datos.

Según la figura 4 se puede apreciar que existen cinco grupos de respuestas, el 27.03% menciona que es muy importante que el sistema genere informes para la toma de decisiones. El segundo grupo que forma el 54.05% responden que es importante, el 2.70% tiene una respuesta neutral frente a esta pregunta; el 10.81% responde que es poco importante; y el 5.41% restante lo consideran muy poco importante. Estas respuestas dan a entender que la presentación de informes al usuario del sistema es importante ya que les permitiría registrar un orden.

Item 4. Ser confiable

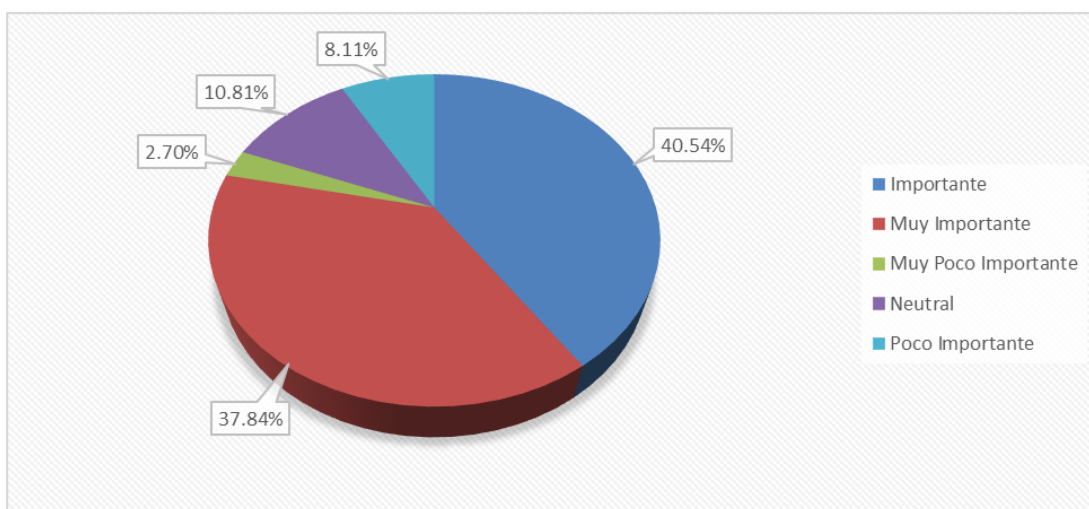


Figura 5. Resultados de la encuesta aplicada. Item 4.

Análisis e interpretación de datos.

Según la figura 5, existe un 37.84% que afirma que es muy importante que el sistema sea confiable, un 40.54% lo considera importante, mientras que un 10.81% se encuentra en una posición neutral; el 8.11% lo considera poco importante y finalmente el 2.70% responde que es muy poco importante. Es importante revisar las respuestas ya que estas demuestran que para el cliente un sistema de facturación debe ser confiable para poder utilizarlo.

2.- ¿Cómo utiliza actualmente la facturación y la gestión de inventario?

Item 5.- ¿Lo realiza manualmente?

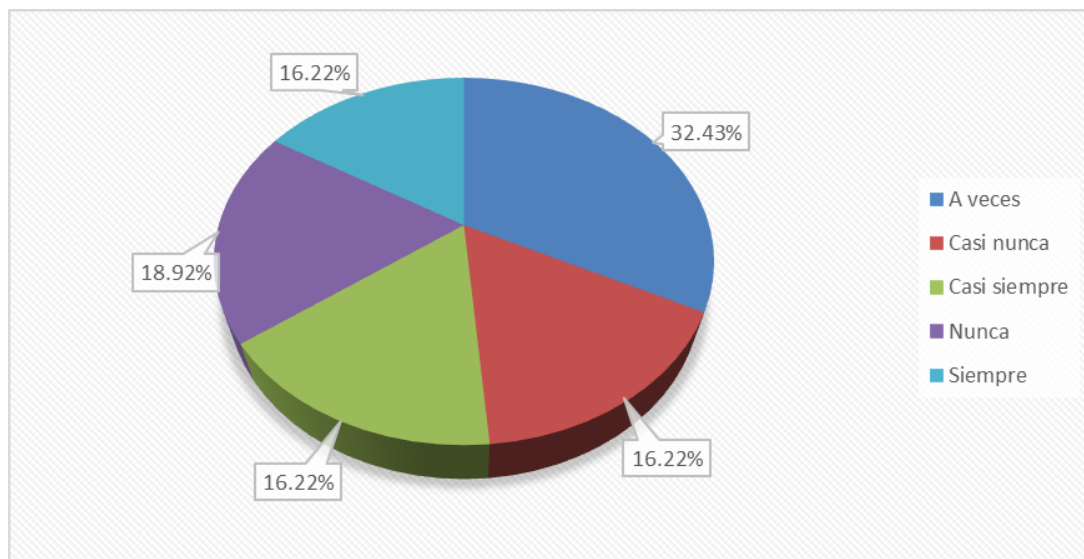


Figura 6. Resultados de la encuesta aplicada. Item 5.

Análisis e interpretación de datos.

Según la figura 6, el 16.22% siempre realiza el proceso de facturación y gestión de inventario de manera manual, el 16.22% casi siempre lo realiza manual, el 32.43% realiza a veces la gestión de manera manual, el 16.22% casi nunca y el 18.92% nunca realiza el proceso de forma manual. Existen diversas respuestas en este ítem debido a la variedad de tipología de cliente, aun así la mayoría aún realiza la facturación manual, lo que puede presentar un vacío legal en sus negocios.

Item 6. ¿Qué sistema de software utiliza actualmente?

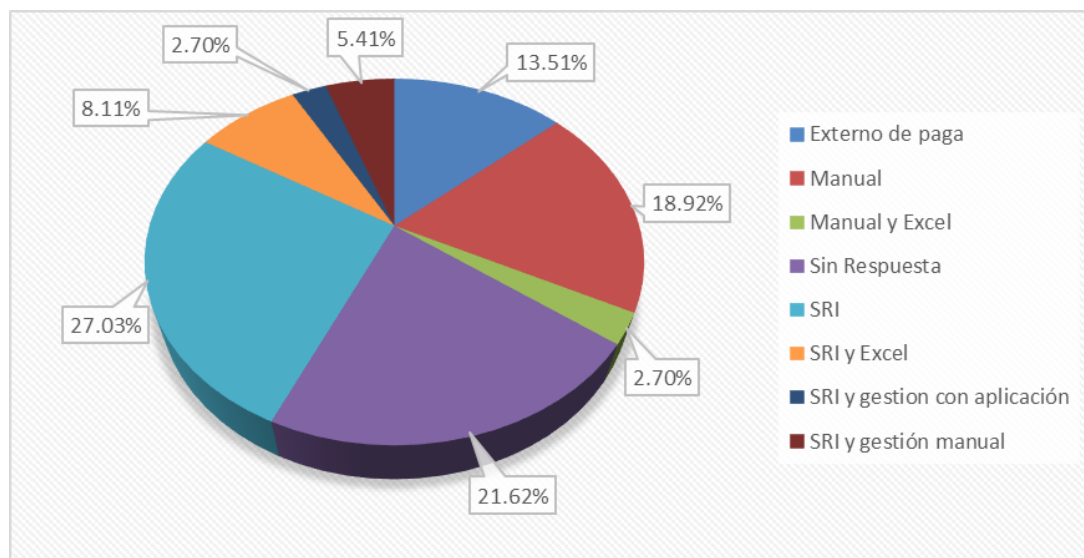


Figura 7. Resultados de la encuesta aplicada. Item 6.

Análisis e interpretación de datos.

Para la anterior pregunta, al ser del tipo abierta, se procedió a tabular de acuerdo con las respuestas obtenidas, determinando así ocho grupos.

Según la figura 7, el 27.03% utiliza únicamente el sistema del SRI para realizar la facturación electrónica; el 8.11% utiliza el mismo sistema gratuito del SRI y también realiza la gestión del inventario con la aplicación de ofimática Excel; por otro lado, está el grupo que realiza la facturación por medio del SRI y la gestión de inventario con una aplicación dedicada perteneciente al 2.70% de la muestra. Además, se encuentra el grupo que realiza la facturación electrónica con el sistema del SRI, pero realiza la gestión del inventario de manera manual. También se encuentra un grupo del 13.51% que utiliza un sistema de facturación externo de paga. De igual manera existe un grupo del 2.70% que realiza facturación manual y gestión de inventario en Excel; e igualmente un grupo del 18.92% que realiza todo de manera manual. Finalmente, el 21.62% restante pertenece al grupo de encuestados que decidieron no responder. Estas respuestas permiten obtener un claro análisis de que la mayoría utiliza los servicios del SRI para enviar y autorizar sus facturas y puede deberse al factor principal de que es gratis; además, la muestra persiste con gestión manual de inventarios y puede deberse

a la facilidad o poca complejidad a corto plazo de sus productos; lo que representa un problema a largo plazo para validación de stock.

3.- ¿Qué desafíos o problemas enfrenta con la facturación y la gestión de inventario actualmente?

Item 7. ¿Cuánto tiempo le lleva completar la tarea de facturación?

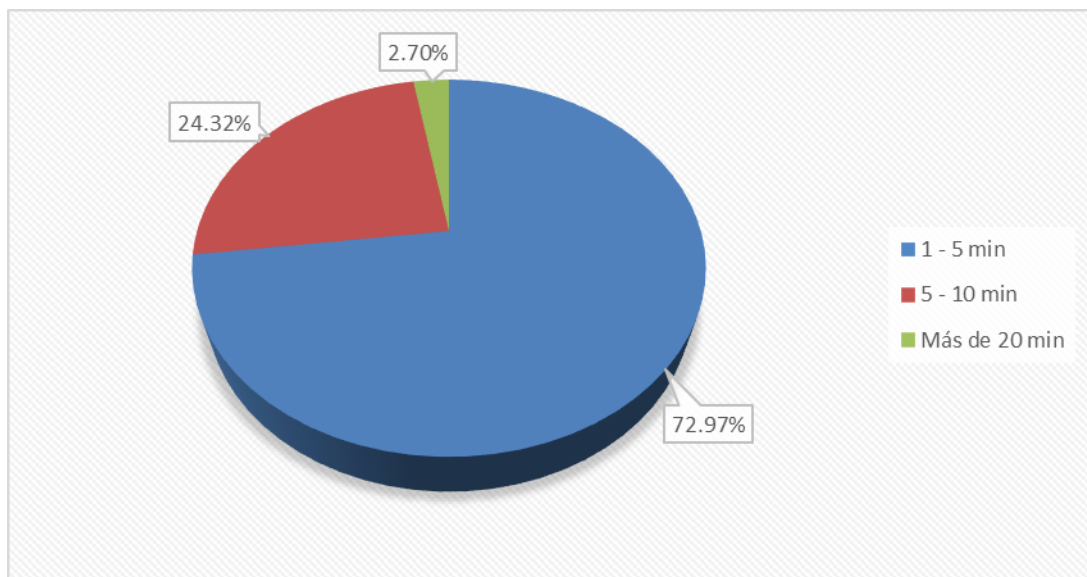


Figura 8. Resultados de la encuesta aplicada. Item 7.

Análisis e interpretación de datos.

Según la figura 8, el 72.97% menciona que le lleva de uno a cinco minutos completar la tarea de facturación, por otro lado, al 24.32% de los encuestados le toma entre cinco a diez minutos realizar este proceso; mientras que el 2.70% restante le toma más de veinte minutos. Según los resultados obtenidos, el tiempo de facturación supera el minuto, lo que representa un tiempo considerable de espera en el proceso de facturación.

Item 8. ¿Cuál es el mayor problema que enfrenta con respecto a la facturación?

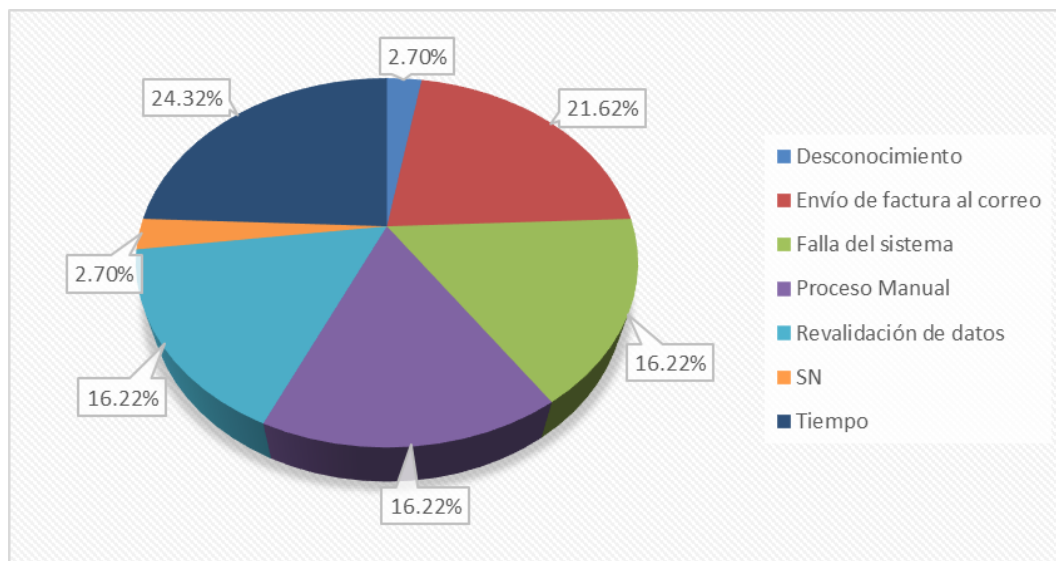


Figura 9. Resultados de la encuesta aplicada. Item 8.

Análisis e interpretación de datos.

Para la anterior pregunta, al ser del tipo abierta, se procedió a tabular de acuerdo con las respuestas obtenidas, determinando así siete grupos.

Según la figura 9, el primer grupo, perteneciente al 24.32% de respuestas menciona que el mayor problema que enfrentan con respecto a la facturación tiene que ver con el tiempo; el segundo conforma un 2.70% y menciona que el problema tiene que ver con el desconocimiento del sistema. Además, se encuentra un amplio grupo del 21.62% que menciona que sus facturas no se envían por correo electrónico o presentan algún problema. Por otra parte, existen tres grupos que conforma un 16.22% cada uno que menciona que el problema en la facturación se presenta al momento de realizar revalidación de datos, que el sistema suele fallar y que tienen que realizar el proceso manual. Finalmente, el 2,70% no presenta ninguna novedad. Estas respuestas presentan una clara deficiencia en temas de tiempo en la facturación, además la falla del sistema también perjudica este proceso.

Item 9. ¿Qué tipo de empresa tiene?

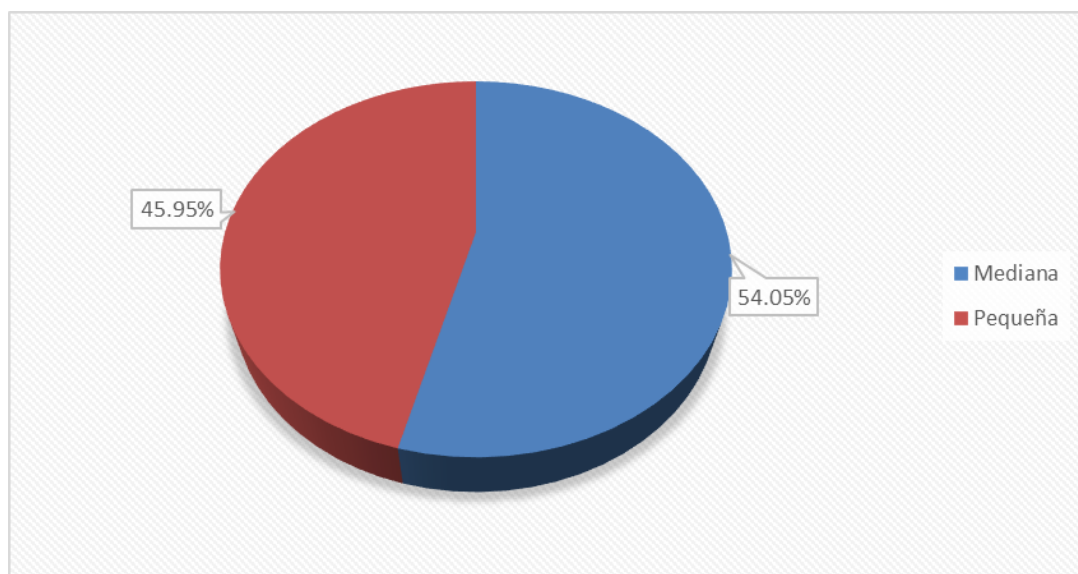


Figura 10. Resultados de la encuesta aplicada. Item 9.

Análisis e interpretación de datos.

Según la figura 10 se tiene dos grupos dentro de los encuestados, el primero que forma un 45.95% menciona que tiene una empresa pequeña, mientras que el 54.05% restante afirma poseer una empresa mediana. Los resultados presentan los dos grupos, y demográficamente se define el alcance del módulo.

Item 10. ¿Cuál es su volumen de facturación diario?

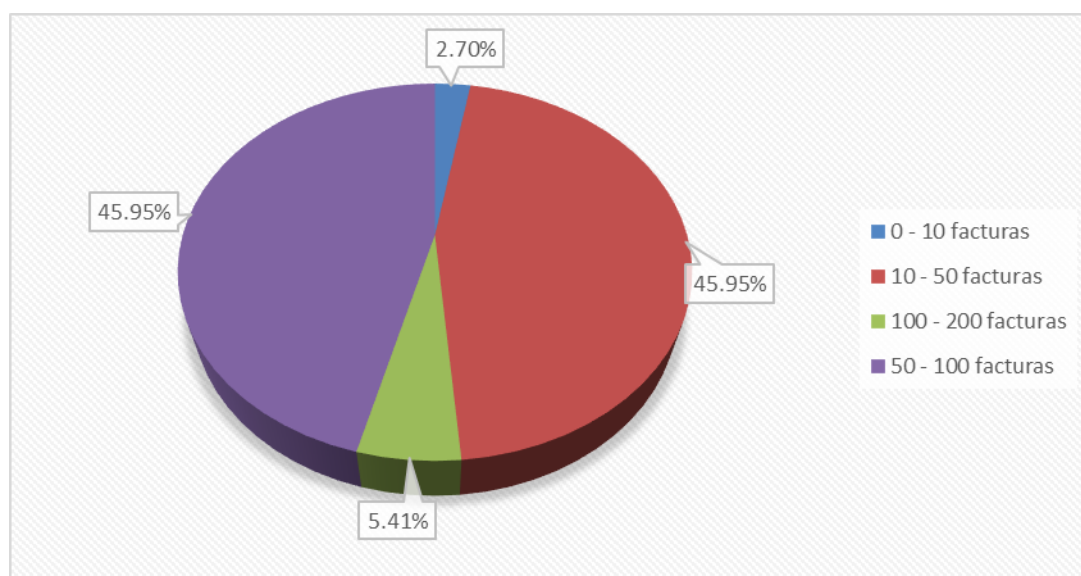


Figura 11. Resultados de la encuesta aplicada. Item 10.

Análisis e interpretación de datos.

Según la figura 11, de todos los encuestados, el 2.70% tiene un volumen de cero a diez facturas diarias, el 45.95% maneja un volumen de diez a cincuenta facturas diarias, el 45.95% considera que maneja de cincuenta a cien facturas diarias, y finalmente el 5.41% restante factura entre cien a doscientas facturas diarias aproximadamente. Al igual que el anterior item, este permite identificar el alcance que se debe tener con el módulo de facturación propuesto.

Item 11. ¿Qué tipo de productos vende?

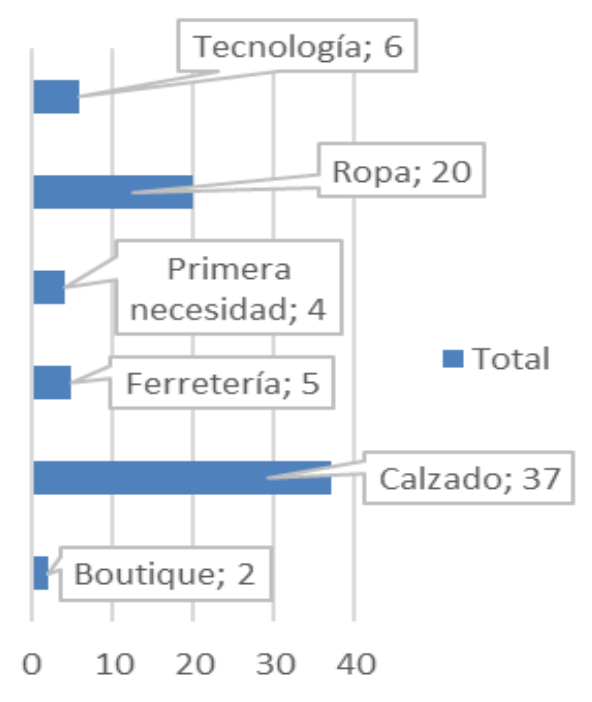


Figura 12. Resultados de la encuesta aplicada. Item 11.

Análisis e interpretación de datos.

Para la anterior pregunta, al ser del tipo abierta, se procedió a tabular de acuerdo con las respuestas obtenidas, determinando así 6 grupos de productos que venden los encuestados.

Según la figura 12, todos los encuestados venden calzado, lo cual se da debido a que son clientes de Plasticaucho Industrial S.A.; por otra parte 20 encuestados ofrecen ropa

o artículos de moda para la venta, además, 6 encuestados ofrecen tecnología, 5 ferretería, 4 primera necesidad y finalmente, 2 encuestados ofrecen artículos de boutique. Al ser clientes directos de Plasticaucho Industrial S.A., ofrecen principalmente calzado lo que da un claro enfoque en este campo.

2.2.4 Procesamiento y análisis de datos

De acuerdo con las encuestas aplicada a clientes de Plasticaucho Industrial S.A. que hacen uso del sistema aTiendo, se concluyó que:

- El proceso actual de facturación en la mayoría de los clientes de Plasticaucho que utilizan el sistema aTiendo es manual y esto representa un riesgo de control y obligatoriedad impuesta por la entidad del Servicio de Rentas Internas.
- El proceso actual de gestión de inventario en los clientes de Plasticaucho que utilizan el sistema aTiendo es variado, entre el proceso manual y dentro de algún sistema, lo cual resalta la importancia de gestionarlo todo con la misma herramienta.
- El módulo de facturación electrónica debe ser adaptado al sistema aTiendo, por lo tanto, será desarrollado para una página web y debe desarrollarse en las herramientas tecnológicas que utiliza la empresa, como Angular, .NET y cPanel. Esta última permite el despliegue de aplicaciones, gestión de archivos y contiene la base de datos MySQL.
- Es fundamental que el módulo permita el registro, edición y consulta de información del inventario.
- Es necesario que la información del inventario sea accesible y debe estar disponible en cualquier momento para el cliente, de esta manera se asegura una gestión eficiente de sus productos y stock.
- Los encuestados consideran que la implementación de un módulo de facturación electrónica es crucial ya que permite reducir la dependencia en procesos manuales y mejorar la eficiencia con un módulo automatizado.
- Es necesario el uso de una misma herramienta en donde se pueda realizar la facturación electrónica y gestionar el inventario ya que brinda accesibilidad e interacción dinámica sin realizar procesos manuales.
- Se debe proporcionar informes útiles al cliente que le permita tomar decisiones sobre sus transacciones.

Con base en la información obtenida, se determinaron los requisitos esenciales para el desarrollo del proyecto, así como las necesidades particulares del cliente para la creación del módulo de facturación electrónica.

CAPÍTULO III. RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de los resultados.

3.1.1 Diagrama del proceso de gestión actual

Actualmente la manera de facturación que aplica la muestra de investigación presenta dos escenarios.

El primer escenario es de la siguiente manera:

1. El consumidor final requiere realizar sus compras en la tienda.
2. El facturador atiende al cliente y verifica stock del producto o material que requiera el consumidor, en donde:
 - a) En el caso de no existir stock el proceso termina.
 - b) En el caso de existir stock se procede a añadir los productos a la lista de venta y restando manualmente el stock.
3. Al consumidor final se le piden los datos de facturación
4. El facturador puede aplicar descuentos porcentuales o nominales de ser necesario.
5. El consumidor final puede elegir su forma de pago (efectivo, tarjeta de débito, tarjeta de crédito).
6. Finalmente, el facturador genera una factura en papel impresa o escrita.

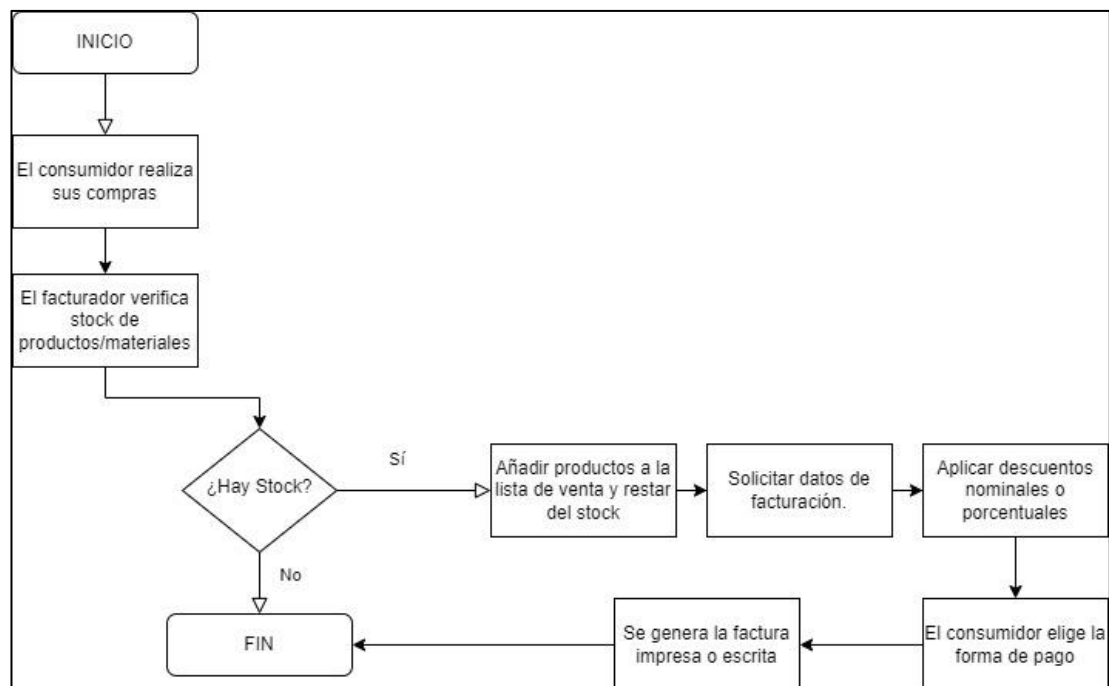


Figura 13. Diagrama de flujo del primer escenario del usuario.

El primer escenario es de la siguiente manera:

1. El consumidor final requiere realizar sus compras en la tienda.
2. El facturador atiende al cliente y verifica stock del producto o material que requiera el consumidor, en donde:
 - c) En el caso de no existir stock el proceso termina.
 - d) En el caso de existir stock se procede a añadir los productos a la lista de venta y restando manualmente el stock.
3. Al consumidor final se le piden los datos de facturación
4. El facturador puede aplicar descuentos porcentuales o nominales de ser necesario.
5. El consumidor final puede elegir su forma de pago (efectivo, tarjeta de débito, tarjeta de crédito).
6. El facturador genera una factura electrónica (archivo xml).
7. Finalmente, se envía la factura mediante correo.

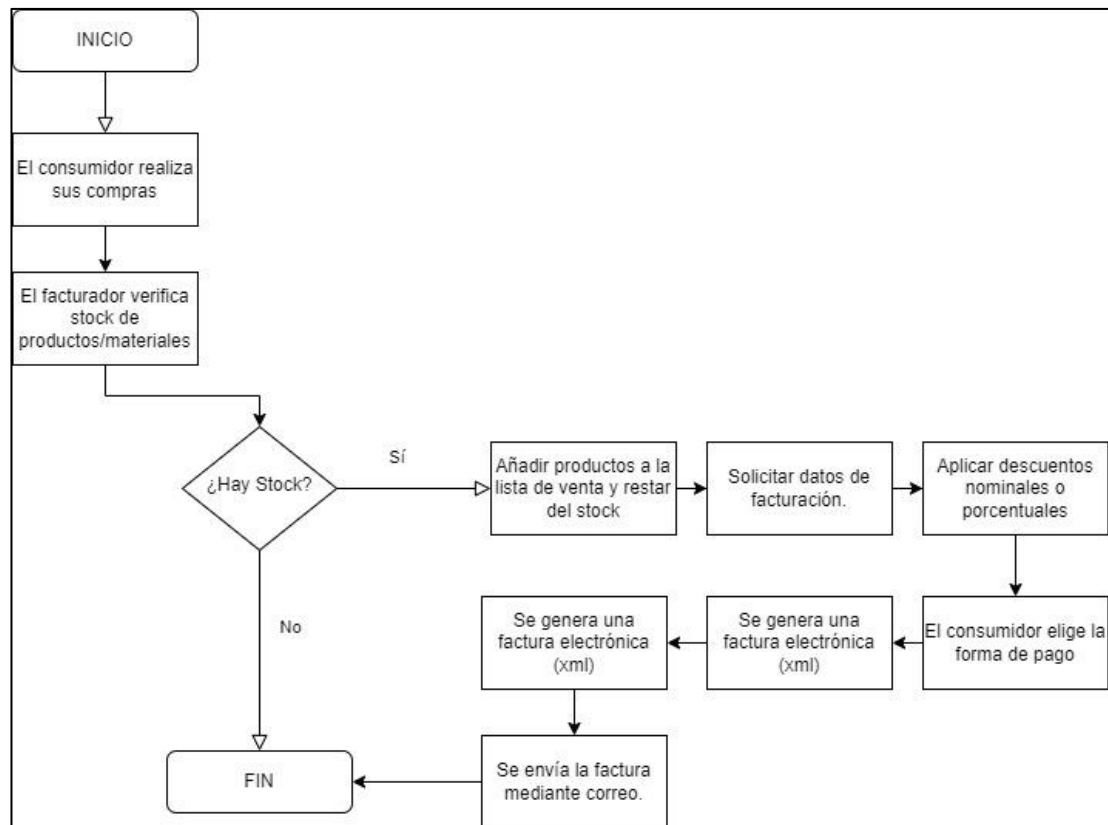


Figura 14. Diagrama de flujo del segundo escenario del usuario.

3.1.2 Work Breakdown Structure (WBS)

Se elaboró un WBS para desglosar el desarrollo del módulo en componentes más pequeños y manejables, permitiendo una mejor visualización de las tareas necesarias para finalizar el proyecto dentro de las fechas establecidas.

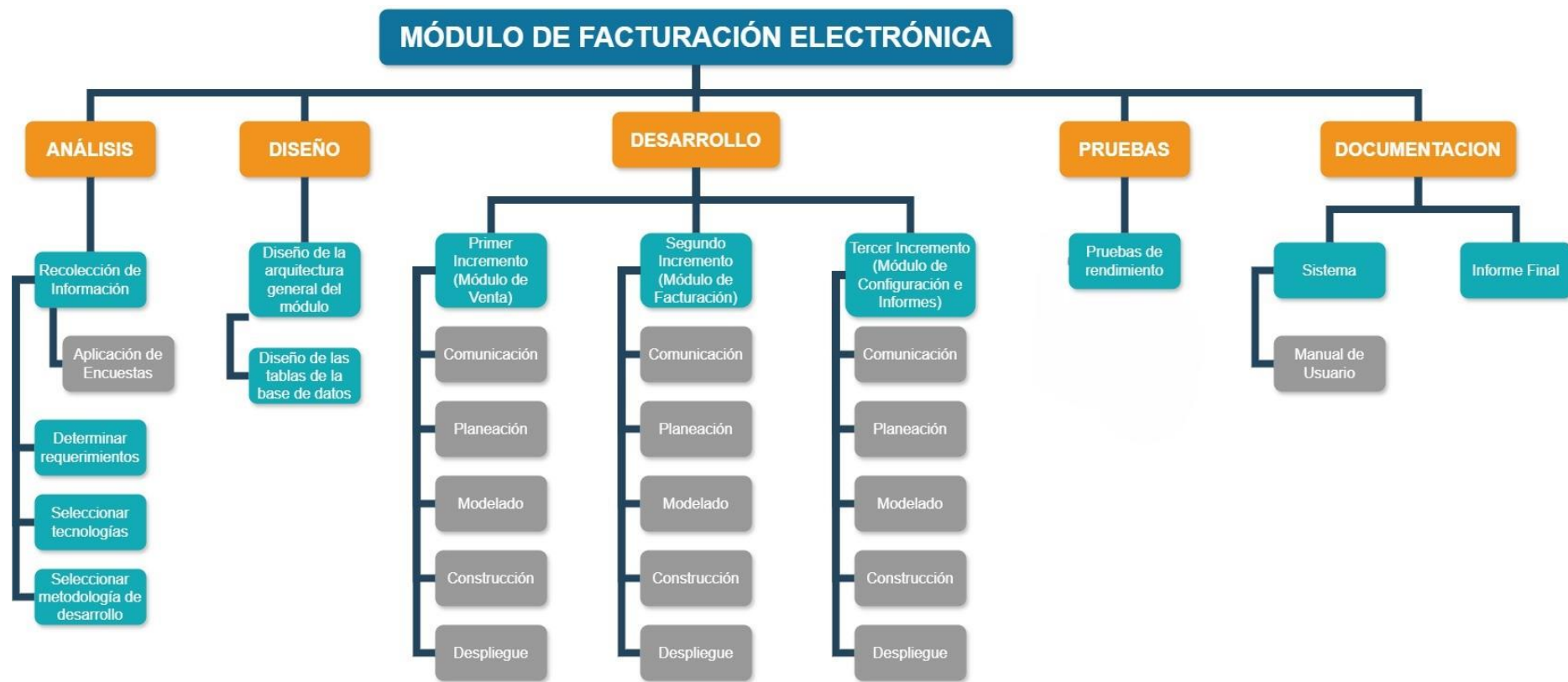


Figura 15. WBS del módulo de facturación electrónica.

3.1.3 Análisis y selección de la metodología de desarrollo.

Tabla 4. Cuadro comparativo de metodologías de desarrollo de software.

Criterio	Metodología Incremental	Metodología en Cascada	Metodología XP (Extreme Programming)	Metodología en Espiral
Enfoque	Desarrolla el sistema en incrementos o versiones parciales	Secuencial, fase por fase	Iterativa e incremental con enfoque en prácticas técnicas rigurosas	Combinación de iterativa de diseño y prototipos
Flexibilidad	Alta, permite ajustes y mejoras en cada incremento	Baja, cambios son costosos y complicados	Alta, se adapta rápidamente a cambios con feedback continuo	Alta, se ajusta en cada iteración
Visibilidad de Progreso	Alta, el cliente ve avances funcionales en cada incremento	Baja, el producto se ve al final	Alta, entregas frecuentes y pequeñas mejoras continuas	Alta, cada ciclo produce un prototipo mejorado
Riesgos	Bajo, los riesgos se identifican y mitigan en cada incremento	Altos, se identifican tarde en el ciclo	Bajos, mitigados mediante desarrollo continuo y pruebas constantes	Moderados, se evalúan en cada ciclo
Retroalimentación del Cliente	Continua, en cada incremento	Tardía, hasta que se completa el sistema	Continua, feedback constante del cliente en cada iteración	Continua, en cada ciclo
Complejidad del Proyecto	Adecuada para proyectos medianos a grandes con requisitos complejos	Baja a media, proyectos pequeños a medianos	Adecuada para proyectos de cualquier tamaño, especialmente	Alta, ideal para proyectos grandes y complejos

			con requisitos cambiantes	
Control y Planificación	Alta, planificación detallada para cada incremento	Alta, planificación detallada inicial	Moderada, planificación constante y adaptable a cambios	Alta, cada ciclo se planifica y controla

Metodología de desarrollo de software elegida

Luego de haber realizado una comparativa exhaustiva de diversas metodologías, tales como Cascada, Programación Extrema (XP), Incremental y en Espiral, se decidió optar por la metodología Incremental para el desarrollo del aplicativo web. Esta elección se basa en su enfoque iterativo, que permite desarrollar el módulo en incrementos o versiones parciales, ofreciendo entregas regulares de componentes funcionales. Esta metodología proporciona alta flexibilidad para realizar ajustes y mejoras continuas, y permite una visibilidad constante del progreso. Además, la retroalimentación continua del cliente en cada incremento y la capacidad de identificar y mitigar riesgos de manera temprana fueron factores determinantes en la elección de esta metodología.

3.1.4 Análisis y selección del framework web frontend.

Tabla 5. Tabla comparativa de frameworks frontend.

Criterio	Angular	React	Vue.js
Arquitectura	MVC (Model-View-Controller)	Biblioteca centrada en la vista (View Library)	MVVM (Model-View-ViewModel)
Complejidad	Alta, debido a su estructura completa y todas sus funcionalidades integradas	Moderada, depende de las bibliotecas y herramientas adicionales	Baja, más sencillo y fácil de aprender
Escalabilidad	Muy alta, adecuado para aplicaciones grandes y complejas	Alta, escalable con la adición de bibliotecas y herramientas	Alta, adecuado para proyectos medianos y grandes

Rendimiento	Alto, aunque puede ser más pesado en comparación	Muy alto, rápido gracias al Virtual DOM	Muy alto, rendimiento comparable a React
Comunidad y Soporte	Muy grande, respaldado por Google	Muy grande, respaldado por Facebook	Grande, con un rápido crecimiento en popularidad
Curva de Aprendizaje	Empinada, requiere tiempo para dominar todas las características	Moderada, más fácil de aprender si se conocen otras bibliotecas	Suave, fácil de aprender y empezar a usar
Documentación	Extensa y completa	Muy buena, con gran cantidad de tutoriales y recursos	Excelente, clara y concisa
Integración con otras Herramientas	Excelente, especialmente con herramientas de Google	Muy buena, altamente compatible con muchas herramientas	Buena, aunque menos robusta que Angular y React
Flexibilidad	Moderada, enfoque más estructurado	Alta, permite elegir y combinar herramientas libremente	Alta, ofrece una buena combinación de estructura y flexibilidad
Actualizaciones y Mantenimiento	Regulares y bien gestionadas por Google	Frecuentes y bien gestionadas por Facebook	Frecuentes y bien gestionadas por la comunidad
Soporte para Móviles	Excelente, con Angular Ionic	Muy bueno, con React Native	Bueno, con soluciones como NativeScript

Framework de desarrollo frontend elegido.

Luego de evaluar diferentes frameworks frontend, se decide utilizar Angular el desarrollo del módulo. Esta decisión se fundamenta en varias razones clave. Angular ofrece una arquitectura robusta que facilita el desarrollo de aplicaciones complejas y bien organizadas, siendo ideal para proyectos de gran envergadura. Su rendimiento es sólido y adecuado para aplicaciones intensivas; además de contar con una amplia comunidad de soporte, lo que garantiza acceso a numerosos recursos y asistencia. La documentación de Angular es detallada y clara, lo cual es esencial para obtener una curva de aprendizaje significativa. Además, cabe recalcar que esta tecnología ya se viene aplicando sobre el proyecto con anterioridad.

3.1.5 Análisis y selección del framework web backend.

Tabla 6. Cuadro comparativo de frameworks backend.

Aspecto	ASP.NET Core	Express.js (Node.js)	Django (Python)	Ruby on Rails
Lenguaje(s) Soportado(s)	C#	JavaScript	Python	Ruby
Rendimiento	Alto	Variable	Bueno	Bueno
Escalabilidad	Alta	Buena	Buena	Buena
Comunidad	Grande	Grande	Grande	Grande
Soporte Empresarial	Muy Bueno	Variable	Bueno	Bueno
Herramientas Integradas	Visual Studio, .NET CLI	NPM, Yarn	Django Admin	Rails Console, Active Record
Documentación	Extensa	Buena	Buena	Buena
Flexibilidad	Alta	Alta	Alta	Alta
Mantenimiento	Activo	Activo	Activo	Activo
Integración con Azure	Muy Bueno	Variable	Bueno	Bueno
Curva de Aprendizaje	Moderada	Moderada	Moderada	Moderada

Framework de desarrollo backend elegido.

Luego de evaluar diferentes frameworks backend, la elección será de .NET como framework backend ya que representa una decisión sólida y confiable para el desarrollo de aplicaciones. Su versatilidad para trabajar con múltiples lenguajes de programación, como C#, brinda flexibilidad y potencial para adaptarse a diversas necesidades y preferencias de los desarrolladores.

Una de las fortalezas más destacadas de .NET es su sólido rendimiento y escalabilidad, lo que lo convierte en una opción ideal para aplicaciones que requieren manejo de carga intensiva y crecimiento a gran escala. Además, el soporte empresarial de .NET es otro factor para considerar, con un ecosistema bien establecido y herramientas de desarrollo como Visual Studio que ofrecen un entorno productivo. Finalmente, cabe recalcar que esta tecnología ya se viene utilizando en el sistema.

3.2 Desarrollo de la propuesta

3.2.1. Recopilación de requerimientos

En esta etapa se recopilieron todos los requerimientos que fueron dados por el cliente para dar inicio al proyecto.

3.2.1.1. Definición de roles

Tres personas colaboraron en el proyecto, los detalles se encuentran en la siguiente tabla.

Tabla 7. Definición de roles.

Responsable	Rol	Función
Daniel Montero	Desarrollador	Responsable del desarrollo del módulo de acuerdo con los requerimientos obtenidos.
Ing. Fernando Ibarra	Tutor	Encargado de verificar y dar seguimiento a la realización del módulo.
Ing. Vanessa Esparza	Cliente	Encargada del sistema “aTiendo” dentro de

		Plasticaucho Industrial S.A.
--	--	------------------------------

3.2.1.2 Requerimientos Técnicos

Los requerimientos técnicos se establecieron en base a las herramientas que provee la empresa Plasticaucho Industrial S.A. basándose en el nivel tecnológico que se encuentra actualmente.

Tabla 8. Requerimientos técnicos.

Tecnología	Versión	Detalle
Angular	14.2	Framework frontend.
ASP .Net	4.7.2	Framework backend.
phpMyAdmin	5.2.1	Gestor de MySQL.
Apache JMeter	5.6.3	Pruebas de rendimiento.

3.2.1.3 Listado de Requerimientos

Los requerimientos relacionados al módulo de facturación electrónica se recogieron con el cliente a través de entrevistas en cada incremento, además, estos requerimientos se fueron adaptando a las distintas etapas de comunicación en cada incremento. Por otra parte, las encuestas realizadas entre los usuarios de “aTiendo” de la organización revelaron puntos clave para el desarrollo del proyecto.

Durante la recopilación del listado de requerimientos, se realizó interacciones con:

- **Jefe administración de ventas:** Responsable del proyecto “aTiendo” dentro de la empresa Plasticaucho Industrial S.A.
- **Asistente de contabilidad:** Responsable de contabilidad en el área de administración de ventas.

La tabla de requerimientos consta de los siguientes campos:

- **ID:** Código del requerimiento.
- **Descripción:** Necesidad u origen del requerimiento.
- **Resultado:** Objetivo del requerimiento.
- **Criterio de aceptación:** Resultado esperado del requerimiento.

Tabla 9. Lista de requerimientos.

ID	DESCRIPCIÓN	RESULTADO	CRITERIOS DE ACEPTACIÓN
R_001	Necesito tener un módulo de ventas en el dashboard.	Con la finalidad de realizar el proceso de venta y facturación desde el módulo.	El usuario debe tener en el dashboard un módulo de ventas.
R_002	Necesito poder listar mis almacenes.	Con la finalidad de verificar los almacenes de los cuales voy a vender.	El usuario debe tener una vista general de cada almacén.
R_003	Necesito un informe con filtros de las ventas realizadas en el almacén.	Con la finalidad de visualizar detalladamente las ventas del almacén.	El usuario debe tener una vista de las ventas realizadas con filtros de búsqueda.
R_004	Necesito generar una nueva venta.	Con la finalidad de iniciar el proceso de venta.	El usuario debe tener un módulo para generar una nueva venta.
R_005	Necesito añadir los datos del cliente	Con la finalidad de que la factura tenga los datos necesarios para que sea válida.	El usuario deber tener los campos para añadir los datos del cliente final.
R_006	Necesito poder visualizar y editar los datos del cliente final.	Con la finalidad de que el usuario tenga la información actualizada en pantalla del cliente.	El usuario debe tener campos en donde puede ver la información del cliente para poder editar.

ID	DESCRIPCIÓN	RESULTADO	CRITERIOS DE ACEPTACIÓN
R_007	Necesito añadir la cantidad y los productos que el cliente solicite de mi almacén.	Con la finalidad de que todos los productos vendidos se vayan llenando en la factura.	El usuario debe tener una vista para añadir los productos con stock mayor a uno que se tenga del almacén.
R_008	Necesito poder ver, editar y eliminar el detalle de los productos añadidos.	Con la finalidad de que la información de los productos añadidos sea actualizada.	El usuario debe tener una vista del detalle con los campos a editar.
R_009	Necesito agregar descuentos porcentuales o nominales al detalle de productos.	Con la finalidad de que el cliente pueda obtener cierto descuento a productos.	El usuario debe tener un campo para añadir descuentos.
R_010	Necesito visualizar los importes totales de la factura.	Con la finalidad de saber cuánto será el total de la factura.	El usuario debe tener la información detallada de los importes.
R_011	Necesito poder añadir la forma de pago de la factura.	Con la finalidad de poder añadir distintas formas de pago a la factura.	El usuario debe tener una vista para poder añadir las formas de pago.
R_012	Necesito enviar la factura para su aprobación por el SRI.	Con la finalidad de que la factura sea aprobada bajo el reglamento del Servicio de Rentas Internas.	El usuario debe tener una vista en donde puede visualizar la factura aprobada.
R_013	Necesito generar un PDF de la factura.	Con la finalidad de que la factura sea visualizada.	El usuario debe tener un botón para descargar la factura en PDF.
R_014	Necesito enviar la factura por correo electrónico.	Con la finalidad de que el cliente final tenga la factura en su correo.	El usuario debe tener un botón para enviar la factura por correo.

ID	DESCRIPCIÓN	RESULTADO	CRITERIOS DE ACEPTACIÓN
R_015	Necesito ingresar los datos de mi firma electrónica.	Con la finalidad de que las facturas realizadas sean firmadas para su validación.	El usuario debe tener un módulo para agregar los datos de su firma electrónica.
R_016	Necesito visualizar y exportar un informe de cuadre de caja con filtros	Con la finalidad de que el cliente del sistema pueda cuadrar sus ventas en base a filtros de fecha y almacén.	El usuario debe poder filtrar por fechas y almacen el monto recaudado de sus ventas.
R_017	Necesito visualizar y exportar los movimientos (ventas) con filtros	Con la finalidad de que el cliente del sistema pueda validar los productos que ha vendido en base a filtro de fechas y almacén.	El usuario debe poder filtrar por fechas y almacen las ventas y productos vendidos.

3.2.1.4 Plan de Entrega

Una vez recopilados los requisitos, definido los resultados y establecidos los criterios de aceptación, se procedió a elaborar el siguiente plan de entrega:

Tabla 10. Planificación de la entrega de cada incremento.

Incremento	Código	Actividades	Tiempo (horas)
Incremento 1	R_001	Creación de la vista “Ventas”.	2
	R_002	Listar almacenes para la venta.	2
	R_003	Ventas realizadas en el almacén.	2
	R_004	Iniciar nueva venta.	1
	R_005	Ingresar datos del consumidor final.	3
	R_006	Ver y editar datos del consumidor final.	5
	R_007	Añadir cantidad de productos al detalle de la venta.	8

	R_008	Ver, editar y eliminar productos del detalle de la venta.	6
Incremento 2	R_009	Aplicar descuentos porcentuales o nominales a la venta.	2
	R_010	Visualizar importes totales de la venta.	2
	R_011	Añadir forma de pago de la factura.	4
	R_012	Enviar la factura al SRI.	8
	R_013	Generar PDF de la factura.	8
	R_014	Enviar factura por correo electrónico al consumidor final.	6
Incremento 3	R_015	Configurar datos de firma electrónica.	8
	R_016	Módulo de cuadre de caja.	8
	R_017	Módulo de movimientos.	8

3.2.1.5 Arquitectura del sistema

Actualmente el sistema maneja una arquitectura n-tier. Esta arquitectura se compone por las siguientes tres capas:

Capa de presentación (Frontend): Es la interfaz de usuario, donde los clientes interactúan con el sistema. Esta capa se encargará de la presentación de la información y de la captura de datos del usuario, todo esto desarrollando mediante la tecnología del framework Angular.

Capa de lógica de negocio (Backend): Es el núcleo del sistema donde se gestionan las reglas de negocio y el procesamiento de datos. Esta capa recibe las peticiones del frontend, realiza el procesamiento necesario y devuelve las respuestas adecuadas. Esta se desarrollará utilizando C# como lenguaje de programación y .Net como framework.

Capa de datos (Base de datos): Es la capa donde se almacenan y gestionan los datos de la aplicación. Aquí se encuentra el sistema de gestión de bases de datos (DBMS)

MySQL. El backend se comunica con la base de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) según las necesidades de la aplicación.

Además, existe una conexión entre el backend y los servicios web del SRI lo cual permite el envío de la factura en formato xml para su posterior autorización por parte de la entidad del Servicio de Rentas Internas.

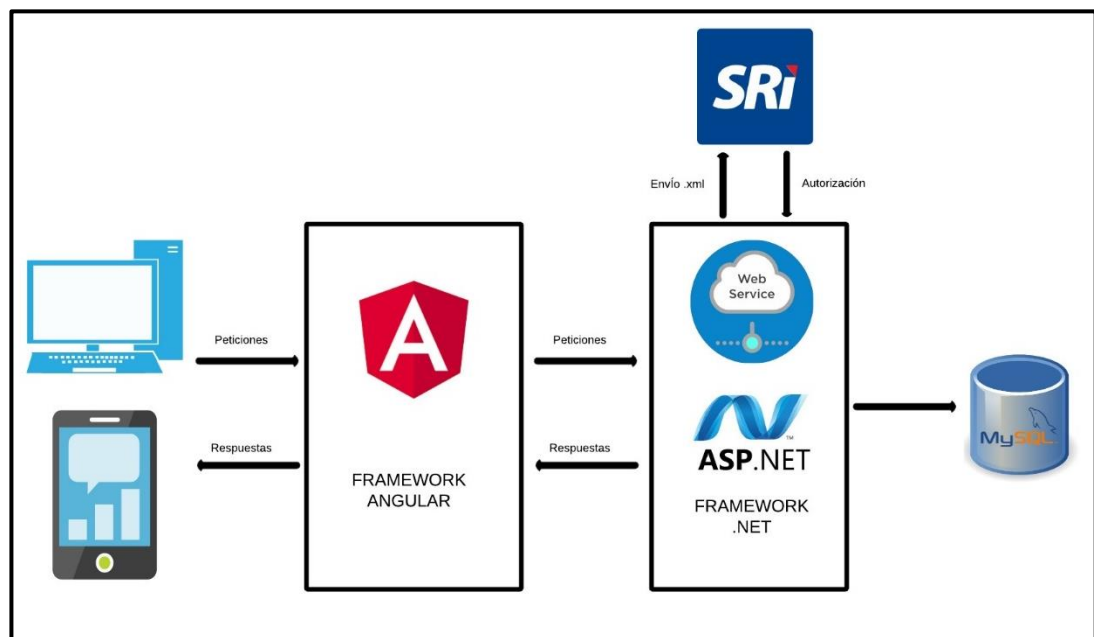


Figura 16. Arquitectura del sistema y módulo de facturación electrónica.

Diseño de la base de datos

Para garantizar un correcto acceso a la información se diseñaron nuevas tablas al sistema ya desarrollado. Estas cuentan con los campos necesarios en cada una para la gestión del módulo de facturación electrónica planteado, lo que brinda organización e integridad de datos en el sistema.

Estas nuevas implementaciones a la base de datos se realizaron en phpMyAdmin y MySQL Workbench para realizar distintos cambios y consultas directas; una vez terminada la implementación se realizó una técnica de ingeniería inversa para obtener el gráfico de las nuevas tablas, tal como se observa en la figura 17.

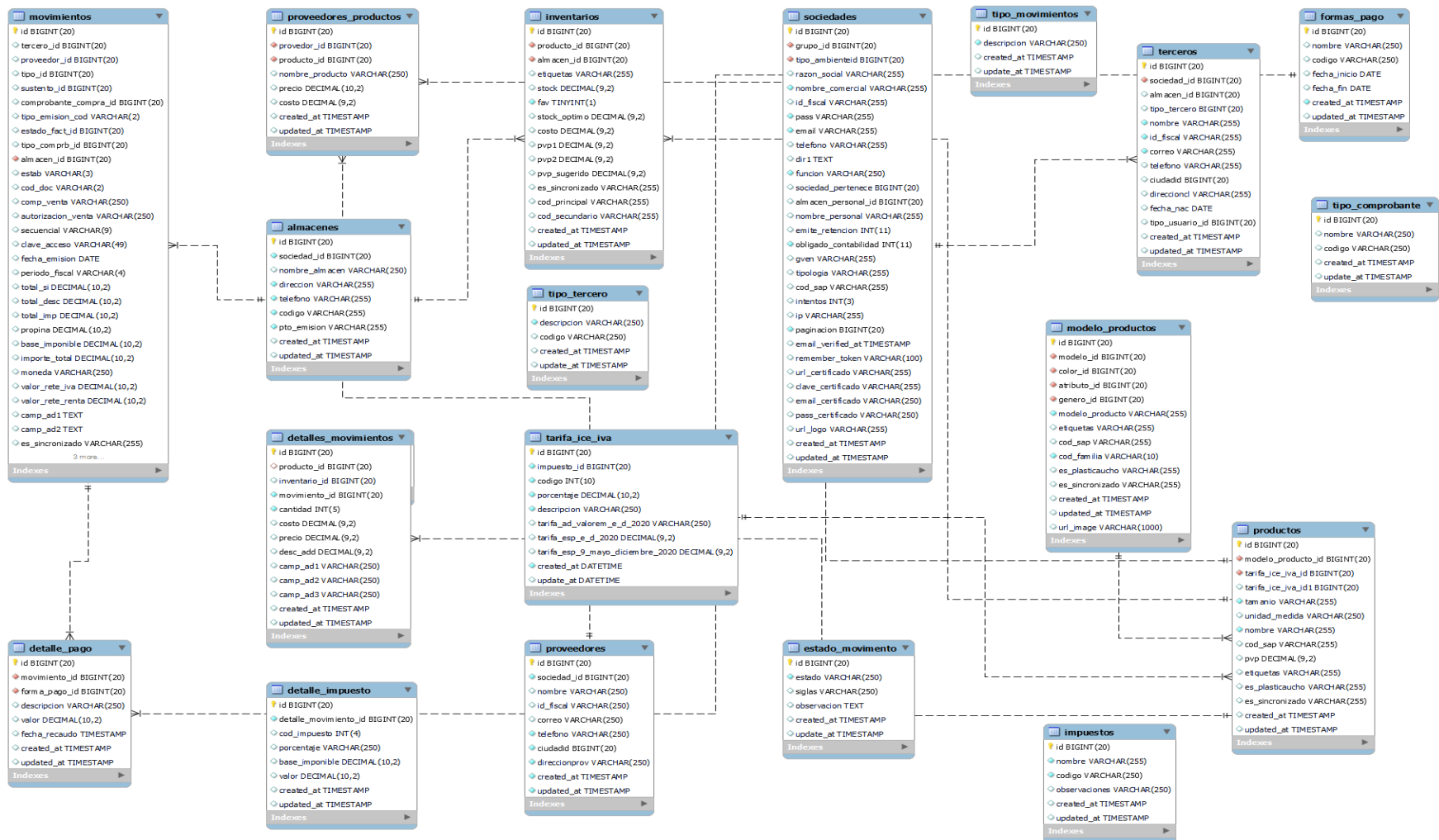


Figura 17. Tablas en la base de datos.

3.2.2 Aplicación del primer incremento (Incremento 1)

3.2.2.1 Comunicación

En esta fase, se realizó un análisis de las necesidades y requerimientos del cliente. La empresa Plasticaucho Industrial S.A. manifestó la necesidad de un módulo de facturación electrónica que se integre de manera adecuada con el sistema “aTiendo”. Se llevaron a cabo reuniones con el cliente para definir los requerimientos y alcance aplicables en este incremento:

1. **Módulo de Ventas en el Dashboard (R_001):** El cliente necesita un módulo en el dashboard para realizar el proceso de venta y facturación.
2. **Listado de Almacenes (R_002):** Es esencial poder listar los almacenes disponibles para verificar desde dónde se realizará la venta.
3. **Informe de Ventas con Filtros (R_003):** Se requiere un informe detallado de las ventas realizadas en el almacén, con opciones de filtros de búsqueda.
4. **Generación de Nueva Venta (R_004):** Se necesita un módulo que permita iniciar el proceso de venta.
5. **Añadir Datos del Cliente (R_005):** Es crucial que la factura contenga los datos necesarios del cliente final para que sea válida.
6. **Visualización y Edición de Datos del Cliente (R_006):** Los usuarios deben poder ver y editar la información del cliente.
7. **Añadir Cantidad y Productos (R_007):** Se requiere una vista donde se pueda añadir productos con stock mayor a uno desde el almacén.
8. **Ver, Editar y Eliminar Detalle de Productos Añadidos (R_008):** Los usuarios deben poder actualizar la información de los productos añadidos.

3.2.2.2 Planeación

La planeación de este incremento se centró en establecer un cronograma detallado para asegurar el cumplimiento de los requerimientos establecidos anteriormente. Se definieron los siguientes pasos clave:

- **Análisis de Requerimientos:** Se realizó la revisión detallada de cada requerimiento para entender su alcance y especificaciones.
- **Definición del Cronograma:** Creación de un cronograma con hitos y horas límite para cada tarea, incluyendo revisiones y pruebas.

Luego de haber revisado de manera detallada los requerimientos planteados para el incremento, se decidió planear el siguiente cronograma:

Tabla 11. Cronograma Incremento 1.

CRONOGRAMA INCREMENTO 1											
Código	Actividades	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	DÍA 6	DÍA 7	DÍA 8	DÍA 9	DÍA 10
R_001	Creación de la vista “Ventas”.										
R_002	Listar almacenes para la venta.										
R_003	Ventas realizadas en el almacén.										
R_004	Iniciar nueva venta.										
R_005	Ingresar datos del consumidor final.										
R_006	Ver y editar datos del consumidor final.										
R_007	Añadir cantidad de productos al detalle de la venta.										
R_008	Ver, editar y eliminar productos del detalle de la venta.										

3.2.2.3 Modelado

En esta fase, se desarrollaron los modelos y prototipos necesarios para la implementación del primer incremento. Los componentes modelados son los siguientes:

- **Prototipos de Interfaces:** Creación de maquetas visuales de las interfaces de usuario para los módulos de ventas, listados de almacenes, informes de ventas, y edición de datos del cliente y productos. Además, se colocará como modelado el ajuste de login del sistema.

a. Login del sistema

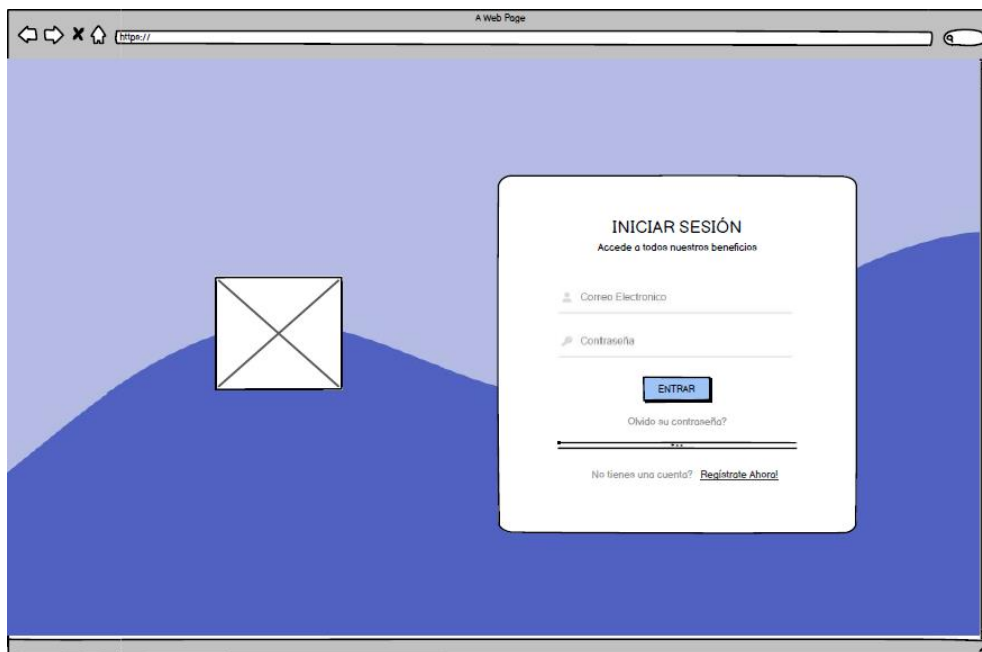


Figura 18. Wireframe del login del sistema

En la figura 17 se observa la manera de realizar el login en el sistema, con un correo y contraseña para su posterior autenticación.

b. Módulo de Ventas en el Dashboard

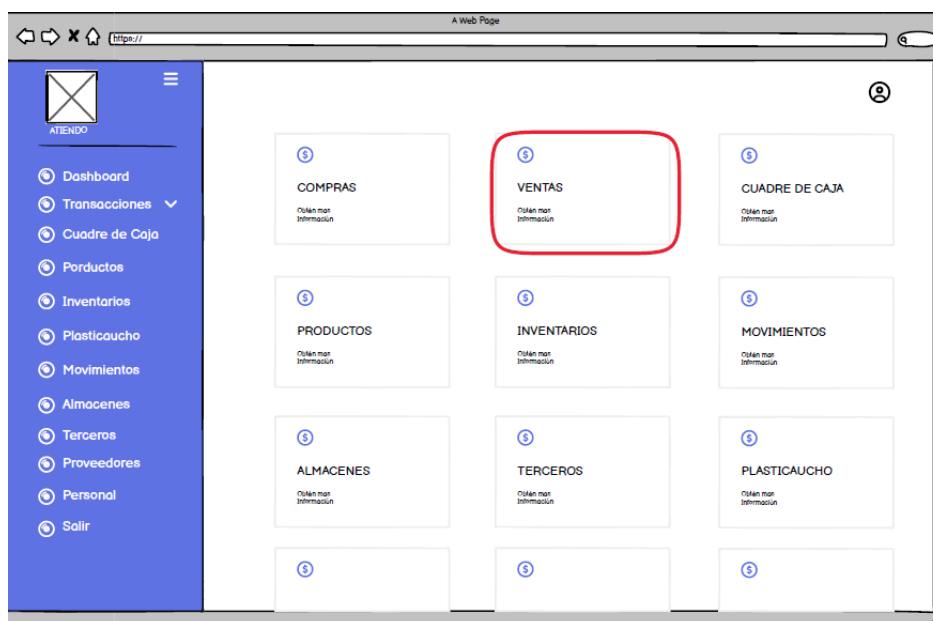


Figura 19. Wireframe del dashboard.

c. Listado de Almacenes



d. Informe de Ventas con Filtros

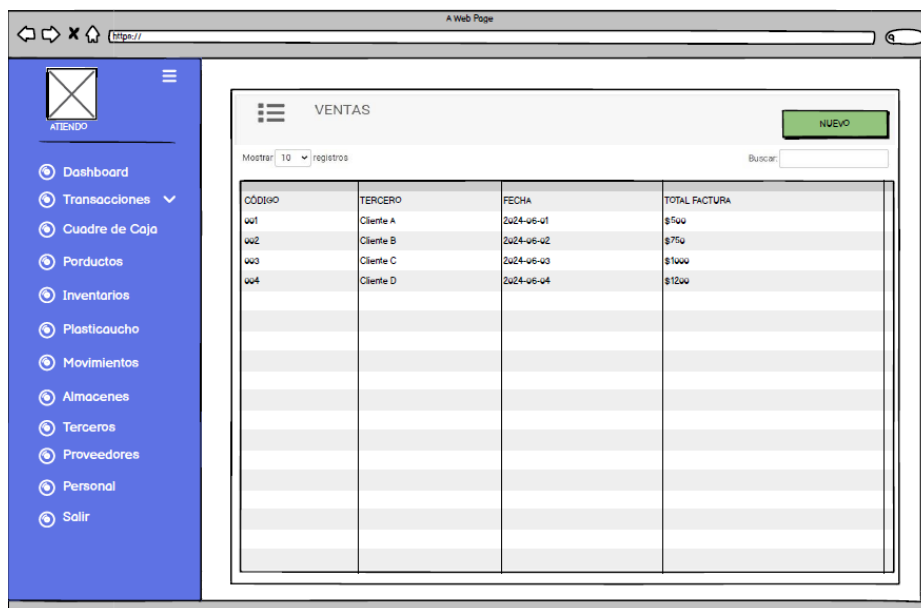


Figura 21. Wireframe del informe de ventas en el almacén.

En la figura 20 se observa el modo en el que debe ver el informe de ventas del almacén seleccionado con anterioridad.

e. Generación de Nueva Venta

FACTURACIÓN

Cedula/Ruc: Nombre: Correo: Ciudad:

Direccion: Telefono:

CÓDIGO	TERCERO	FECHA	TOTAL FACTURA

SUBTOTAL:

SUBTOTAL IVA 15%:

SUBTOTAL IVA 0%:

DESCUENTO:

SUBTOTAL:

IVA 15%:

ICE:

TOTAL:

Figura 22. Wireframe generación de nueva venta.

Como se observa en la figura 21, se creará una vista completa para la generación de una nueva venta, la misma tendrá el detalle de la venta, datos del cliente e importes.

f. Añadir Datos del Cliente

The wireframe shows a web application interface for a 'FACTURACIÓN' (Billing) section. On the left is a blue sidebar menu with the title 'ATENIDO' and a list of navigation items: Dashboard, Transacciones (with a dropdown arrow), Cuadre de Caja, Productos, Inventarios, Plástico/caucho, Movimientos, Almacenes, Terceros, Proveedores, Personal, and Salir. The main content area is titled 'FACTURACIÓN' and contains a form for adding client data. This form is highlighted with a red border and includes fields for 'Cedula/Ruc', 'Nombre', 'Correo', 'Ciudad', 'Dirección', and 'Teléfono', along with an 'ACTUALIZAR' button. Below this form is a green 'AÑADIR' button. Underneath the 'AÑADIR' button is a table with four columns: 'CÓDIGO', 'TERCERO', 'FECHA', and 'TOTAL FACTURA'. To the right of the table are several input fields for financial calculations: 'SUBTOTAL', 'SUBTOTAL IVA 15%', 'SUBTOTAL IVA 4%', 'DESCUENTO', 'SUBTOTAL', 'IVA 15%', 'ICE', and 'TOTAL'. A blue 'CONTINUAR' button is located at the bottom center of the main content area.

Figura 23. Wireframe de la vista de adición de datos del cliente.

Como se observa en la figura 22, existirá un popup en donde se ingresan los datos del cliente, sea consumidor final o con datos de facturación.

g. Visualización y Edición de Datos del Cliente

The wireframe shows a web application interface for managing client data and generating invoices. It features a sidebar menu on the left, a main content area with a client data form, a table for invoices, and a summary section on the right.

Sidebar Menu:

- Dashboard
- Transacciones
- Cuadre de Caja
- Productos
- Inventarios
- Plasticacho
- Movimientos
- Almacenes
- Terceros
- Proveedores
- Personal
- Salir

Client Data Form (FACTURACIÓN):

Fields for client information:

- Cédula/Ruc
- Nombre
- Correo
- Ciudad
- Dirección
- Teléfono

Buttons: **ACTUALIZAR** (Update), **AGREGAR** (Add).

Table:

CÓDIGO	TERCERO	FECHA	TOTAL FACTURA

Summary Section:

- SUBTOTAL:
- SUBTOTAL IVA 15%:
- SUBTOTAL IVA 0%:
- DESCUENTO:
- SUBTOTAL:
- IVA 15%:
- ICE:
- TOTAL:

Buttons: **CONTINUAR** (Continue).

Figura 24. Wireframe de visualización y edición de datos del cliente

De acuerdo con la figura 23, se tendrán los datos del cliente en la pantalla de la vista de la generación de venta, esto con la finalidad de que se pueda ver y editar dichos datos. Además, para este prototipo hay un feedback del cliente, el cual sugiere se pueda controlar que si el monto total de la venta supera \$50 se debe pedir de manera obligatoria los datos del cliente, es decir, no puede ser consumidor final.

h. Añadir Cantidad y Productos

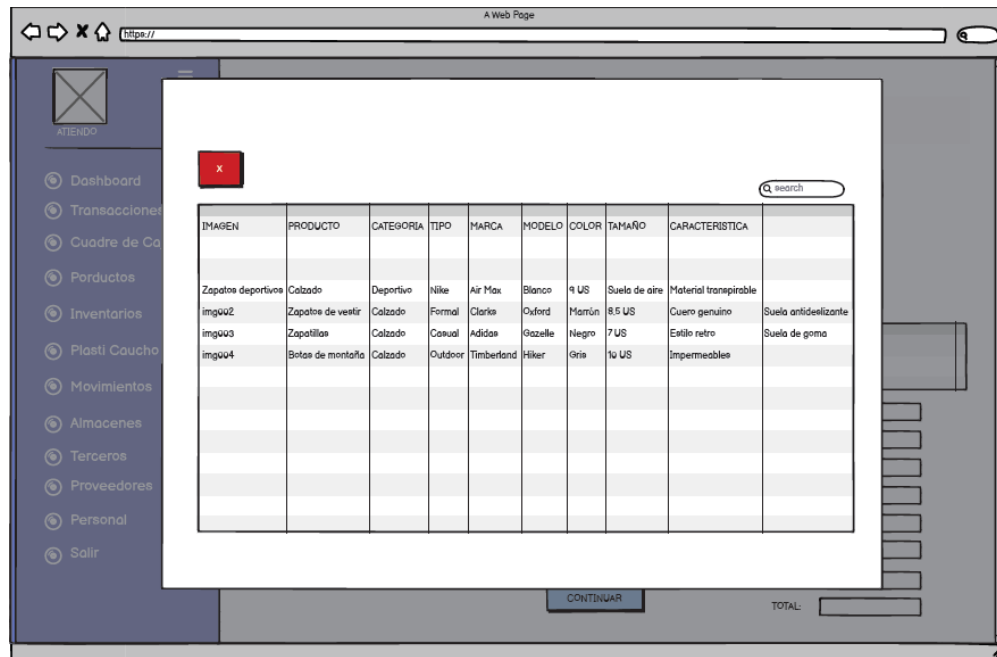


Figura 25. Wireframe de la adición de productos a la lista de venta.

Según la figura 24, el prototipo para la vista diseñada para añadir cantidad de productos a la lista de ventas se debe crear un popup en donde las cantidades se vayan añadiendo de manera dinámica al carrito.

i. Ver, Editar y Eliminar Detalle de Productos Añadidos

Figura 26. Wireframe de la vista, edición y eliminación de la lista de venta.

Como se observa en la figura 25, la vista, edición y eliminación de productos añadidos al carrito se encontrará en una tabla interna de la vista de generación de la venta. Esta contará con las funcionalidades necesarias.

3.2.2.4. Construcción

La fase de construcción correspondiente al primer incremento involucró el desarrollo e implementación del módulo de facturación electrónica, de acuerdo con los requerimientos habilitados para este primer incremento.

Estructura del módulo de ventas y facturación electrónica en Angular.

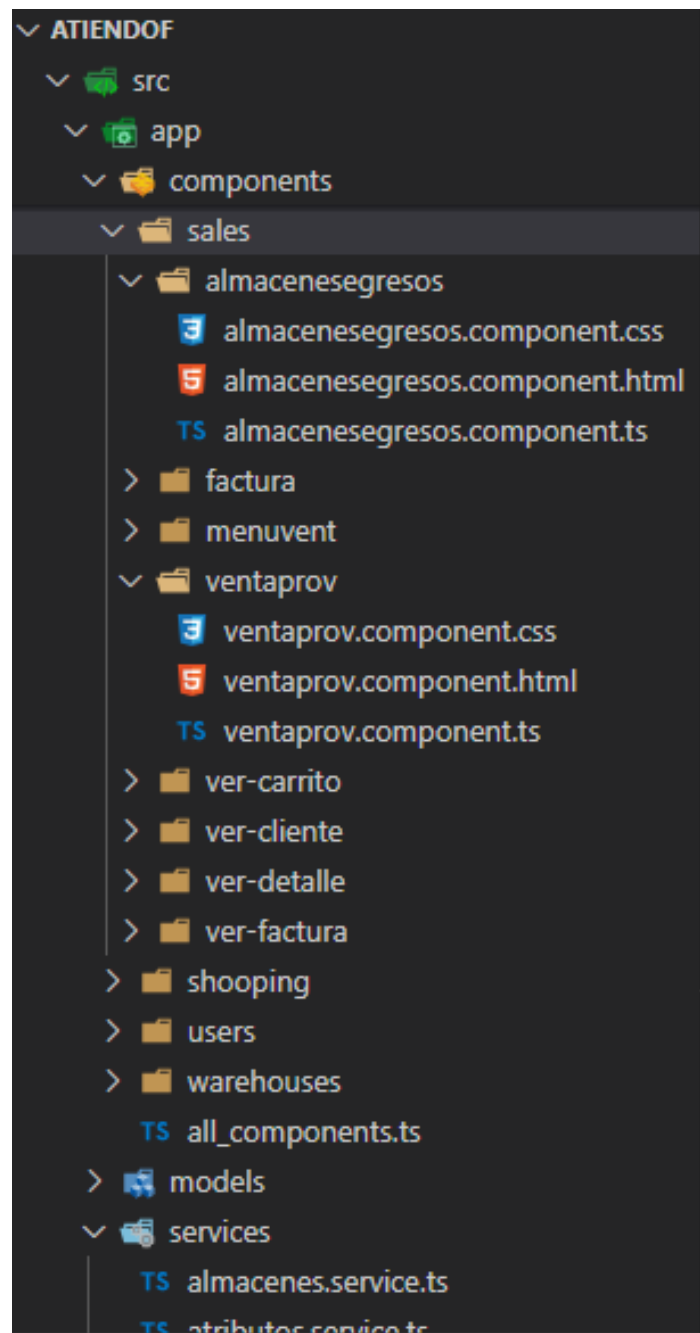


Figura 27. Estructura del módulo con servicios y modelos en Angular.

Estructura del módulo de ventas y facturación electrónica en C# .Net.

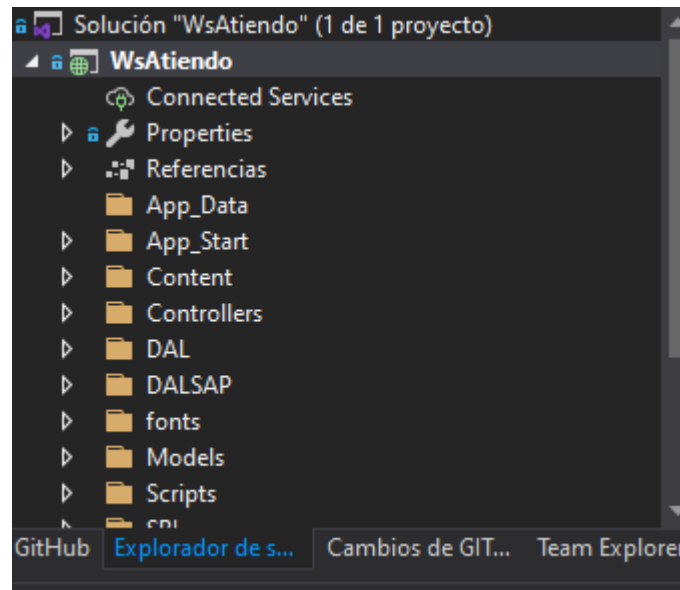


Figura 28. Estructura del módulo de facturación en la aplicación ASP. Net.

Como se observa en la figura 26, la estructura en Angular tendrá tres capas, es decir que existe la carpeta de modelos en donde se crearan los mismos necesarios para la correcta operabilidad del módulo. Además, estará la carpeta de servicios en donde se encontrarán los métodos para consumir los servicios web creados en el backend. Finalmente, se encuentra la carpeta de componentes, los cuales tienen la estructura de Angular, proporcionando directamente la vista y la lógica.

Por otro lado, en la figura 27, se observa la estructura en ASP. Net y tendrá igualmente 3 capas, es decir que hay una carpeta para los modelos o entidades, en donde se crearán los atributos necesarios, así como los métodos get y set. Además, hay otra carpeta llamada DAL o capa de acceso a datos la cual contiene las clases y métodos por los cuales se harán consultas a la base de datos, entre otros. Finalmente, se encuentra la carpeta de Controladores, la cual posee la manera en la que se crean los servicios web en la aplicación para su posterior uso en el frontend.

a. Ingreso al sistema.

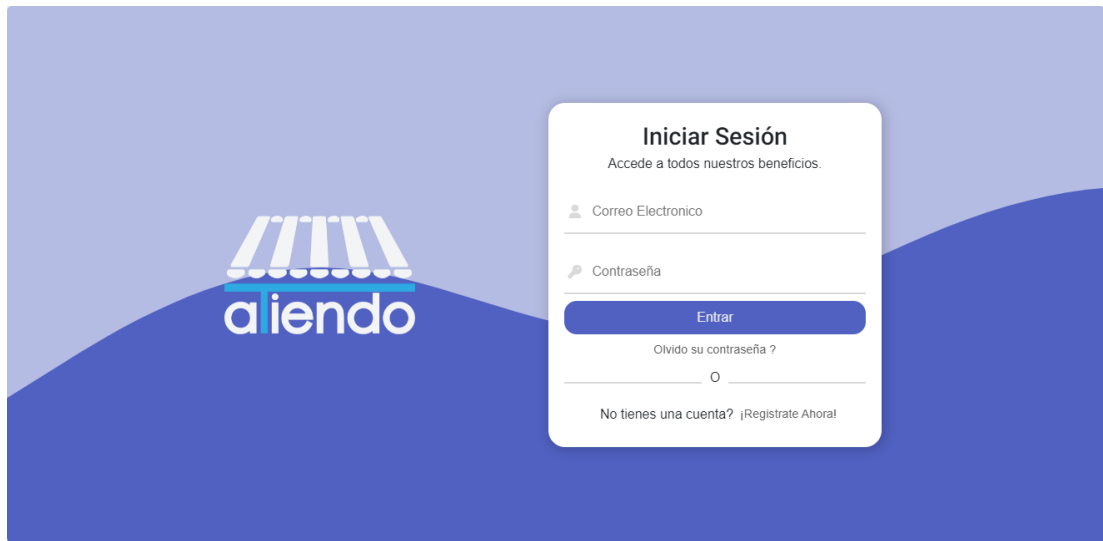


Figura 29. Ingreso al sistema "aTiendo".

```

73 this.httpService.obtenerUsuario(userEntity).subscribe(res => {
74
75     //Si la sociedad es undefined se notifica y se retorna al login
76     if (res.lstSociedades[0] == undefined) {
77         Swal.fire({
78             icon: 'error',
79             title: 'Credenciales Incorrectas.'
80         }).finally(() => {
81             this.router.navigate(['/login-nav']);
82         });
83     }
84     const idsociedad = res.lstSociedades[0].idSociedad;
85     const idsociedad2 = res.lstSociedades[0].sociedad_pertenece;
86     const idalmacenPertenece = res.lstSociedades[0].almacen_personal_id;
87     localStorage.setItem('sociedadid', idsociedad);
88     if (res.codigoError == "OK") {
89         var salt = CryptoJS.enc.Base64.parse("SXZhb1BNZWZ2ZWRldg==");
90         var iv = CryptoJS.enc.Hex.parse("69135769514182d0eded589ff874cacd");
91         var key564bits10000Iterations = CryptoJS.PBKDF2("Venus21!", salt, { keySize: 256 / 32 + 128 / 32, iterations: 10000, hasher: CryptoJS.algo.SHA512 });
92         const pass = this.categoryForm.value!.codigoSAP ?? "";
93         var encrypted = CryptoJS.AES.encrypt(pass, key564bits10000Iterations, {
94             iv: iv,
95             mode: CryptoJS.mode.CBC,
96             padding: CryptoJS.pad.Pkcs7
97         });
98         this.passwre = pass;
99         this.encPass = encrypted.toString();
100         const sociedadEntity: SociedadesEntity = {
101             idGrupo: '',
102             nombre_comercial: '',
103             id_fiscal: '',
104             email: this.categoryForm.value!.categoria ?? "",
105             funcion: '',
106             tipo_ambienteid: '',
107             telefono: '',
108             password: this.encPass,
109             idSociedad: '',
110             razon_social: ''
111         }

```

Figura 30. Método de encriptación de contraseña.

```

150 this.httpService.obtenerSociedadL(sociedadEntity).subscribe(res => {
151     if (res.codigoError == "OK") {
152         const rol = res.lstSociedades[0].funcion;
153         const passwa = this.passwre;
154         localStorage.setItem('passwa', passwa);
155         switch (rol) {
156             case "admin":
157                 Swal.fire({
158                     icon: 'success',
159                     title: 'Bienvenido Administrador!!!'
160                 }).finally(() => {
161                     this.router.navigate(['/navegation-adm']);
162                 })
163                 break;
164             case "client":
165                 Swal.fire({
166                     icon: 'success',
167                     title: 'Bienvenido!!!'
168                 }).finally(() => {
169                     this.router.navigate(['/navegation-cl']);
170                 })
171                 break;
172             case "personal":
173                 Swal.fire({
174                     icon: 'success',
175                     title: 'Bienvenido!!!'
176                 }).finally(() => {
177                     this.router.navigate(['/navegation-facturador']);
178                     localStorage.setItem('sociedadid', idsociedad2!);
179                     localStorage.setItem('almacenid', idalmacenPertenece!);
180                 })
181             }
182         }
183     }
184 }
185

```

Figura 31. Método de validación de credencial y rol.

En la figura 19 y 30, se puede validar la forma en la que se realiza la autenticación del usuario en el sistema. Esta cuenta con una implementación de seguridad ya que contiene un método de encriptación para no guardar las credenciales en texto plano en la base de datos.

```

12 path: 'navigation-cl', component: NavigationClComponent,
13 children: [
14   { path: "", component: DashboardClComponent, outlet: "contentClient" },
15   { path: "almacenes", component: AlmacenessociedadComponent, outlet: "contentClient" },
16   { path: "crearAlmacenes", component: AlmacenessociedadCreateComponent, outlet: "contentClient" },
17   { path: "editarAlmacenes", component: AlmacenessociedadEditComponent, outlet: "contentClient" },
18   { path: "inventarios", component: InventariosComponent, outlet: "contentClient" },
19   { path: "editarAlmacenes", component: AlmacenessociedadEditComponent, outlet: "contentClient" },
20   { path: "inventarios-pedido", component: InventariosPedidoComponent, outlet: "contentClient" },
21   { path: "inventarios-pedido-categoria", component: InventariosPedidoCategoriaComponent, outlet: "contentClient" },
22   { path: "inventarios-pedido-colores", component: InventariosPedidoColoresComponent, outlet: "contentClient" },
23   { path: "inventarios-pedido-lineas", component: InventariosPedidoLineasComponent, outlet: "contentClient" },
24   { path: "inventarios-pedido-modelos", component: InventariosPedidoModelosComponent, outlet: "contentClient" },
25   { path: "inventarios-almacen", component: InventariosAlmacenComponent, outlet: "contentClient" },
26   { path: "inventarios-productos", component: InventariosProductosComponent, outlet: "contentClient" },
27   { path: "inventarios-create", component: InventariosCreateComponent, outlet: "contentClient" },
28   { path: "inventarios-edit", component: InventariosEditComponent, outlet: "contentClient" },
29   { path: "pedido-sugeridos", component: PedidoSugeridoComponent, outlet: "contentClient" },
30   { path: "usuario", component: UsuarioComponent, outlet: "contentClient" },
31   { path: "usuario-pass", component: UsuarioPassComponent, outlet: "contentClient" },
32   { path: "tercerosalmacen", component: TercerosalmacenComponent, outlet: "contentClient" },
33   { path: "tercerosusuarios-create", component: TercerosusuariosCreateComponent, outlet: "contentClient" },
34   { path: "tercerosusuarios-edit", component: TercerosusuariosEditComponent, outlet: "contentClient" },
35   { path: "tercerosusuarios", component: TercerosusuariosComponent, outlet: "contentClient" },
36   { path: "modeloproductos", component: ModeloproductosComponent, outlet: "contentClient" },
37   { path: "almacenesegresos", component: AlmacenesegresosComponent, outlet: "contentClient" },
38   { path: "almaceningresos", component: AlmaceningresosComponent, outlet: "contentClient" },
39   { path: "vistapedidos", component: VistapedidosComponent, outlet: "contentClient" },
40   { path: "pedidoprov", component: PedidoprovComponent, outlet: "contentClient" },
41   { path: "vistamarcas", component: VistamarcasComponent, outlet: "contentClient" },
42   { path: "portafolios", component: PortafoliosComponent, outlet: "contentClient" },
43   { path: "portafolios-comprar", component: PortafoliosComprarComponent, outlet: "contentClient" },
44   { path: "carrito", component: CarritoComponent, outlet: "contentClient" },
45   { path: "portafolios-editar", component: PortafoliosEditarComponent, outlet: "contentClient" },
46   { path: "vistafactura", component: VistafacturaComponent, outlet: "contentClient" },
47   { path: "vistamarcas-dev", component: VistamarcasDevComponent, outlet: "contentClient" },
48   { path: "portafolios-dev", component: PortafoliosDevComponent, outlet: "contentClient" },
49   { path: "portafolios-comprar-dev", component: PortafoliosComprarDevComponent, outlet: "contentClient" },
50   { path: "carrito-dev", component: CarritoDevComponent, outlet: "contentClient" },
51   { path: "estadopedido", component: EstadopedidoComponent, outlet: "contentClient" },
52   { path: "vistaventas", component: VistaventasComponent, outlet: "contentClient" },
53   { path: "ventaprov", component: VentaprovComponent, outlet: "contentClient" },
54   { path: "pedidovent", component: PedidoventComponent, outlet: "contentClient" },
55   { path: "menuvent", component: MenuventComponent, outlet: "contentClient" },
56   { path: "menucompr", component: MenucomprComponent, outlet: "contentClient" },
57   { path: "ver-factura", component: VerFacturaComponent, outlet: "contentClient" },
58   { path: "almaceneshooping", component: AlmaceneshoopingComponent, outlet: "contentClient" },
59   { path: "compraprov", component: CompraprovComponent, outlet: "contentClient" },
60   { path: "movimientos", component: MovimientosComponent, outlet: "contentClient" },
61   { path: "proveedores", component: ProveedoresComponent, outlet: "contentClient" },
62   { path: "proveedores-create", component: ProveedoresCreateComponent, outlet: "contentClient" },
63   { path: "proveedores-edit", component: ProveedoresEditComponent, outlet: "contentClient" },
64   { path: "cuadre-caja", component: CashBalancingComponent, outlet: "contentClient" },

```

Figura 32. Módulo de rutas en Angular.

Angular posee un archivo llamado “Routing Module”, el cual se puede observar en la figura 31. Este módulo permite la creación de rutas y acceso a diferentes vistas; una de estas es la validación del inicio de sesión del usuario, llevándolo a una ruta en función del rol y aumentando este nivel de seguridad para controlar las vistas de cada usuario.

b. Dashboard del sistema.

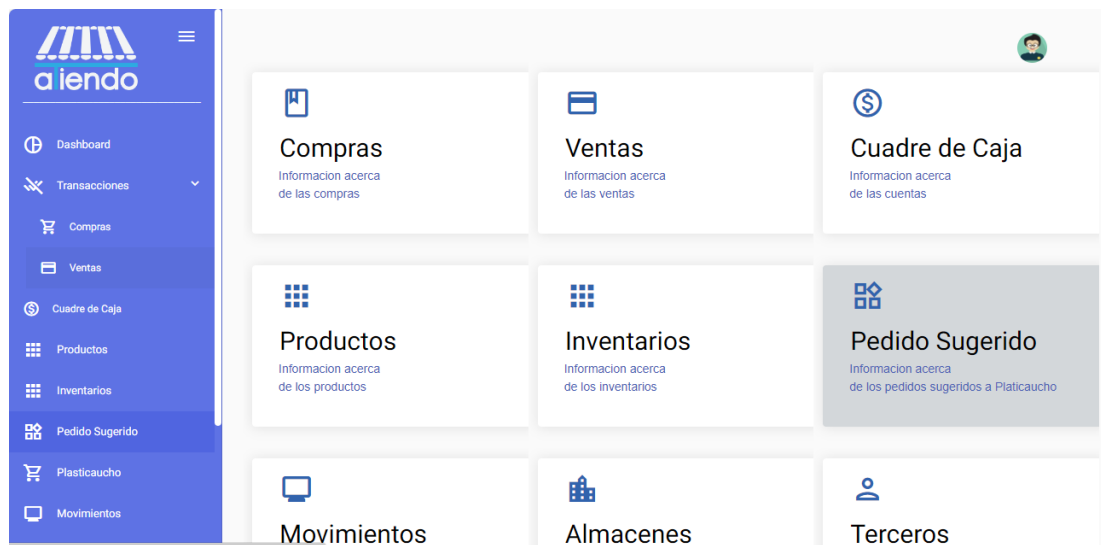


Figura 33. Dashboard del sistema "aTiendo" con módulo ventas.

```

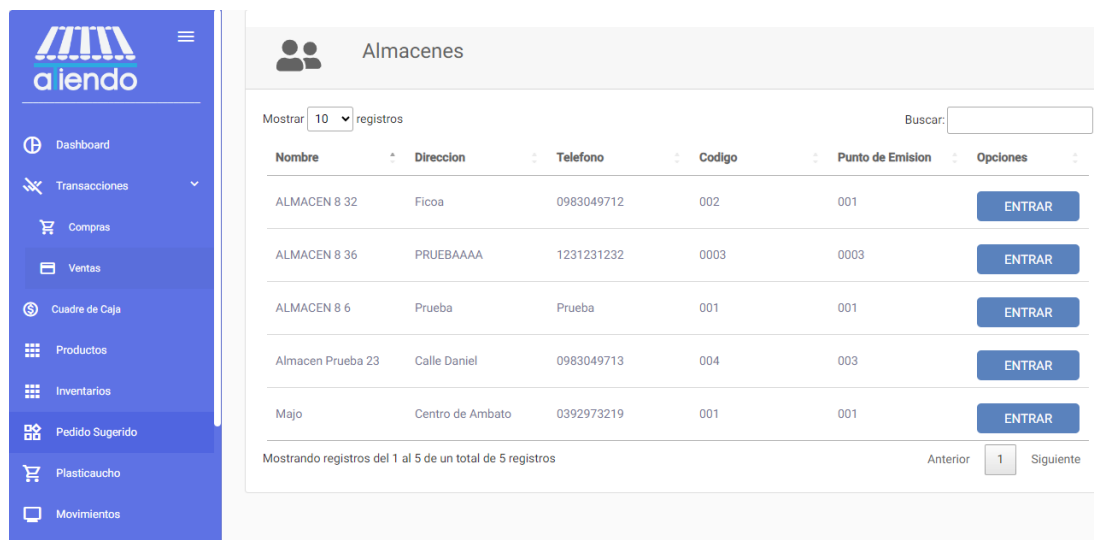
17 export class DashboardComponent {
18
19   getGridCols(): number {
20     return this.breakpointObserver.isMatched('(min-width: 1200px)') ? 3 : // Si la pantalla es de al menos 1200px de ancho, mostrar 3 columnas
21           this.breakpointObserver.isMatched('(min-width: 768px)') ? 2 : // Si la pantalla es de al menos 900px de ancho, mostrar 2 columnas
22           1; // De lo contrario, mostrar solo 1 columna
23   }
24
25   getRowHeight(): string {
26     return this.breakpointObserver.isMatched('(min-width: 1200px)') ? '225px' : // Si la pantalla es de al menos 1200px de ancho, usar una altura de fila de 225px
27           this.breakpointObserver.isMatched('(min-width: 768px)') ? '225px' : // Si la pantalla es de al menos 900px de ancho, usar una altura de fila de 200px
28           '200px'; // De lo contrario, usar una altura de fila de 150px
29   }
30
31   /** Based on the screen size, switch from standard to one column per row */
32   cards = this.breakpointObserver.observe(Breakpoints.Handset).pipe(
33     map(({ matches }) => {
34       if (matches) {
35         return [
36           { title: 'Compras', cols: 1, rows: 1, name: 'las compras', figure: 'book', dir: 'navigation-cl/(contentClient:almacenes:shopping)' },
37           { title: 'Ventas', cols: 1, rows: 1, name: 'las ventas', figure: 'payment', dir: 'navigation-cl/(contentClient:almacenes:ventas)' },
38           { title: 'Cuadre de Caja', cols: 1, rows: 1, name: 'las cuentas', figure: 'monetization_on', dir: 'navigation-cl/(contentClient:cuadre-caja)' },
39           { title: 'Productos', cols: 1, rows: 1, name: 'los productos', figure: 'apps', dir: 'navigation-cl/(contentClient:inventarios-productos)' },
40           { title: 'Inventarios', cols: 1, rows: 1, name: 'los inventarios', figure: 'apps', dir: 'navigation-cl/(contentClient:inventarios-almacen)' },
41           { title: 'Pedido Sugerido', cols: 1, rows: 1, name: 'los pedidos sugeridos a Plasticaucho', figure: 'widgets', dir: 'navigation-cl/(contentClient:pedido-sugeridos)' },
42           { title: 'Movimientos', cols: 1, rows: 1, name: 'los movimientos', figure: 'monitor', dir: 'navigation-cl/(contentClient:movimientos)' },
43           { title: 'Almacenes', cols: 1, rows: 1, name: 'los almacenes', figure: 'location_city', dir: 'navigation-cl/(contentClient:almacenes)' },
44           { title: 'Terceros', cols: 1, rows: 1, name: 'los terceros', figure: 'person', dir: 'navigation-cl/(contentClient:terceros:almacen)' },
45           { title: 'Pedidos Plasticaucho', cols: 1, rows: 1, name: 'plasticaucho', figure: 'book', dir: 'navigation-cl/(contentClient:almacenes:ventas)' },
46           { title: 'Proveedores', cols: 1, rows: 1, name: 'los proveedores', figure: 'supervisor_account', dir: 'navigation-cl/(contentClient:proveedores)' },
47           { title: 'Personal', cols: 1, rows: 1, name: 'el personal', figure: 'supervisor_account', dir: 'navigation-cl/(contentClient:personal)' },
48         ];
49       }
50     });
51   return [
52     { title: 'Compras', cols: 1, rows: 1, name: 'las compras', figure: 'book', dir: 'navigation-cl/(contentClient:almacenes:shopping)' },
53     { title: 'Ventas', cols: 1, rows: 1, name: 'las ventas', figure: 'payment', dir: 'navigation-cl/(contentClient:almacenes:ventas)' },
54     { title: 'Cuadre de Caja', cols: 1, rows: 1, name: 'las cuentas', figure: 'monetization_on', dir: 'navigation-cl/(contentClient:cuadre-caja)' },
55     { title: 'Productos', cols: 1, rows: 1, name: 'los productos', figure: 'apps', dir: 'navigation-cl/(contentClient:inventarios-productos)' },
56     { title: 'Inventarios', cols: 1, rows: 1, name: 'los inventarios', figure: 'apps', dir: 'navigation-cl/(contentClient:inventarios-almacen)' },
57     { title: 'Pedido Sugerido', cols: 1, rows: 1, name: 'los pedidos sugeridos a Plasticaucho', figure: 'widgets', dir: 'navigation-cl/(contentClient:pedido-sugeridos)' },
58     { title: 'Movimientos', cols: 1, rows: 1, name: 'los movimientos', figure: 'monitor', dir: 'navigation-cl/(contentClient:movimientos)' },
59     { title: 'Almacenes', cols: 1, rows: 1, name: 'los almacenes', figure: 'location_city', dir: 'navigation-cl/(contentClient:almacenes)' },
60     { title: 'Terceros', cols: 1, rows: 1, name: 'los terceros', figure: 'person', dir: 'navigation-cl/(contentClient:terceros:almacen)' },
61     { title: 'Plasticaucho', cols: 1, rows: 1, name: 'plasticaucho', figure: 'book', dir: 'navigation-cl/(contentClient:almacenes:ventas)' },
62     { title: 'Proveedores', cols: 1, rows: 1, name: 'los proveedores', figure: 'supervisor_account', dir: 'navigation-cl/(contentClient:proveedores)' },
63     { title: 'Personal', cols: 1, rows: 1, name: 'el personal', figure: 'supervisor_account', dir: 'navigation-cl/(contentClient:personal)' },
64   ];
65   });
66   });
67   });

```

Figura 34. Lógica del dashboard del cliente.

En la figura 33 se observa la manera en la que se crean las tarjetas que se visualizan en el dashboard del cliente. Además, existe el control para que en pantallas más pequeñas el dashboard se vaya ajustando; con la finalidad de entregar una vista responsiva.

c. Listado de almacenes.



Nombre	Direccion	Telefono	Codigo	Punto de Emision	Opciones
ALMACEN 8 32	Ficoa	0983049712	002	001	ENTRAR
ALMACEN 8 36	PRUEBAAAA	1231231232	0003	0003	ENTRAR
ALMACEN 8 6	Prueba	Prueba	001	001	ENTRAR
Almacen Prueba 23	Calle Daniel	0983049713	004	003	ENTRAR
Majo	Centro de Ambato	0392973219	001	001	ENTRAR

Mostrando registros del 1 al 5 de un total de 5 registros

Figura 35. Lista de almacenes del usuario.

```

87     this.httpService.obtenerAlmacenesSociedad(almacen).subscribe(res => {
88         if (res.codigoError != "OK") {
89             Swal.fire({
90                 icon: 'error',
91                 title: 'Ha ocurrido un error.',
92                 text: res.descripcionError,
93                 showConfirmButton: false,
94                 // timer: 3000
95             });
96         } else {
97             this.lstAlmacenes = res.lstAlmacenes;
98             this.dtTrigger.next('');
99         }
100     });
101 }
102
103
104
105 abrirVista(almacen: AlmacenesEntity) {
106
107     localStorage.setItem('almacenid', almacen.idAlmacen)
108     this.router.navigate(['/navegacion-cl', { outlets: { 'contentClient': ['ventapro] } }]);
109
110 }

```

Figura 36. Método para listar almacenes.

En la figura 35 se puede observar el método mediante el cual se obtienen los almacenes del usuario utilizando un servicio web con el cual se consultan los almacenes en la

base de datos. Además, se puede observar el método para abrir la vista del almacén al que se ingrese.

d. Informe de ventas

The screenshot displays the 'a iendo' application's 'VENTAS' (Sales) module. On the left is a blue sidebar menu with icons and labels for various system functions: Dashboard, Transacciones, Cuadre de Caja, Productos, Inventarios, Pedido Sugerido, Plasticaucho, Movimientos, Almacenes, and Terceros. The main content area is titled 'VENTAS' and features a 'NUEVO' (New) button in the top right. Below the title, there's a 'Volver' (Back) button. The interface includes filter controls for 'Fecha Desde' (Date From) and 'Fecha Hasta' (Date To), with a 'Reiniciar Filtros' (Reset Filters) button. A 'Mostrar' (Show) dropdown is set to '10' registros, and a search bar labeled 'Buscar:' is present. The data is presented in a table with the following columns: CÓDIGO, TERCERO, FECHA, and TOTAL FACTURA. Each row also contains a green 'GENERAR' button. The table lists several sales entries, including those for Daniel Eduardo Montero Lopez and Edison Orlando Lopez.

CÓDIGO	TERCERO	FECHA	TOTAL FACTURA	
2043	Daniel Eduardo Montero Lopez	29/05/2024	10.93	GENERAR
2041	Daniel Eduardo Montero Lopez	29/05/2024	10.93	GENERAR
2014	Edison Orlando Lopez	28/05/2024	51.75	GENERAR
1936	Edison Orlando Lopez	22/05/2024	57.50	GENERAR
1925	Edison Orlando Lopez	21/05/2024	60.00	GENERAR

Figura 37. Vista de informe de ventas con filtros.

```

80 ngOnInit(): void {
81   this.dtOptions = {
82     language: {
83       url: "https://cdn.datatables.net/plug-ins/1.10.15/i18n/Spanish.json"
84     },
85     paging: true,
86     search: false,
87     searching: true,
88     ordering: false,
89     info: true,
90     responsive: true
91   }
92   const newMovimiento: MovimientosEntity = {
93     id: '',
94     tipo_id: '',
95     tipo_emision_cod: '',
96     estado_fact_id: '',
97     tipo_comprob_id: '',
98     almacen_id: localStorage.getItem("almacenid"),
99     cod_doc: '',
100     secuencial: ''
101   }
102   Swal.fire({
103     title: 'CARGANDO...',
104     html: 'Se están cargando los pedidos.',
105     timer: 30000,
106     didOpen: () => {
107       Swal.showLoading();
108       this.httpService.obtenerMovimientosAlmacen(newMovimiento).subscribe(res => {
109         if (res.codigoError != "OK") {
110           Swal.fire({
111             icon: 'error',
112             title: 'Ha ocurrido un error.',
113             text: res.descripcionError,
114             showConfirmButton: false,
115             // timer: 3000
116           });
117         } else {
118           this.lstMovimientos = res.lstMovimientos;
119           this.dtTrigger.next('');
120           Swal.close();
121         }
122       });
123     },
124     }).then((result) => {
125       /* Read more about handling dismissals below */
126       if (result.dismiss === Swal.DismissReason.timer) {
127         console.log('I was closed by the timer');
128       }
129     });
130     this.fechaActual = new Date().toISOString().split('T')[0];
131   }
132   ngOnDestroy(): void {
133     this.dtTrigger.unsubscribe();
134   }
135 }

```

Figura 38. Método para listar las ventas por almacén.

```

190 filterByDate(): void {
191   const fechaDesdeControl = this.filtroForm.get('fechaDesde');
192   const fechaHastaControl = this.filtroForm.get('fechaHasta');
193   const fechaDesde = fechaDesdeControl?.value;
194   const fechaHasta = fechaHastaControl?.value;
195   const almacen = localStorage.getItem('almacenid');
196   this.httpService.obtenerMovimientosVentf(almacen, fechaDesde, fechaHasta).subscribe(res => {
197     console.log(res)
198     if (res.codigoError != "OK") {
199       this.filtroForm.get('almacen')?.enable();
200       Swal.fire({
201         icon: 'error',
202         title: 'No se pudo obtener movimientos.',
203         text: res.descripcionError,
204         showConfirmButton: false,
205       });
206     } else {
207       this.lstMovimientos = res.lstMovimientos;
208       this.datatableElement.dtInstance.then((dtInstance: DataTables.Api) => {
209         // Destruye la tabla existente y elimina los datos
210         dtInstance.destroy();
211
212         // Renderiza la tabla con los nuevos datos
213         this.dtTrigger.next('');
214
215         // Opcional: Reinicia la página a la primera página
216         dtInstance.page('first').draw('page');
217       });
218     }
219   });
220 }

```

Figura 39. Método para filtrar por fechas.

Como se observa en la figura 37 y 38, se utiliza un método para listar las ventas en función del almacén al que se haya ingresado anteriormente; además existe un método para filtrar por fechas dichas ventas. Por otra parte, existe un campo para buscar en la tabla por cualquier criterio de columnas de la misma, este campo se puede observar en la figura 36

e. Generación de nueva venta e ingreso de datos del cliente.

Figura 40. Generación de nueva venta.

Figura 41. Popup de ingreso de datos del cliente.

```

213     const tercerodatos: TercerosEntity = {
214       id: '',
215       almacen_id: localStorage.getItem('almacenid')!,
216       sociedad_id: localStorage.getItem('sociedadid')!,
217       tipotercero_id: this.lstTipoTerceros2[0].idTipo_tercero ?? 0,
218       tipousuario_id: '1',
219       nombresociedad: '',
220       nombrealmacen: '',
221       nombretercero: this.TercerosForm.value!.tipo_tercero ?? '',
222       tipousuario: '',
223       nombre: (this.TercerosForm.value!.nombre ?? "").concat(" " + this.TercerosForm.value!.apellido ?? ""),
224       id_fiscal: this.TercerosForm.value!.id_fiscal ?? '',
225       direccion: this.TercerosForm.value!.direccion ?? '',
226       telefono: this.TercerosForm.value!.telefono ?? '',
227       correo: this.TercerosForm.value!.correo ?? '',
228       fecha_nac: "01-01-2000",
229       ciudad: '',
230       provincia: '',
231       ciudadid: this.TercerosForm.value!.ciudad_id ?? ''
232     }
233     localStorage.setItem('idfiscalCl', this.TercerosForm.value!.id_fiscal!);
234
235     console.log("Este es el id fiscal en ver cliente", this.TercerosForm.value!.id_fiscal!);
236     this.httpService.obtenerTerceroCedula(tercerodatos).subscribe(res1 => {
237       if (res1.codigoError == "NEXISTE") {
238         this.httpService.agregarTerceros(tercerodatos).pipe(finally(() => {
239           this.httpService.obtenerTerceroCedula(tercerodatos).subscribe(res3 => {
240             const newMovimiento: MovimientosEntity = {
241               id: localStorage.getItem('movimiento_id')!,
242               tipo_id: '',
243               tipo_emision_cod: '',
244               estado_fact_id: '',
245               tipo_comprb_id: '',
246               almacen_id: '',
247               cod_doc: '',
248               secuencial: '',
249               tercero_id: res3.lstTerceros[0].id!
250             }
251             localStorage.setItem('idClVenta', res3.lstTerceros[0].id!);
252             this.httpServiceMov.actualizarClientePedido(newMovimiento).subscribe(res2 => {
253               if (res2.codigoError != "OK") {
254                 Swal.fire({
255                   icon: 'info',
256                   title: 'Información',
257                   text: 'Ha ocurrido un error',
258                   showConfirmButton: true,
259                   // timer: 3000
260                 });
261               }
262             });
263           });
264         });
265       }
266     });

```

Figura 42. Método para agregar datos de facturación del cliente.

Como se puede observar en la figura 41, se creó un método para actualizar la información de los datos del cliente a la factura. En primer lugar se valida si el cliente ya realizó alguna compra en los almacenes para poder precargar los datos; caso contrario, se deben agregar los datos del cliente.

f. Visualización y edición de datos del cliente.

FACTURACIÓN

Volver

CLIENTE

ELEGIR

Cédula/RUC:

1850294875

Nombre:

Daniel Eduardo Montero López

Correo:

dannymonterodm123@gmail.com

Ciudad:

Ambato

Dirección:

ECU 911

Teléfono:

0993604424

Actualizar

Figura 43. Vista de edición de datos del cliente.

```

193     this.ciudadId = res.lstCiudades[0].idCiudadId;
194     const newTercero: TercerosEntity = {
195       almacen_id: '',
196       id: localStorage.getItem('idClVenta')!,
197       sociedad_id: '',
198       tipotercero_id: '',
199       tipousuario_id: '',
200       nombresociedad: '',
201       nombrealmacen: '',
202       nombretercero: this.nombre,
203       tipousuario: '',
204       nombre: this.nombre,
205       id_fiscal: '',
206       direccion: this.direccion,
207       telefono: this.telefono,
208       correo: this.correo,
209       fecha_nac: '',
210       ciudad: '',
211       provincia: '',
212       ciudadid: this.ciudadId,
213     };
214     console.log(newTercero);
215     this.httpServiceTer.actualizarCliente(newTercero).subscribe((res) => {
216       if (res.codigoError != 'OK') {
217         Swal.fire({
218           icon: 'info',
219           title: 'Información',
220           text: res.descripcionError,
221           showConfirmButton: true,
222           // timer: 3000
223         });
224       } else {
225         Swal.fire({
226           icon: 'success',
227           title: 'Actualizado',
228           text: 'Se han actualizado los datos',
229           showConfirmButton: true,
230           // timer: 3000
231         });

```

Figura 44. Método de actualización de datos del cliente.

Los datos del cliente se pueden visualizar en cualquier momento mientras se realiza la venta como se observa en la figura 42. Además, también se puede editar estos datos desde la misma vista con el método que se presenta en la figura 43.

g. Agregar productos a la lista de venta.

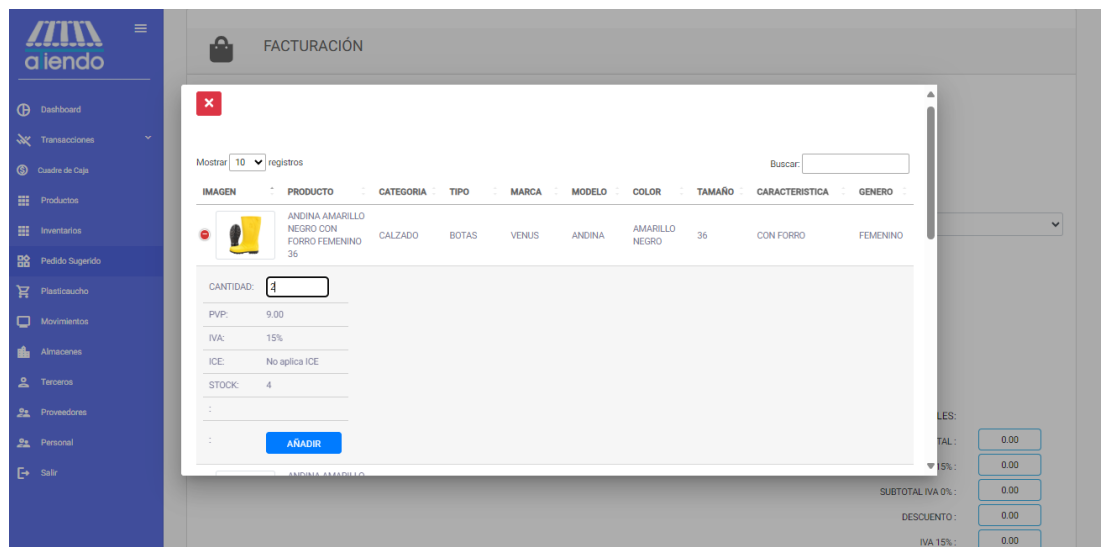


Figura 45. Popup para agregar productos al carrito o lista de venta.

```

231 crearDetalle(Inventario: InventarioEntity): void {
232   this.httpServiceInventarios.obtenerInventario$.pipe(take(1)).subscribe((res) => {
233     const newDetalle: DetallesMovimientoEntity = {
234       id: '',
235       producto_nombre: '',
236       inventario_id: this.codigo,
237       producto_id: res.producto_id,
238       movimiento_id: JSON.parse(localStorage.getItem('movimiento_id')) || '[]',
239       cantidad: inventario.cantidad!,
240       costo: this.costo,
241       precio: this.precio
242     }
243     this.httpServiceDetalle.agregarDetallePedido(newDetalle).pipe(finally(() => {
244       this.httpServiceDetalle.obtenerUltDetalleMovimiento(newDetalle).subscribe(res1 => {
245         if (res1.codigoError == 'OK') {
246           const newDetalleImp: DetalleImpuestosEntity = {
247             id: '',
248             detalle_movimiento_id: res1.lstDetalleMovimientos[0].id,
249             cod_impuesto: res1.lstDetalleMovimientos[0].cod_tarifa!,
250             porcentaje: res1.lstDetalleMovimientos[0].tarifa!,
251             base_imponible: '',
252             valor: res1.lstDetalleMovimientos[0].costo!,
253             created_at: '',
254             updated_at: ''
255           }
256           this.httpServiceDetalleImp.agregarDetalleImpuestos(newDetalleImp).subscribe(res2 => {
257             if (res2.codigoError != 'OK') {
258               Swal.fire({
259                 icon: 'error',
260                 title: 'Ha ocurrido un error.',
261                 text: res2.descripcionError,
262                 showConfirmButton: false
263               });
264             }
265           });
266         }
267       });
268     }
269   )}.subscribe(res => {
270     if (res.codigoError != 'OK') {
271       Swal.fire({
272         icon: 'error',
273         title: 'Ha ocurrido un error.',
274         text: 'No existe suficiente stock.2',
275         showConfirmButton: false
276       });
277     } else {
278       Swal.fire({
279         icon: 'success',
280         title: 'Se ha agregado al carrito',
281         text: 'Se ha guardado con éxito el producto',
282         showConfirmButton: true,
283         confirmButtonText: 'Ok',
284       }).then(() => {
285         this.cerrarDialog();
286       });
287       this.productoAgregado.emit(inventario);
288     }
289   });
290 }

```

Figura 46. Método para agregar productos al detalle de la venta.

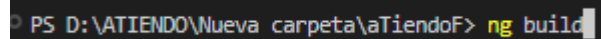
Como se observa en la figura 45, existe el método que permite agregar los productos al detalle de la venta. Cabe recalcar que mientras se va agregando productos, estos se van bajando del stock del inventario del usuario.

h. Vista, edición y eliminación del detalle de la venta.

En la figura 47 se puede validar los dos métodos existentes para modificar o eliminar el detalle de la venta, estos métodos también van validando el stock disponible del inventario del usuario y según se realicen las acciones de modificación o eliminación, este stock cambiará de forma dinámica.

3.2.2.5. Despliegue

En la fase de despliegue, se llevó a cabo la integración del módulo hasta el requerimiento dispuesto del primer incremento. Este despliegue se realizó a nivel de frontend como de backend.



```
PS D:\ATIENDO\Nueva carpeta\atiendoF> ng build
```

Figura 49. Comando para generar carpetas de despliegue en Angular.

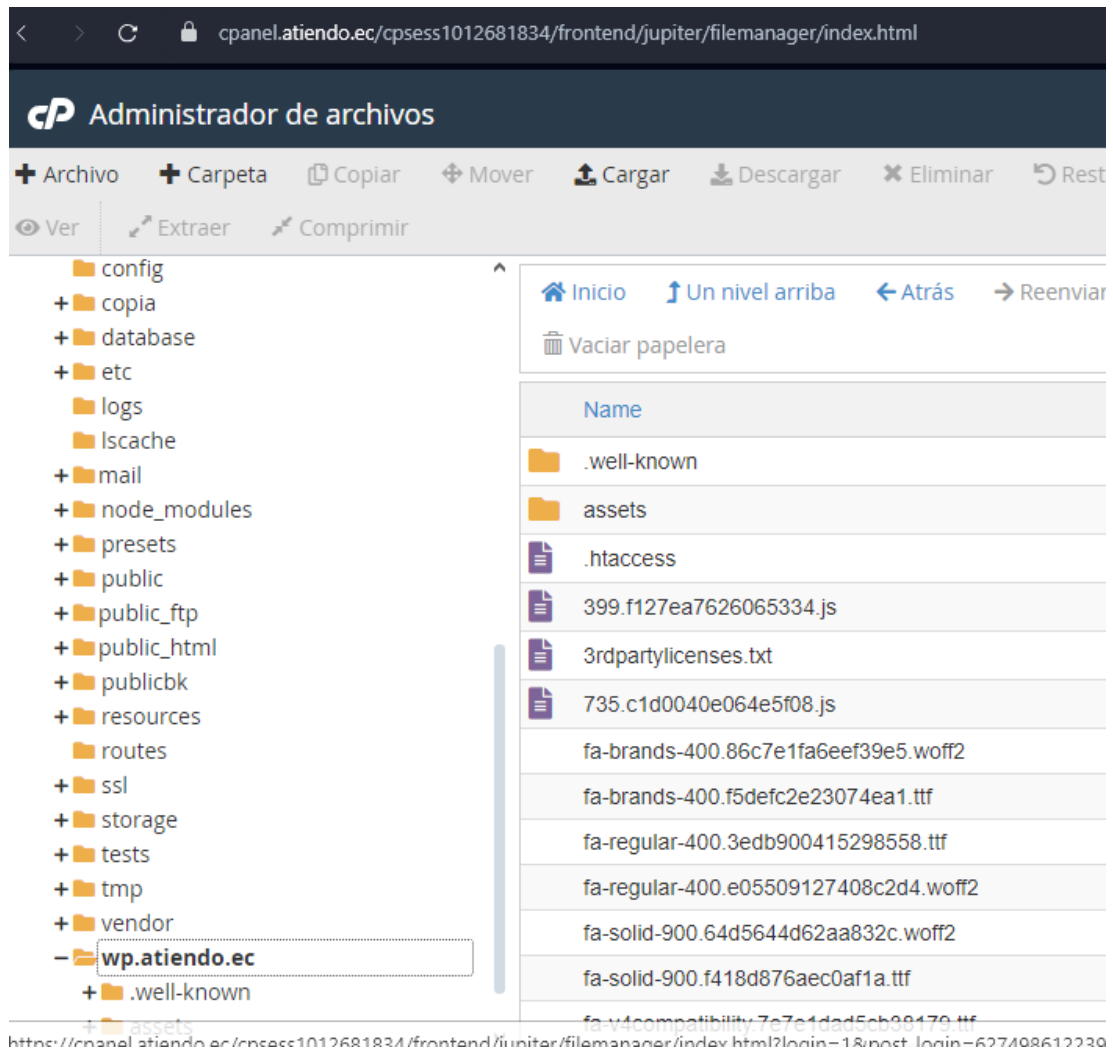


Figura 50. Despliegue de proyecto en cPanel.

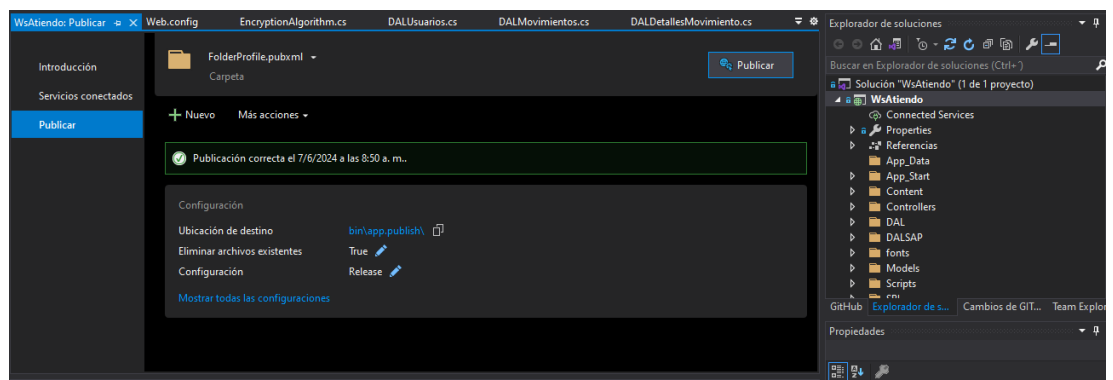


Figura 51. Generación de archivos de despliegue en ASP .Net

Como se observa en las figuras 49 y 50, se desplegaron los archivos necesarios para el correcto funcionamiento en un ambiente de producción mediante el cual se puede acceder al dominio y utilizar las funcionalidades creadas hasta el momento.

3.2.3 Aplicación del segundo incremento (Incremento 2)

3.2.3.1 Comunicación

En esta fase, se realizó un análisis de las necesidades y requerimientos del cliente. El cliente, mediante reuniones manifestó la necesidad de integrar la facturación en este incremento. Se llevaron a cabo reuniones con el cliente para definir los requerimientos y alcance aplicables en este incremento:

1. **Agregar Descuentos a Productos (R_009):** Es necesario que los usuarios puedan añadir descuentos porcentuales o nominales a los productos.
2. **Visualización de Importes Totales de la Factura (R_010):** Los usuarios deben poder ver el total de la factura para saber el monto que el cliente debe pagar.
3. **Añadir Forma de Pago (R_011):** Se requiere la posibilidad de añadir diferentes formas de pago a la factura.
4. **Envío de la Factura para Aprobación del SRI (R_012):** Es esencial que la factura pueda ser enviada al Servicio de Rentas Internas para su aprobación.
5. **Generación de PDF de la Factura (R_013):** Los usuarios necesitan la opción de generar un PDF de la factura para su visualización.
6. **Envío de la Factura por Correo Electrónico (R_014):** Se debe permitir a los usuarios enviar la factura al cliente final por correo electrónico.

3.2.3.2 Planeación

La planeación de este incremento contará con las mismas directrices aplicadas al anterior incremento, analizando los requerimientos y definiendo un cronograma.

Luego de haber revisado de manera detallada los requerimientos planteados para el incremento, se decidió planear el siguiente cronograma:

Tabla 12. Cronograma Incremento 2.

CRONOGRAMA INCREMENTO 2											
Código	Actividades	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	DÍA 6	DÍA 7	DÍA 8	DÍA 9	DÍA 10
R_009	Aplicar descuentos porcentuales o nominales a la venta.										
R_010	Visualizar importes totales de la venta.										
R_011	Añadir forma de pago de la factura.										
R_012	Enviar la factura al SRI.										
R_013	Generar PDF de la factura.										
R_014	Enviar factura por correo electrónico al consumidor final.										

3.2.2.3 Modelado

En esta fase, se desarrollaron los modelos y prototipos necesarios para la implementación del segundo incremento. Los componentes modelados serán prototipos al igual que en el primer incremento.

a. Aplicación de descuentos y visualización de importes.

The wireframe shows a web application interface for invoice management. On the left is a blue sidebar with a menu containing: ATENCIÓN, Dashboard, Transacciones, Cuadre de Caja, Productos, Inventarios, Plásticoaucho, Movimientos, Almacenes, Terceros, Proveedores, Personal, and Salir. The main content area is titled 'FACTURACIÓN' and contains a section for 'Factura Electrónica'. This section displays fields for 'Prueba1', 'RUC: 1803742079002', 'Dirección: Ficoa', and 'Teléfono: 0683049712'. Below this, there are fields for 'Número de factura:', 'Fecha de emisión: 6/13/2024', 'Cédula/RUC:', 'Cliente: Consumidor Final', 'Teléfono: 999999999', and 'Email: 999999999'. A table below these fields has columns for 'CÓDIGO', 'TERCERO', 'FECHA', and 'TOTAL FACTURA'. To the right of the table are input fields for 'SUBTOTAL:', 'SUBTOTAL IVA 15%', 'SUBTOTAL IVA 0%', 'DESCUENTO:', 'SUBTOTAL:', 'IVA 15%', 'ICE:', and 'TOTAL:'. At the bottom, there is a 'Forma de pago' dropdown, an 'ABONAR' button, a 'RESTANTE' input field, and a 'FINALIZAR VENTA' button.

Figura 52. Wireframe de la aplicación de descuentos y vista de importes.

En la figura 51 se observa la manera en la que se podrá aplicar descuentos nominales o porcentuales sobre el total de la factura. Además, se puede visualizar los importes como el subtotal y total, entre otros.

b. Añadir formas de pago de la factura.

Figura 53. Wireframe para agregar la forma de pago y monto de la factura.

Como se observa en la figura 52, se coloca en la vista anterior la manera en la que se agrega el monto y la forma de pago que decida el cliente sobre la factura. Cabe recalcar que estos montos deben tener la funcionalidad de ser editables.

c. Generación y envío de factura electrónica.

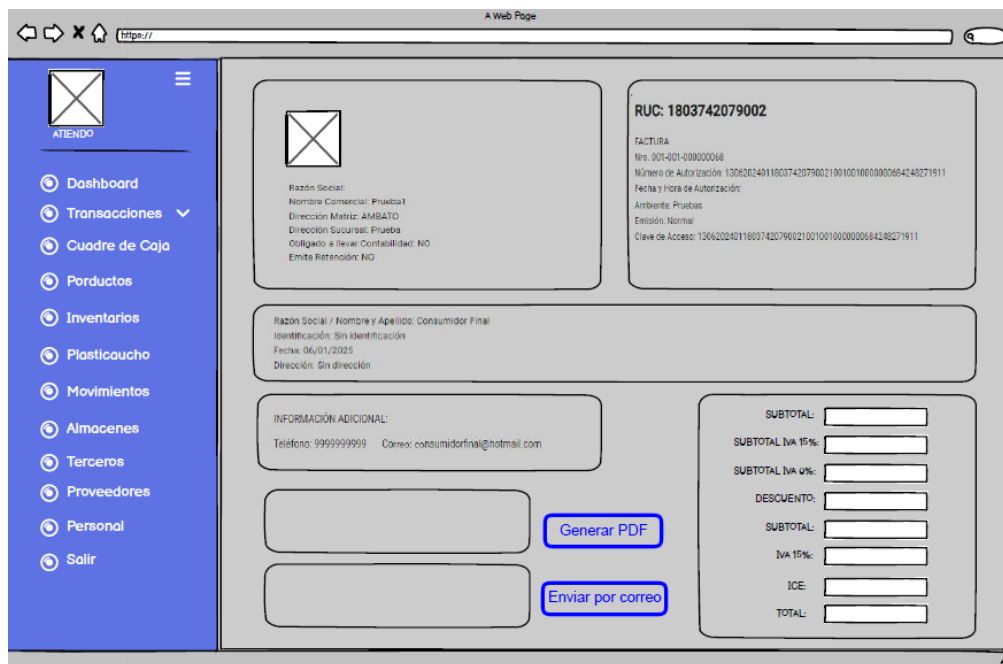


Figura 54. Wireframe de la vista para generar y enviar comprobante por correo.

En la figura 53 se puede validar la vista con la que el usuario podrá generar el archivo pdf y enviar el comprobante electrónico mediante correo. Además, esta vista será la que permita identificar si la factura electrónica fue recibida y autorizada por el SRI.

3.2.3.4. Construcción

La fase de construcción correspondiente al segundo incremento involucró la continuación del desarrollo e implementación del módulo de facturación electrónica, de acuerdo con los requerimientos habilitados para este segundo incremento.

Estructura del consumo del servicio web del SRI en Angular.

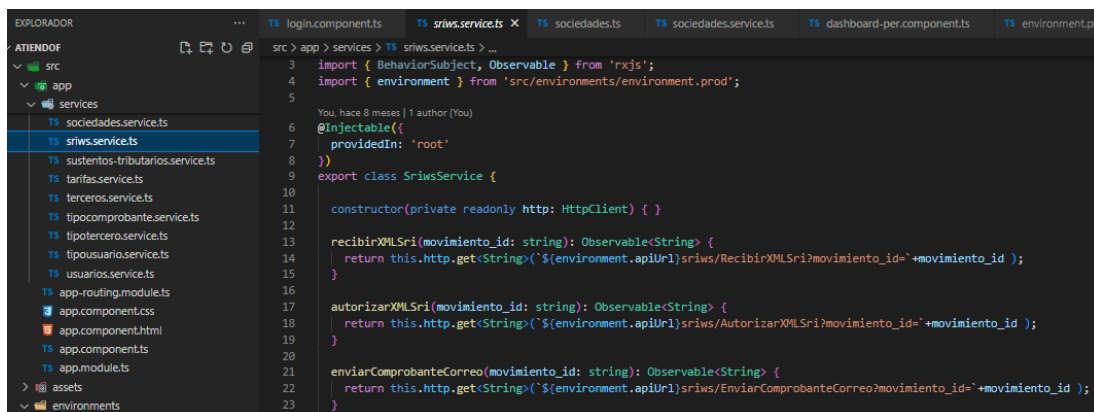


Figura 55. Archivo para consumir servicio web del SRI.

Estructura de archivos de consumo del servicio web del SRI en C# .Net.

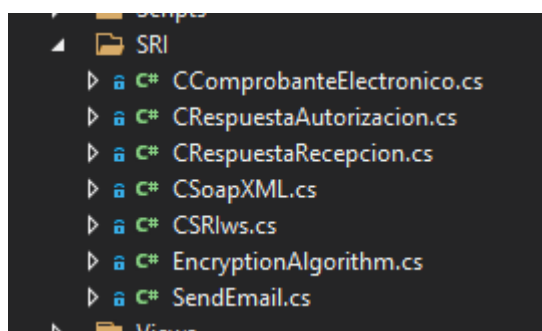


Figura 56. Estructura de consumo de servicio web de SRI en ASP .Net.

Como se observa en la figura 54, a la estructura presentada en Angular en el primer incremento, se le añadió el archivo de consumo de los servicios web que ofrece el SRI. Este archivo se comunica con el backend ya que este es el que se conecta directamente con los webservice ofrecidos por esta entidad con relación a la facturación electrónica.

Por otro lado, en la figura 55, se observa la estructura en ASP. Net de los archivos necesarios para consumir los servicios web que ofrece el SRI. Estos se han catalogado de acuerdo con su funcionalidad como lo son, la recepción y autorización del archivo de la factura electrónica.

a. Agregar descuentos nominales y porcentuales.



- Dashboard
- Transacciones
- Cuadre de Caja
- Productos
- Inventarios
- Pedido Sugerido
- Plasticaucho
- Movimientos
- Almacenes
- Terceros
- Proveedores
- Personal
- Salir

Cliente: Daniel Eduardo Montero López
 Telefono: 0993604424
 Email: dannymonterodm123@gmail.com

Producto	Cantidad	Precio Uni.	Precio	Descuento	Tamaño	Uni. Medida	Color
AND AM NG 36	3	9.00	27.00	0.00	36	par	AMARILLO NEGRO
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	5	9.00	45.00	0.00	40	par	AZUL BLANCO

SUBTTOTALES:

SUBTOTAL : 72.00

SUBTOTAL IVA 15% : 72.00

SUBTOTAL IVA 0% : 0.00

Dto DE PRODUCTOS : 0.00

Dto NOMINAL : 10

DESCUENTO % : 50

IVA 15% : 10.80

ICE : 0

TOTAL: 36.4

Figura 57. Vista para agregar descuentos nominales y porcentuales a la factura.

```

483 onInput2(event: any) {
484   let inputValue = event.target.value;
485   const filteredValue = inputValue.replace(/[^0-9.]/g, ''); // Filtra solo números y punto
486
487   if (/^\d*\.\d*$/.test(inputValue) && !filteredValue.startsWith('.')) {
488     this.calcularSumaTotal();
489   } else {
490     event.target.value = inputValue.length === 0 ? '0' : '';
491   }
492 }
493
494
495
496 validarDescuento(): number {
497   if (this.sumaTotal) {
498     const descuento = (((this.descuentoP) * (this.totalF)) / 100);
499
500     return descuento;
501   }
502   return 0;
503 }
504
505
506 calcularDescuento(): number {
507   const descuento = this.lstDetalleMovimientos
508     .filter((detalleMovimientos) => detalleMovimientos.desc_add)
509     .reduce((total, detalleMovimientos) => {
510       return total + parseFloat(detalleMovimientos.desc_add!.replace(',', '.'));
511     }, 0);
512
513   return descuento;
514 }
515
516 You, hace 1 segundo • Uncommitted changes
517
518 calcularTotalAbonado(): number {
519   const totalAb = this.lstDetallePagos
520     .filter((detallePagos) => detallePagos.valor)
521     .reduce((total, detallePagos) => {
522       console.log(total)
523       return total + parseFloat(detallePagos.valor.replace(',', '.'));
524     }, 0);
525
526   return totalAb;
527 }
528
529
530 calcularSubtotal(): number {
531   const subtotal = this.lstDetalleMovimientos
532     .filter((detalleMovimientos) => detalleMovimientos.tarifa)
533     .reduce((total, detalleMovimientos) => {
534       return total + parseFloat(detalleMovimientos.precio.replace(',', '.'));
535     }, 0);
536
537   return subtotal;
538 }
539
540

```

Figura 58. Método para aplicar descuentos y visualizar importes de la factura.

En la figura 57, se puede validar la forma en la que se realiza la aplicación de descuentos sobre la factura. Además se observa el calculo que se realiza sobre los totales de la factura para visualizar correctamente los importes.

b. Añadir formas de pago a la factura.

Dashboard

Transacciones

Cuadre de Caja

Productos

Inventarios

Pedido Sugerido

Plasticaucho

Movimientos

Almacenes

Terceros

Dto DE PRODUCTOS : 0.00

Dto NOMINAL : 10

DESCUENTO % : 50

IVA 15% : 10.80

ICE : 0

TOTAL: 36,4

Efectivo

Selección método de pago

Efectivo

Tarjeta de Débito

Tarjeta de Crédito

36.40

Abonar

Restante: 0.00

306

Efectivo

36.40

Finalizar venta

```

665 changePago(event: any): void {
666   const seleccionado = event.target.value;
667   this.isBotonHabilitado = seleccionado != "0";
668   const newPago: FormasPagoEntity = {
669     id: "",
670     nombre: event.target.value,
671     codigo: "",
672     fecha_inicio: "",
673     fecha_fin: "",
674     created_at: "",
675     updated_at: ""
676   }
677   this.httpServiceFormas.obtenerFormasPagoN(newPago).subscribe(res => {
678     if (res.codigoError != "OK") {
679     } else {
680       this.listFormasPago2 = res.listFormasPago;
681     }
682   })
683 }
684
685 //o, hace id nuevo + Servicio de detalle pago
686
687 abonar() {
688   const monto = document.getElementById('monto') as HTMLInputElement;
689   if (parseFloat(monto.value) <= this.sumaTotal.replace('.', '')) && parseFloat(monto.value) <= this.resto {
690     this.resto = (this.resto - parseFloat(monto.value)).toFixed(2);
691     this.actualizarColor();
692     const newDetallePago: DetallesPagoEntity = {
693       id: "",
694       movimiento_id: localStorage.getItem('movimiento_id')!,
695       forma_pago_id: this.listFormasPago2[0].id,
696       description: "PAGO",
697       valor: monto.value,
698       fecha_recaudo: "",
699       created_at: "",
700       updated_at: ""
701     }
702     this.httpServiceDetallePago.agregarDetallePago(newDetallePago).subscribe(res => {
703       if (res.codigoError == "OK") {
704         Swal.fire({
705           icon: 'success',
706           title: 'Abonado',
707           text: 'Has abonado' + ' ' + '$' + ' ' + monto.value,
708           showConfirmButton: false,
709         });
710         this.desabilitarIn = true;
711         const newMovimiento: MovimientosEntity = {
712           id: localStorage.getItem('movimiento_id')!,
713           tipo_id: "",
714           tipo_exicion_cod: "",
715           estado_fact_id: "",
716           tipo_comprob_id: "",
717           almacen_id: "",
718           cod_doc: "",
719           secuencial: ""
720         }
721         this.httpServiceDetallePago.obtenerDetallePagoN(newMovimiento).subscribe(res => {
722           if (res.codigoError != "OK") {
723           } else {
724             this.listDetallePagos = res.listDetallePagos;
725             this.datatablelement.dtInstance.then((dtInstance2: DataTable.Api) => {
726               // Destruye la tabla existente y elimina los datos
727               dtInstance2.destroy();
728             });
729             // Renderiza la tabla con los nuevos datos
730             this.dtTrigger2.next("");
731             // Opcional: Reinicia la página a la primera página
732             dtInstance2.page("first").draw("page");
733           });
734         }
735       }
736     }
737   }
738   } else {
739     console.log("ERROR")
740   }
741 }

```

Figura 60. Lógica para agregar las formas de pago y abonar.

En la figura 59 se observa la manera en la que se selecciona la forma de pago que requiera el cliente, y además, una vez seleccionada la manera en la que se va a pagar, se puede abonar el total en distintas formas de pago.

c. Enviar la factura al SRI.

```

29  export class VerFacturaComponent implements OnInit {
499  finalizarPedido() {
504      this.httpServiceMovimiento.obtenerUltimoSecuencial(localStorage.getItem('almacenid')!).subscribe(res1 => {
531          }).then((result) => {
535              finalize() => {
589                  cod_doc: '',
590                  secuencial: '',
591                  estab: '',
592                  importe_total: '',
593                  url_factura: 'ftp://calidad.atiende.ec/FacturasXML/factura_' + claveAccesoConDV + '.xml'
594              }
595              console.log(newMovCA)
596              this.httpServiceMovimiento.actualizarCLAVEACCESO(newMovCA).pipe(finalize() => {
597                  this.httpServiceMovimiento.crearXML(localStorage.getItem('movimiento_id')!).pipe(finalize() => {
598                      this.httpServiceSRI.recibirXMLSri(localStorage.getItem('movimiento_id')!).subscribe(res4 => {
599                          console.log(res4);
600                          if(res4 == 'RECIBIDA'){
601                              this.httpServiceSRI.autorizarXMLSri(localStorage.getItem('movimiento_id')!).subscribe(res5 => {
602                                  console.log(res5);
603                                  const palabras: string[] = res5.split(' ');
604                                  console.log(palabras)
605                                  localStorage.setItem('fechaAutoriz', palabras.slice(1).join(' '));
606                              });
607                          } else {
608                              console.log('NO SE PUEDE ENVIAR LA FACTURA PARA SU AUTORIZACIÓN')
609                          }
610                      });
611                  }).subscribe(res2 => {
612                      if(res2 == 'CREADO'){
613                          console.log('XML CREADO');
614                      } else {
615                          console.log(res2);
616                      }
617                  });
618              }).subscribe(res3 => {
619                  console.log(res3)
620                  if(res3.codigoError == 'OK'){
621                      console.log('ACTUALIZADO')
622                  } else {
623                      console.log('ERROR')
624                  }
625              });
626          }

```

Figura 61. Método para enviar la factura al SRI.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<factura id="comprobante" version="1.1.0">
  <infoTributaria>
    <ambiente>1</ambiente>
    <tipoEmision>1</tipoEmision>
    <razonSocial>Prueba1</razonSocial>
    <nombreComercial>Prueba1</nombreComercial>
    <ruc>1803742079002</ruc>
    <claveAcceso>3008202301180374207900210030030000000833172584519</claveAcceso>
    <codDoc>01</codDoc>
    <estab>003</estab>
    <ptoEmi>003</ptoEmi>
    <secuencial>000000083</secuencial>
    <dirMatriz>AMBATO</dirMatriz>
  </infoTributaria>
  <infoFactura>
    <fechaEmision>30/08/2023</fechaEmision>
    <obligadoContabilidad>SI</obligadoContabilidad>
    <tipoIdentificacionComprador>05</tipoIdentificacionComprador>
    <razonSocialComprador>DANIEL EDUARDO MONTERO LOPEZ</razonSocialComprador>
    <identificacionComprador>1850294875</identificacionComprador>
    <direccionComprador>Av. Albert Einstein</direccionComprador>
    <totalSinImpuestos>20.00</totalSinImpuestos>
    <totalDescuento>0.00</totalDescuento>
    <totalConImpuestos>
      <totalImpuesto>
        <codigo>2</codigo>
        <codigoPorcentaje>2</codigoPorcentaje>
        <baseImponible>20.00</baseImponible>
        <valor>2.40</valor>
      </totalImpuesto>
    </totalConImpuestos>
    <propina>
    </propina>
    <importeTotal>22.40</importeTotal>
    <moneda>USD</moneda>
  </infoFactura>
  <pagos>
    <pago>
      <formaPago>16</formaPago>
      <total>22.40</total>
    </pago>
  </pagos>
</factura>
```

Figura 62. Ejemplo de factura xml.

En la figura 60 se puede observar el método mediante el cual se envía la factura al SRI. Primero se crea el archivo xml ya firmado digitalmente, a partir de los datos tanto del cliente como de la venta, tomando productos vendidos e importes totales. Luego de generar el archivo xml, se envía mediante el servicio web proporcionado por el SRI para su futura autorización.

Esta generación de facturas se guardan dentro del sistema de archivos de cPanel para su posterior envío por correo electrónico.

[illegible]

Figura 63. Vista de generación de PDF y envío por correo.

```

354 generarPDF() {
355   // Crear un contenedor para el PDF en un elemento de tipo
356   const fixedContainer = document.createElement('div');
357   fixedContainer.style.width = '794px'; // Ancho en puntos para A4
358   fixedContainer.style.position = 'absolute';
359   fixedContainer.style.top = '0';
360   fixedContainer.style.left = '0';
361   fixedContainer.style.background = 'white';
362
363   // Clonar el contenido original y agregarlo al contenedor fijo
364   const contentClone = $QX.cloneNode(true) as HTMLDocument;
365   fixedContainer.appendChild(contentClone);
366   document.body.appendChild(fixedContainer);
367
368   // Descargar la imagen del logo y convertirla a base64
369   const logoUrl = "https://i.ibb.co/qdD8r/Logo-Comercial.png";
370
371   this.getBase64Image(logoUrl).then((logoBase64) => {
372     // Asegurarse de que la imagen se haya descargado correctamente
373     if (logoBase64) {
374       html2canvas(fixedContainer, {options}).then((canvas) => {
375         const imgData = canvas.toDataURL('image/png');
376         const pdfWidth = doc.internal.pageSize.getWidth();
377         const pdfHeight = doc.internal.pageSize.getHeight();
378
379         // Agregar la imagen del contenido al PDF
380         doc.addImage(imgData, 'PNG', 0, 0, pdfWidth, pdfHeight);
381
382         // Agregar la imagen del logo al PDF
383         doc.addImage(logoBase64, 'PNG', 10, 10, 100, 100); // Ajusta las coordenadas y el tamaño según sea necesario
384         document.body.removeChild(fixedContainer); // Eliminar el contenedor fijo después de generar el PDF
385
386         // Guardar el PDF
387         const pdfName = 'documento.pdf';
388         doc.save(pdfName);
389
390         // Subir el PDF o realizar otras acciones necesarias
391         const imageEntity: ImageEntity = {
392           imageBase64: imgData,
393           nombreArchivo: pdfName,
394           codigoError: '',
395           descripcionError: '',
396           nombreArchivoLimpiar: '',
397         };
398
399         this.httpServiceImage.agregarPDF(imageEntity).subscribe(res => {
400           if (res.codigoError === 'OK') {
401             console.log('PDF subido correctamente');
402             this.pdfGenerado = true;
403           } else {
404             console.error('Error al subir el PDF: ', res.descripcionError);
405           }
406         });
407       });
408     } else {
409       console.error('Error al descargar la imagen del logo. ');
410     }
411   });
412 }
413
414 enviarComprobante() {
415   const movimientoID = localStorage.getItem('movimiento_id');
416   this.httpServiceGHI.enviarComprobanteCorreo(movimientoID).subscribe(res => {
417     if (res === 'Correo enviado correctamente') {
418       console.log(res);
419       Swal.fire({
420         icon: 'success',
421         title: 'Finalizado Correctamente.',
422         text: 'Se ha finalizado la venta y enviado el comprobante',
423         showConfirmButton: true,
424         confirmButtonText: 'OK'
425       }).finally(() => {
426         // this.grafForm.reset();
427         let ruta = this.router.url;
428
429         if (ruta.includes('navegacion-cl')) {
430           this.router.navigate(['/navegacion-cl'], { outlets: { 'contentClient': ['ventaspro'] } });
431         }
432         if (ruta.includes('navegacion-facturador')) {

```

Figura 64. Método con la generación y envío de la factura por correo.

Resultados

Por Fecha

sandra7 yp@gmail.com

Factura

Factura con archivos correspondientes

1/5/2024

Más antiguos

edisonlopezec@gmail.com

Factura

Factura con archivos correspondientes

26/4/2024

Carlos Fernando Cortes Landaz...

RE: Cliente no aparece en K2 403753 Nes...

Buen día Carlos.

11/4/2024

Factura

Daniel Montero ECU-VEN

Para sandra7 yp@gmail.com

factura_0105202401180374207900210010010000000015544815919.xml

7 KB

factura_0105202401180374207900210010010000000015544815919.pdf

392 KB

Factura con archivos correspondientes XML y PDF

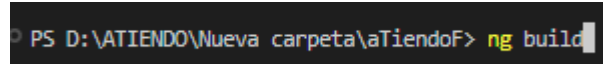
Figura 65. Ejemplo de envío de factura por correo electrónico.

93

En la figura 63 se puede validar la manera en la que se genera el archivo pdf de la factura, así como el envío mediante correo electrónico de la factura electrónica al cliente. Además, en la figura 64 se puede observar un ejemplo del correo que llegaría al cliente al momento de enviar de que el usuario envíe la factura.

3.2.3.5. Despliegue

En la fase de despliegue, se llevó a cabo la integración del módulo hasta el requerimiento dispuesto del segundo incremento. Este despliegue se realizó a nivel de frontend como de backend.



```
PS D:\ATIENDO\Nueva carpeta\atiendoF> ng build
```

Figura 66. Comando para generar carpetas de despliegue en Angular.

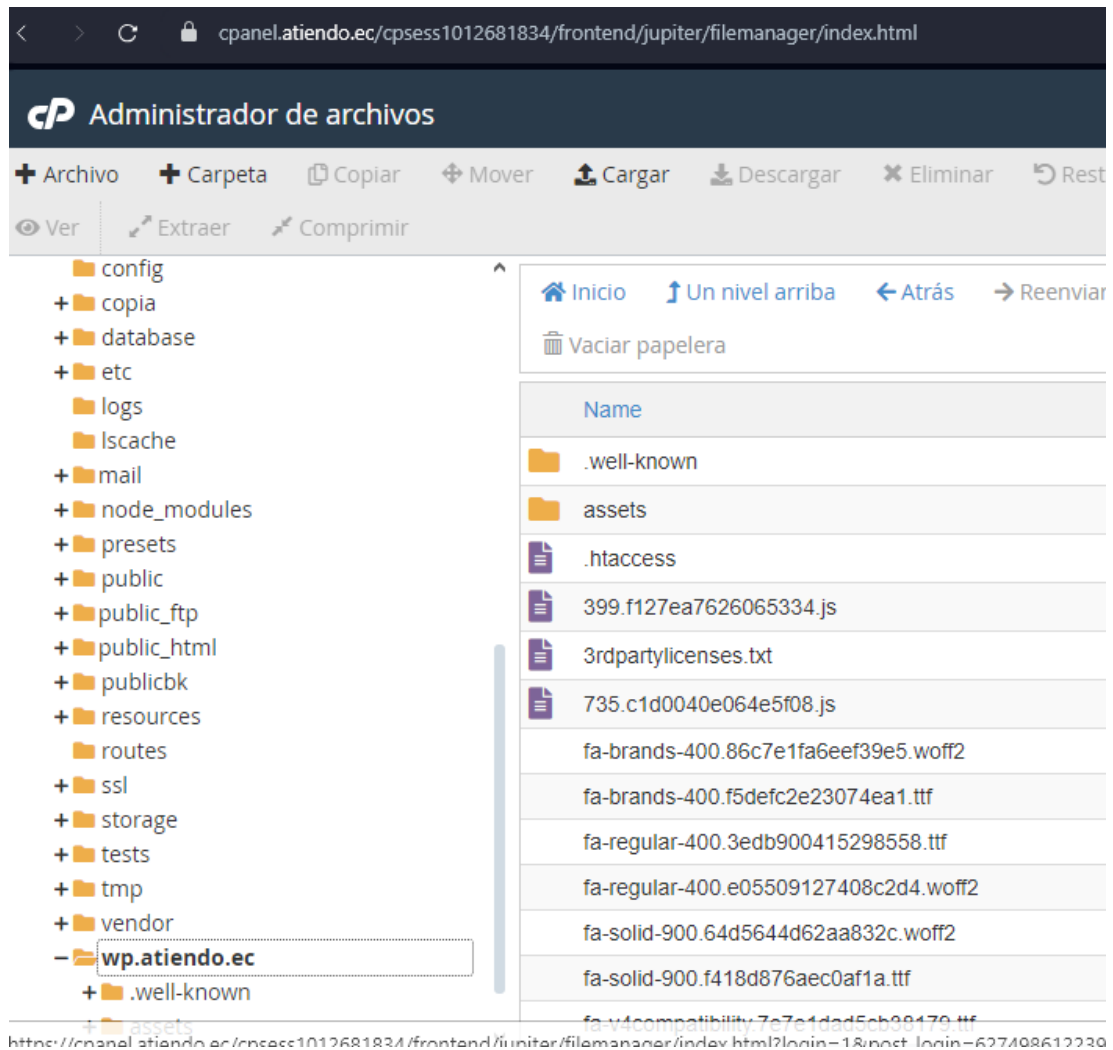


Figura 67. Despliegue de proyecto en cPanel.

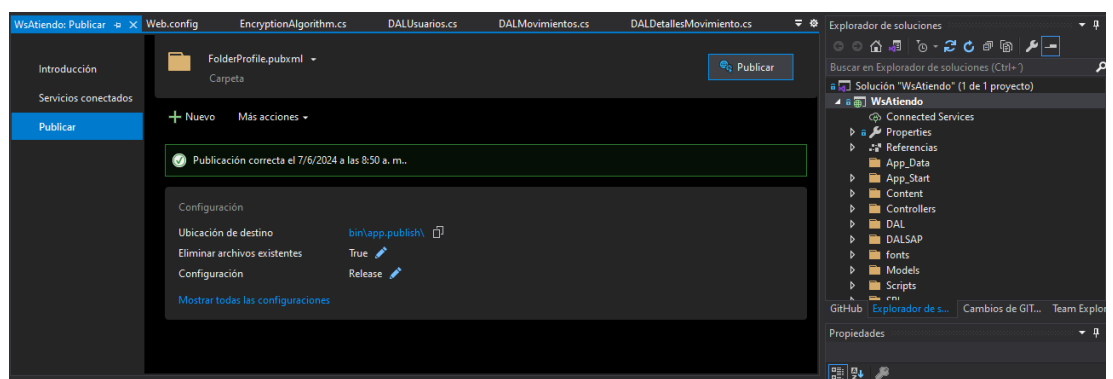


Figura 68. Generación de archivos de despliegue en ASP .Net

Como se observa en las figuras 66 y 67, se desplegaron los archivos necesarios para el correcto funcionamiento en un ambiente de producción mediante el cual se puede acceder al dominio y utilizar las funcionalidades creadas hasta el momento.

3.2.4 Aplicación del tercer incremento (Incremento 3)

3.2.4.1 Comunicación

En esta fase, se realizó un análisis de las necesidades y requerimientos del cliente. El cliente, mediante reuniones manifestó la necesidad de integrar la facturación en este incremento. Se llevaron a cabo intercomunicaciones con el cliente y distintos actores para definir los requerimientos y alcance aplicables en este incremento:

1. **Ingresar Datos de la Firma Electrónica (R_015):** Es necesario que los usuarios puedan ingresar los datos de su firma electrónica para validar las facturas.
2. **Visualización y Exportación de Informe de Cuadre de Caja con Filtros (R_016):** Los usuarios deben poder visualizar y exportar informes de cuadre de caja, aplicando filtros por fecha y almacén.
3. **Visualización y Exportación de Movimientos de Ventas con Filtros (R_017):** Se requiere la capacidad de visualizar y exportar los movimientos de ventas aplicando filtros por fecha y almacén.

3.2.4.2 Planeación

La planeación de este incremento contará con las mismas directrices aplicadas al primer y segundo incremento, analizando los requerimientos y definiendo un cronograma.

Luego de haber revisado de manera detallada los requerimientos planteados para el incremento, se decidió planear el siguiente cronograma:

Tabla 13. Cronograma INcremento 3.

CRONOGRAMA INCREMENTO 3									
Código	Actividades	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	DÍA 6	DÍA 7	DÍA 8
R_015	Configurar datos de firma electrónica.								
R_016	Módulo de cuadro de caja.								
R_017	Módulo de movimientos.								

3.2.4.3 Modelado

En esta fase, se desarrollaron los modelos y prototipos necesarios para la implementación del tercer incremento. Los componentes modelados serán prototipos al igual que en el primer y segundo incremento.

a. Configurar datos de firma electrónica.

Figura 69. Wireframe de la vista de configuración de datos de firma electrónica.

En la figura 68 se observa la manera en la que se podrá configurar los datos de la firma electrónica, como la carga del certificado digital y su contraseña. Así como el correo de envío de mails.

b. Módulo de cuadre de caja

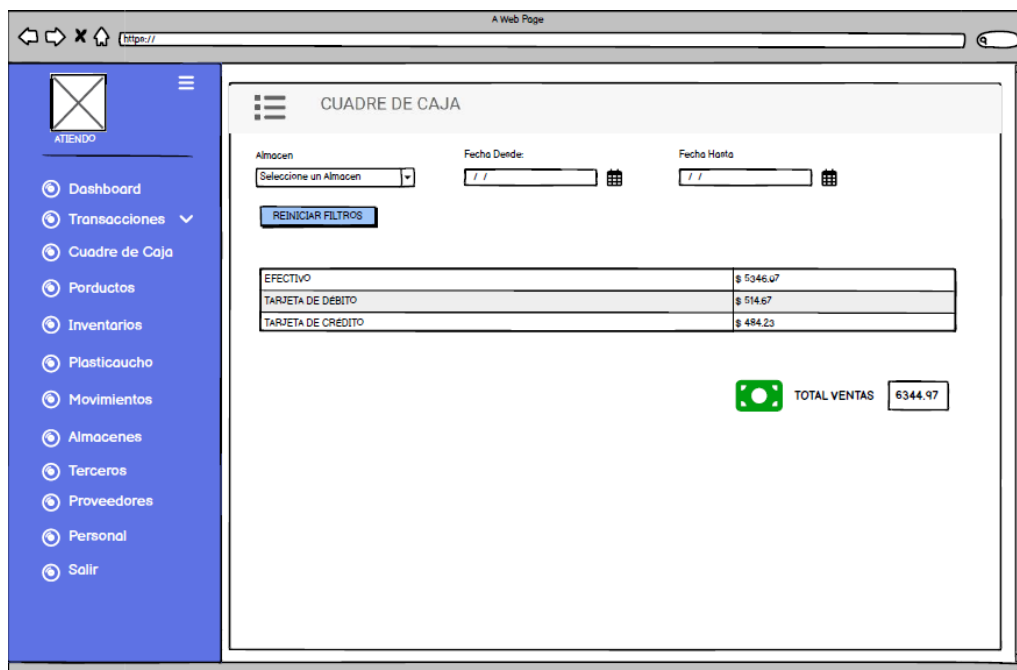


Figura 70. Wireframe del módulo de Cuadre de Caja.

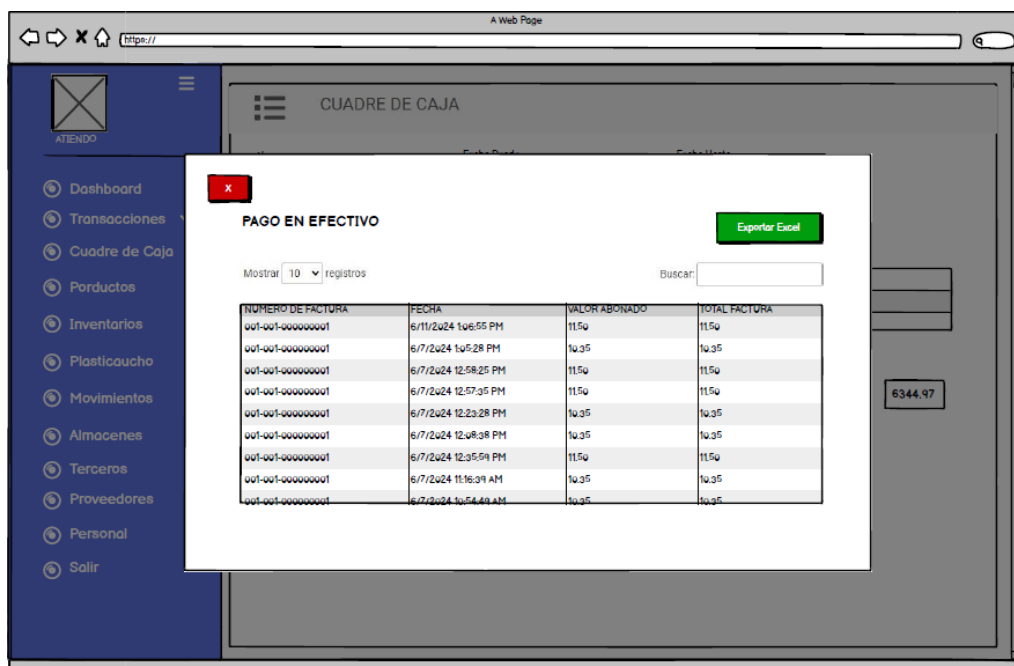


Figura 71. Wireframe del detalle del cuadre de caja.

Como se observa en la figura 69, se creará una vista tipo informe detallado con el cuadro de caja del usuario. En la parte superior contará con filtros de fechas o por el almacén; además, se podrá exportar el informe detallado a Excel tal como se observa en la figura 70.

c. Módulo de Movimientos

The wireframe shows a web application interface for the 'Movimientos' (Movements) module. On the left is a blue sidebar menu with the user's name 'ATIENDO' and a list of navigation items: Dashboard, Transacciones, Cuadre de Caja, Productos, Inventarios, Plásticocha, Movimientos, Almacenes, Terceros, Proveedores, Personal, and Salir. The main content area is titled 'MOVIMIENTOS' and features filter controls for 'Almacen' (a dropdown menu), 'Fecha Desde' and 'Fecha Hasta' (date pickers), and 'Tipo' (a dropdown menu). Below the filters are buttons for 'REINICIAR FILTROS' and 'Exportar A Excel'. A table displays a list of transactions with columns for PRODUCTO, PUNTO DE EMISION, FECHA, TIPO, CANTIDAD, and VALOR TOTAL. The table shows a mix of 'Egreso' (Withdrawal) and 'Ingreso' (Deposit) transactions for various products and dates in June 2024.

PRODUCTO	PUNTO DE EMISION	FECHA	TIPO	CANTIDAD	VALOR TOTAL
AND AM N9 36	001	6/11/2024 12:51:45 PM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 10:5:13 PM	Egreso	1	10.00
AND AM N9 36	001	6/7/2024 12:58:10 PM	Egreso	1	10.00
AND AM N9 36	001	6/7/2024 12:36:31 PM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 12:01:05 PM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 12:05:53 PM	Egreso	1	10.00
AND AM N9 36	001	6/7/2024 12:05:14 PM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 11:50:07 AM	Ingreso	100	100.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 11:38:52 AM	Ingreso	100	100.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 11:16:08 AM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 10:41:01 AM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 10:46:56 AM	Egreso	1	10.00
ARES AZUL BLANCO PASADOR UNISEX PARA JUGAR	001	6/7/2024 9:01:39 AM	Egreso	1	10.00

Figura 72. Wireframe del módulo de Movimientos.

En la figura 70 se puede validar la vista con la que el usuario podrá revisar sus movimientos en función de filtros que podrá configurar en la parte superior de la tabla de detalles. Finalmente, este informe se podrá exportar a Excel para su análisis más detallado.

3.2.4.4. Construcción

La fase de construcción correspondiente al tercer incremento involucró la continuación del desarrollo e implementación de módulos de información, de acuerdo con los requerimientos habilitados para este tercer y último incremento.

Estructura de los módulos en Angular.

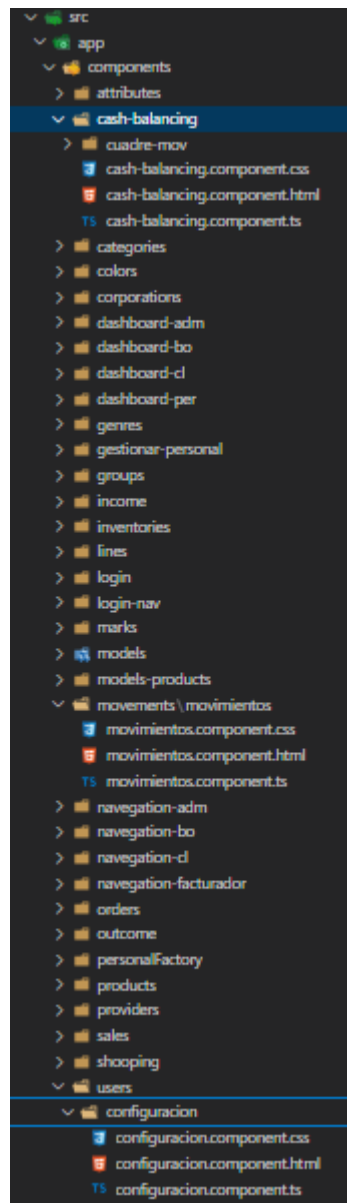


Figura 73. Estructura de los tres componentes en Angular.

Como se observa en la figura 72, a la estructura presentada en Angular en el primer y segundo incremento, se le añadieron tres componentes adicionales con los archivos necesarios para la ejecución de los tres módulos. Además, se validó la creación de los modelos como también los servicios.

Por otro lado, en el backend se procedió con la creación del modelo (entidad), DAL y controladores correspondientes a los tres módulos presentados para el actual incremento.

a. Módulo de configuración.

The screenshot shows the 'aiendo' application's configuration interface. On the left is a blue sidebar with a menu containing items like Dashboard, Transacciones, Cuadro de Caja, Productos, Inventarios, Pedido Superido, Plasticocho, Movimientos, Almacenes, Terceros, Proveedores, Personal, and Settings. The main content area is white and titled 'CERTIFICADOS DIGITALES'. It features a 'Ver Perfil' button, a 'Certificado Electronico (Firma)' dropdown menu, a 'Nueva Clave' input field, and an 'Actualizar certificado' button. Below this is the 'Correo Electronico' section with an 'Email de Envio de Comprobantes' input field, 'Contraseña Actual' and 'Contraseña Nueva' input fields, and an 'Actualizar correo' button. The 'Cambiar Imagen' section at the bottom has an 'Imagen Marca' input field, an 'Eliminar Imagen' button, and another dropdown. On the right, a user profile card for 'Geo/Co Founder' displays a bio and social media icons.

Figura 74. Vista de configuración de datos del certificado, correo e imagen de la factura.

```

141 actualizarCert() {
142
143     console.log("entro al metodo")
144     const passnuevo = this.corporationForm.value.nombreComercial2 ?? "";
145     const passactual = this.corporationForm.value.razonSocial ?? "";
146     const emailCert = this.emailForm.value.email ?? "";
147
148     console.log(passnuevo + " - " + passactual + " - " + emailCert)
149
150
151
152
153     if (!this.corporationForm.valid || !this.razonSocialInclude) {
154         console.log("entro al if principal")
155         this.corporationForm.markAllAsTouched();
156     } else {
157
158         if (!this.razonSocialInclude) {
159             console.log("entro al if principal2")
160             //Agregar el certificado y la clave
161             console.log("andad")
162         }
163
164         if (passactual == passnuevo) {
165             Swal.fire({
166                 icon: 'success',
167                 title: 'Actualizado Correctamente.',
168                 text: 'Se ha actualizado la informacion',
169                 showCancelButton: true,
170                 confirmButtonText: 'Ok'
171             }).finally(() => {
172                 switch (this.fun) {
173                     case 'admin':
174                         this.router.navigate(['/navegacion-admin'], { outlets: { 'contentAdmin': 'usuario' } });
175                         break;
176                     case 'client':
177                         this.router.navigate(['/navegacion-cl'], { outlets: { 'contentClient': 'usuario' } });
178                         break;
179                 }
180             });
181         } else {
182             console.log("entro al else")
183             const imageEntity: ImagesEntity = {
184                 imageBase64: this.certificadoBase64,
185                 nombreArchivo: this.certificadoName,
186                 codigoError: "",
187                 descripcionError: "",
188                 nombreArchivoFirmar: "",
189             };
190
191             console.log("ESTE ES EL EMAIL ADMIN ", localStorage.getItem('socioId'))
192
193             //Sociedad con el id
194
195             const sociedad: SociedadEntity = {
196                 idSociedad: JSON.parse(localStorage.getItem('socioId')) || "[]",
197                 idGrupo: "",
198                 nombre_comercial: "",
199                 id_fiscal: "",
200                 email: "",
201                 tipo_ambienteId: "",
202                 telefono: "",
203                 password: "",
204                 funcion: "",
205                 razon_social: "",
206                 url_certificado: "",
207                 clave_certificado: ""
208             }
209
210
211
212

```

Figura 75. Método para modificar los datos del certificado digital.

```

1189 public static Sociedades ActualizarCertificado(SociedadesEntity sociedad)
1190 {
1191     string key = "Venus2022@!";
1192     Sociedades sociedadesResponse = new Sociedades();
1193     if (sociedad.url_certificado.Equals(""))
1194     {
1195         using (SqlConnection mConnection = new SqlConnection(ConnectionString))
1196         {
1197             string query5 = @"UPDATE sociedades
1198                             SET clave_certificado=?clave_certificado,
1199                                 updated_at=?updated_at
1200                             WHERE id=?id";
1201             mConnection.Open();
1202             using (MySQLTransaction trans = mConnection.BeginTransaction())
1203             {
1204                 try
1205                 {
1206                     using (MySQLCommand myCmd = new MySQLCommand(query5, mConnection, trans))
1207                     {
1208                         myCmd.Parameters.AddWithValue("?clave_certificado", EncryptionAlgorithm.Encrypt(sociedad.clave_certificado, key));
1209                         myCmd.Parameters.AddWithValue("?id", sociedad.idSociedad);
1210                         myCmd.Parameters.AddWithValue("?updated_at", System.DateTime.Now);
1211                         myCmd.CommandType = CommandType.Text;
1212                         var rowCount = myCmd.ExecuteNonQuery();
1213                         if (rowCount > 0)
1214                         {
1215                             trans.Commit();
1216                             sociedadesResponse.codigoError = "OK";
1217                             SendEmail enviarCorreoC = new SendEmail();
1218                             enviarCorreoC.Envia("sistemas@linda@elasticocho.com", "Inicio 2023", sociedad.email);
1219                         }
1220                         else
1221                         {
1222                             trans.Rollback();
1223                             sociedadesResponse.codigoError = "E";
1224                             sociedadesResponse.descripcionError = "3. No se pudo actualizar la Sociedad.";
1225                         }
1226                     }
1227                 }
1228             }
1229         }
1230     }
1231 }

```

Figura 76. Método de encriptación de la contraseña en el backend.

```

303 actualizarEmail() {
304     const email = this.emailForm.value.email ?? "";
305     const password = this.emailForm.value.password ?? "";
306
307     if (this.emailForm.valid) {
308         this.emailForm.markAsUntouched();
309     } else {
310         const userEntity: SociedadesEntity = {
311             idGrupo: "",
312             nombre_comercial: "",
313             tipo_ambiental: "",
314             id_fiscal: "",
315             email: "",
316             telefono: "",
317             password: "",
318             funcion: "",
319             idSociedad: JSON.parse(localStorage.getItem('societadid') || '{}'),
320             razon_social: "",
321             url_certificado: "",
322             clave_certificado: "", // cambiar por new
323             pass_certificado: password, // activar validacion
324             email_certificado: email
325         };
326         this.HttpService.actualizarClaveCorreo(userEntity).subscribe(res => {
327             if (res.codigoError == "OK") {
328                 Swal.fire({
329                     icon: "success",
330                     title: "Actualizado Correctamente.",
331                     text: "Se ha actualizado la información",
332                     showCancelButton: true,
333                     confirmButtonText: "OK"
334                 }).finally(() => {
335                     switch (this.fun) {
336                         case "admin":
337                             this.router.navigate(['/navigation-admin'], { outlets: { 'contentAdmin': ['configuracion-user'] } });
338                             break;
339                         case "client":
340                             this.router.navigate(['/navigation-client'], { outlets: { 'contentClient': ['configuracion-user'] } });
341                             break;
342                     }
343                 });
344             } else {
345                 Swal.fire({
346                     icon: "error",
347                     title: "Ha ocurrido un error.",
348                     text: res.descripcionError,
349                     showCancelButton: false,
350                     confirmButtonText: "OK"
351                 });
352             }
353         });
354     }
355 }
356
357 onChargeFile(target: any): void {
358     if (target.value != "") {
359         this.fileToUpload = target.files[0];
360         let reader = new FileReader();
361         reader.onload = (event: any) => {
362             this.imageUrl = event.target.result;
363             this.imageName = this.imageUrl.split('.')[1];
364             this.imageName = this.fileToUpload.name;
365         }
366         reader.readAsDataURL(this.fileToUpload);
367     }
368 }

```

Figura 77. Método para modificar los datos del mail de envío de comprobantes.

En la figura 74 y 76, se puede validar la forma en la que se realiza modificación de datos correspondientes al certificado digital y al correo de envío de mails. Por otra parte, en la figura 75 se visualiza en la aplicación ASP .Net la manera en la que se encripta la contraseña del certificado digital ingresada por el usuario, este se encripta con la metodología AES; este campo es muy sensible ya que la información debe ser inaccesible y poseer un método confiable de encriptación.

b. Módulo de cuadre de caja.

NUMERO DE FACTURA	FECHA	VALOR ABONADO	TOTAL FACTURA
001-001-000000001	6/17/2024 9:19:43 AM	11.50	11.50
001-001-000000001	6/17/2024 1:43:59 AM	36.40	36.40
001-001-000000001	6/11/2024 1:06:55 PM	11.50	11.50
001-001-000000001	6/7/2024 1:05:28 PM	10.35	10.35
001-001-000000001	6/7/2024 12:58:25 PM	11.50	11.50
001-001-000000001	6/7/2024 12:57:35 PM	11.50	11.50
001-001-000000001	6/7/2024 12:23:28 PM	10.35	10.35
001-001-000000001	6/7/2024 12:08:38 PM	10.35	10.35
001-001-000000001	6/7/2024 12:35:59 PM	11.50	11.50

Figura 78. Vista detallada del cuadre de caja con filtros.

```

19 export class CuadroMovimientos implements OnInit {
20   ngOnInit(): void {
21     didOpen: () => {
22       if (almacen != '' && fechadesde != '' && fechahasta != '') {
23         this.httpsMovimientosService.obtenerMovimientosAlmacenFechaPAGO(almacen, fechadesde, fechahasta, this.tipoPago).subscribe((resi) => {
24           if (resi.codigoError != 'OK') {
25             Swal.fire({
26               icon: 'error',
27               title: 'No se pudo obtener movimientos.',
28               text: resi.descripcionError,
29               showCancelButton: false,
30             });
31           } else {
32             this.lstMovimientos = resi.lstMovimientos;
33             this.lstMovimientos.forEach((movimiento, index) => {
34               const numeroFactura = `${movimiento.pto_emision}-${movimiento.camp_adit}-${movimiento.secuencial}`;
35               movimiento.nroFactura = numeroFactura;
36             });
37             this.dtTrigger.next('');
38             Swal.close();
39           }
40         });
41       } else if (almacen != '' && fechadesde == '' && fechahasta == '') {
42         this.httpsMovimientosService.obtenerMovimientosAlmacenNombrePAGO(almacen, this.tipoPago).subscribe((resi) => {
43           if (resi.codigoError != 'OK') {
44             Swal.fire({
45               icon: 'error',
46               title: 'No se pudo obtener movimientos.',
47               text: resi.descripcionError,
48               showCancelButton: false,
49             });
50           } else {
51             this.lstMovimientos = resi.lstMovimientos;
52             this.lstMovimientos.forEach((movimiento, index) => {
53               const numeroFactura = `${movimiento.pto_emision}-${movimiento.camp_adit}-${movimiento.secuencial}`;
54               movimiento.nroFactura = numeroFactura;
55             });
56             this.dtTrigger.next('');
57             Swal.close();
58           }
59         });
60       } else if (almacen == '' && fechadesde != '' && fechahasta != '') {
61         this.httpsMovimientosService.obtenerMovimientosFechasPAGO(sociedadid, fechadesde, fechahasta, this.tipoPago).subscribe((resi) => {
62           if (resi.codigoError != 'OK') {
63             Swal.fire({
64               icon: 'error',
65               title: 'No se pudo obtener movimientos.',
66               text: resi.descripcionError,
67               showCancelButton: false,
68             });
69           } else {
70             this.lstMovimientos = resi.lstMovimientos;
71             console.log("Aquí lstMovimientos", this.lstMovimientos);
72             this.lstMovimientos.forEach((movimiento, index) => {
73               const numeroFactura = `${movimiento.pto_emision}-${movimiento.camp_adit}-${movimiento.secuencial}`;
74               console.log("Aquí numero factura", numeroFactura);
75               movimiento.nroFactura = numeroFactura;
76             });
77             this.dtTrigger.next('');
78             Swal.close();
79           }
80         });
81       } else if (almacen == '' && fechadesde == '' && fechahasta == '') {

```

Figura 79. Lógica para visualizar los detalles para el cuadro de caja.

```

281 exportarXLSX() {
282   const movimientosSinProductId = this.listaMovimientos.map(movimiento => {
283     const { ...movimientoSinProductId } = movimiento;
284     return movimientoSinProductId;
285   });
286   const renombrar = movimientosSinProductId.map(movimiento => {
287     return {
288       'NUMERO DE FACTURA': movimiento.nroFactura,
289       'FECHA EMISION': movimiento.updated_at,
290       'VALOR ABONADO': movimiento.detalle_pago,
291       'TOTAL FACTURA': movimiento.importe_total,
292     };
293   });
294   const wb = XLSX.utils.book_new();
295   const wsData = XLSX.utils.json_to_sheet(renombrar);
296   wsData['!cols'] = [
297     { width: 20 },
298     { width: 20 },
299     { width: 20 },
300     { width: 20 }
301   ];
302   wsData['!margins'] = [{ s: { r: 0, c: 0 }, e: { r: 0, c: 3 } }];
303   wsData['A1'].s = {
304     font: { bold: true, sz: 24 },
305     alignment: { horizontal: 'center', vertical: 'center' },
306     fill: {
307       patternType: 'solid',
308       fgColor: { rgb: 'FFFF00' }
309     }
310   };
311   wsData['A1'] = { v: `${this.metodoPago}`, t: 's' };
312   // Establecer estilos para el titulo
313   wsData['A1'].s = {
314     font: { bold: true, sz: 24 },
315     alignment: { horizontal: 'center', vertical: 'center' },
316     fill: {
317       patternType: 'solid',
318       fgColor: { rgb: 'FFFF00' }
319     }
320   };
321   // Agregar los datos de la tabla
322   XLSX.utils.sheet_add_json(wsData, renombrar, { origin: 'A2' });
323   // Agregar la hoja de Excel al libro
324   XLSX.utils.book_append_sheet(wb, wsData, 'Movimientos');
325   // Generar el archivo Excel y guardarlo
326   const excelBuffer = XLSX.write(wb, { bookType: 'xlsx', type: 'array' });
327   const date = new Date();
328   const day = date.getDate();
329   const month = date.getMonth() + 1;
330   const year = date.getFullYear();
331   const fecha = `${day}-${month}-${year}`;
332   this.saveExcelFile(excelBuffer, `${this.metodoPago}_${fecha}.xlsx`);
333 }
334
335 private saveExcelFile(buffer: any, fileName: string): void {
336   const data: Blob = new Blob([buffer], { type: 'application/octet-stream' });
337   const url: string = window.URL.createObjectURL(data);
338   const link: HTMLAnchorElement = document.createElement('a');
339   link.href = url;
340   link.download = fileName;
341   link.click();
342   setTimeout(() => {
343     window.URL.revokeObjectURL(url);
344   }, 1000);
345 }

```

Figura 80. Exportar archivo a Excel del cuadro de caja.

En la figura 78 se observa la manera en la que, por medio de lógica de programación, se realiza la visualización de datos en el módulo correspondiente. Además, en la figura 79 se visualiza la manera en la que se realiza la exportación del archivo a Excel en función de los filtros aplicados con anterioridad.

c. Módulo de movimientos.

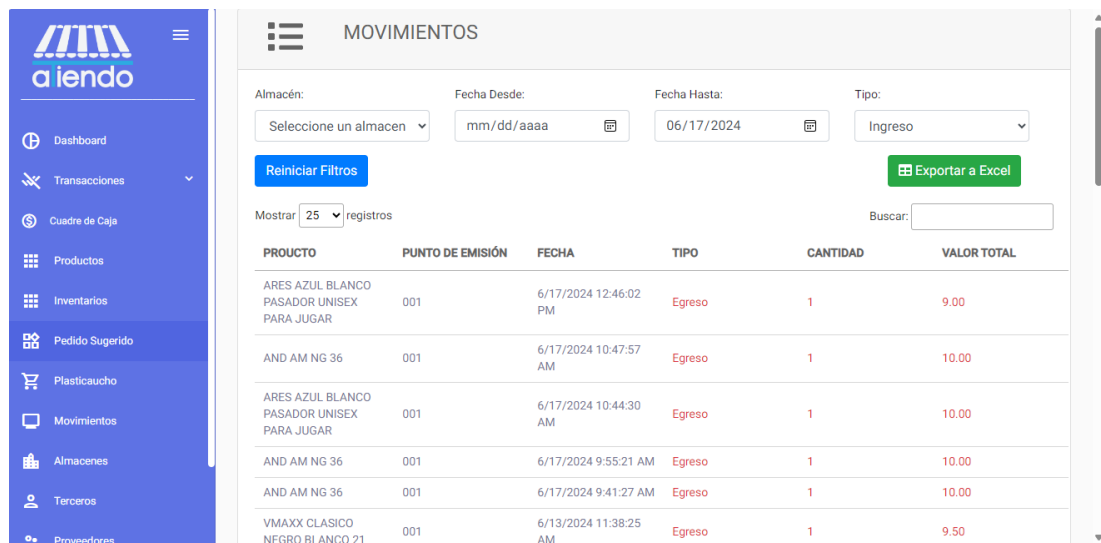


Figura 81. Vista del módulo de movimientos

```
ngOnInit(): void {
  this.httpServiceAlm.obtenerAlmacenesSociedad(sociedadNew).subscribe(res => {
    Swal.fire({
      icon: 'error',
      title: 'Ha ocurrido un error.',
      text: res.descripcionError,
      showConfirmButton: false,
    });
  } else {
    console.log("ENTRO 161")

    if(!this.esFacturador){
      this.lstAlmacenes = res.lstAlmacenes;
    }
  }
});
Swal.fire({
  title: 'CARGANDO...',
  html: 'Se están cargando los detalles.',
  timer: 30000,
  didOpen: () => {
    Swal.showLoading();
    this.httpService.obtenerDetalleMovimientoSociedad(sociedadNew).subscribe(res => {
      if (res.codigoError != "OK") {
        Swal.fire({
          icon: 'error',
          title: 'Ha ocurrido un error.',
          text: res.descripcionError,
          showConfirmButton: false,
        });
      } else {
        console.log("ENTRO 180")

        if(!this.esFacturador){
          this.lstDetalleMovimientos = res.lstDetalleMovimientos;
        }

        this.dtTrigger.next('');
        Swal.close();
      }
    });
  },
}).then((result) => {
  /* Read more about handling dismissals below */
  if (result.dismiss === Swal.DismissReason.timer) {
    console.log('I was closed by the timer');
  }
});
this.fechaActual = new Date().toISOString().split('T')[0];
}
```

Figura 82. Método para listar los movimientos.

En la figura 81 se puede observar el método mediante el cual se listan los movimientos de acuerdo a la sociedad o usuario; además, existen filtros para esta vista como el almacén, fecha desde, fecha hasta y el tipo si es una egreso (venta) o ingreso (compra).

Además, este módulo también posee la generación de un archivo Excel que sirve como informe de ventas realizadas en función de los filtros anteriormente seleccionados.

3.2.4.5. Despliegue

En la fase de despliegue, se llevó a cabo la integración completa de los módulos revisados anteriormente, completando así hasta el requerimiento dispuesto del tercer incremento. Este despliegue se realizó a nivel de frontend como de backend.

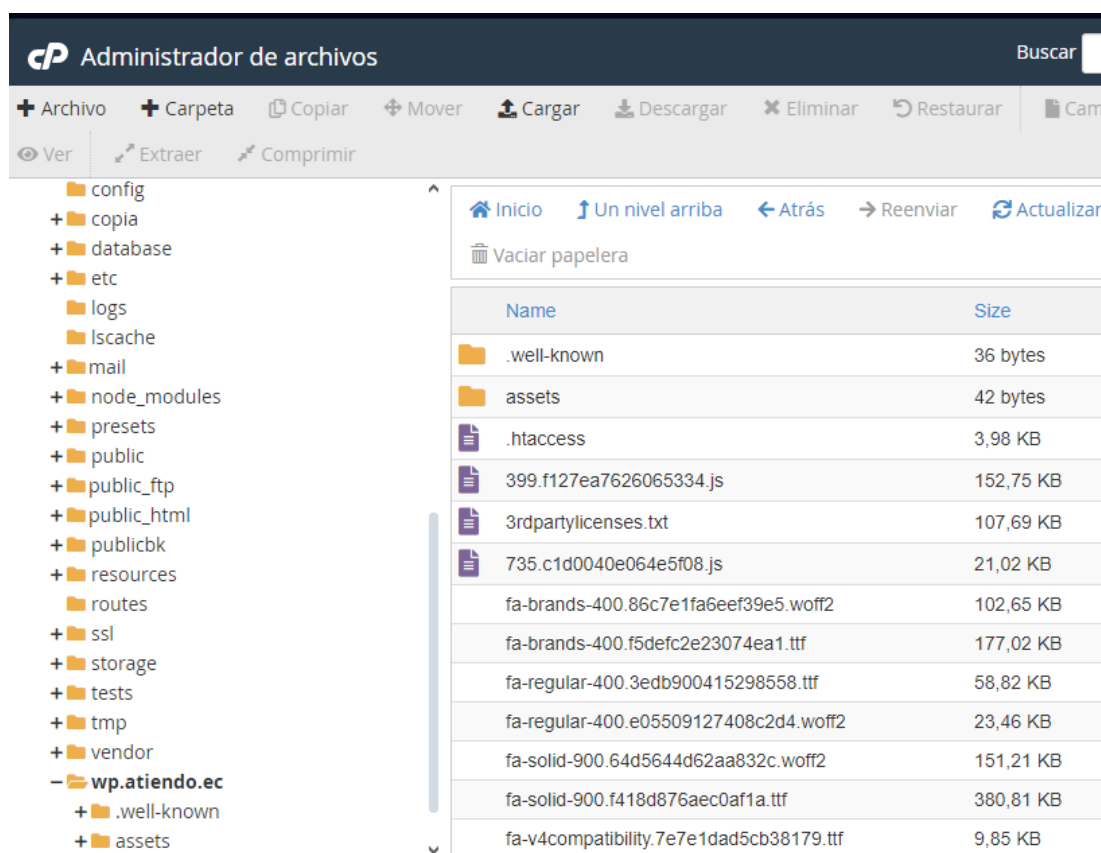


Figura 83. Despliegue de proyecto en cPanel.

o_2024-05-	^			
	Nombre	Fecha de modificación	Tipo	Tamaño
	bin	7/6/2024 12:56 p. m.	Carpeta de archivos	
	Content	7/6/2024 12:56 p. m.	Carpeta de archivos	
	fonts	7/6/2024 12:56 p. m.	Carpeta de archivos	
	Scripts	7/6/2024 12:56 p. m.	Carpeta de archivos	
	Views	7/6/2024 12:56 p. m.	Carpeta de archivos	
	favicon.ico	22/8/2023 10:27 a. m.	Icono	32 KB
	PrecompiledApp.config	7/6/2024 12:56 p. m.	XML Configuratio...	1 KB
	Web.config	4/6/2024 12:40 p. m.	XML Configuratio...	6 KB

Figura 84. Generación de archivos de despliegue del backend.

Como se observa en las figuras 82 y 83, se desplegaron los archivos necesarios para el correcto funcionamiento en un ambiente de producción mediante el cual se puede acceder al dominio y utilizar las funcionalidades finales, teniendo en conjunto el despliegue completo de los tres incrementos.

3.3. Evaluación aplicando la norma ISO/IEC 25010

Finalizando la aplicación del modelo de desarrollo de software incremental con los tres incrementos propuestos, y obtener buenas retroalimentaciones, se procede con las pruebas de rendimiento de los servicios web que se relacionen directamente con la facturación electrónica, entre otros. Estas pruebas se evaluarán bajo la normativa ISO/IEC 25010.

3.3.1 Norma ISO/IEC 25010

La norma ISO/IEC 25010 define la eficiencia de desempeño como la característica de calidad que representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Comportamiento temporal: capacidad del producto para completar las tareas dentro de los límites de tiempo especificados.
- Respuesta: capacidad del producto para proporcionar una respuesta a una solicitud dentro de los límites de tiempo especificados.
- Utilización de recursos: capacidad del producto para utilizar los recursos disponibles de manera eficiente.

Para evaluar el rendimiento de un producto software, se deben seleccionar las métricas adecuadas para el propósito de la evaluación. Las métricas deben ser relevantes para las subcaracterísticas de rendimiento que se quieren evaluar y deben ser capaces de cuantificar el rendimiento de manera precisa.

La evaluación del rendimiento se puede realizar mediante una serie de técnicas, como:

- Pruebas de rendimiento: pruebas que se realizan para medir el rendimiento del producto en condiciones controladas.
- Monitorización del rendimiento: seguimiento del rendimiento del producto en condiciones reales de funcionamiento.

- Análisis de rendimiento: análisis de los datos de rendimiento para identificar áreas de mejora.

En el marco del proceso de evaluación de rendimiento de los servicios web POST, se llevó a cabo un exhaustivo conjunto de pruebas de estrés y carga utilizando la herramienta Apache JMeter. El objetivo principal era evaluar la capacidad de los servicios para manejar situaciones de carga elevada y estrés, identificar cuellos de botella y asegurar la estabilidad del sistema bajo condiciones extremas.

Para la realización de las pruebas del uso de CPU correspondientes, se utilizó una máquina con especificaciones regulares y sin sobrepasar el límite de un computador de gama media o baja. Además, estas pruebas se trabajan mediante el modelo proporcionado por Apache JMeter, en donde se trabaja con un agente que permite monitorizar el uso de CPU por cada servicio web.

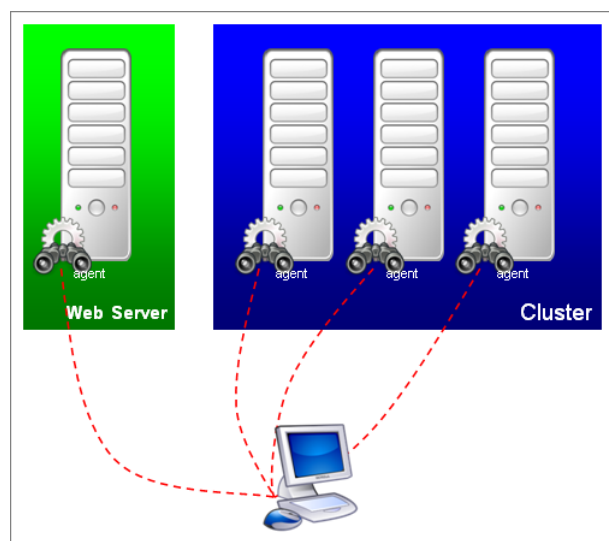


Figura 85. Modelo de pruebas de rendimiento de CPU.

Configuración del Entorno:

- Se implementaron cinco servicios web POST y GET, identificados por el nombre de este.
- Cada servicio se encuentra alojado en un entorno aislado con configuraciones estándar de producción.

- Se utilizó Apache JMeter para simular el tráfico de usuarios, emitiendo solicitudes POST y GET realistas.

Pruebas de Carga: Se realizaron pruebas de carga para evaluar la capacidad del sistema para manejar un volumen creciente de solicitudes. Se configuró JMeter para aumentar gradualmente la carga hasta alcanzar 37 solicitudes por segundo. Durante esta prueba, se monitorearon las siguientes métricas clave:

- Tiempo de respuesta: Se midió el tiempo que tardaban las solicitudes en ser procesadas.
- Uso de CPU: Se evaluó el rendimiento del sistema en términos de utilización de la CPU.
- Tasa de error: Se registraron y analizaron las solicitudes que resultaron en errores.

Pruebas de Estrés: Las pruebas de estrés se diseñaron para evaluar la robustez del sistema al someterlo a una carga superior a la capacidad prevista. JMeter fue configurado para aumentar la carga de manera agresiva hasta alcanzar 74 solicitudes por segundo, simulando un escenario extremo.

Tabla 14. Servicios web aplicables para las pruebas de rendimiento.

Método	URL de Web Service	Descripción
POST	https://outgoapp.plasticaucho.com/WsAtiendoFinal/api/almacenes/ObtenerAlmacenesSociedad	Este servicio devuelve todos los almacenes que existan en la sociedad del usuario logueado.
POST	https://outgoapp.plasticaucho.com/WsAtiendoFinal/api/inventarios/ObtenerPortafolios	Devuelve el inventario en función del almacén
GET	http://atiendoprueba.somee.com/api/movimientos/CrearXML	Genera el archivo xml de la factura para su envío al SRI.
GET	https://outgoapp.plasticaucho.com/WsAtiendoFinal/api/sriws/RecibirXMLSri	Se recibe la respuesta del SRI confirmando la recepción del archivo xml.
GET	https://outgoapp.plasticaucho.com/WsAtiendoFinal/api/sriws/AutorizarXMLSri	Se recibe la respuesta de autorización de la factura o archivo xml del SRI.

Los resultados serán medidos de acuerdo con los números que la herramienta brinde. La tabla muestra los valores promedio, mínimo y máximo de tiempo de respuesta, la desviación estándar, los errores y otros detalles relacionados con el rendimiento y la eficiencia del sistema o aplicación probada. La fila 'HTTP Request' muestra los valores para cada solicitud HTTP realizada y la fila 'TOTAL' muestra los valores totales para todas las solicitudes HTTP. Aquí está una interpretación de los valores en la tabla:

- **Samples:** El número total de solicitudes HTTP realizadas durante las pruebas.
- **Average:** El tiempo de respuesta promedio de todas las solicitudes HTTP.
- **Min:** El tiempo de respuesta más corto de todas las solicitudes HTTP.
- **Max:** El tiempo de respuesta más largo de todas las solicitudes HTTP.
- **Std. Dev.:** La desviación estándar de los tiempos de respuesta de todas las solicitudes HTTP.
- **Error %:** El porcentaje de solicitudes HTTP que resultaron en un error.
- **Throughput:** La cantidad de solicitudes HTTP por segundo que se realizaron durante las pruebas.
- **Received KB/sec.:** La cantidad de datos recibidos por segundo durante las pruebas.
- **Sent KB/sec.:** La cantidad de datos enviados por segundo durante las pruebas.
- **Avg.Bytes:** El tamaño promedio de las respuestas HTTP recibidas.

Finalmente, se establecieron las métricas aplicables para las pruebas de rendimiento, las mismas que serán evaluadas de acuerdo con los siguientes criterios de aceptación:

Tabla 15. Criterios de aceptación ISO/IEC 25010.

Criterio	Resultado Óptimo Pruebas de Carga	Resultado Óptimo Pruebas de Estrés
Tiempo de Respuesta Promedio	0.01 a 3 segundos	0.01 a 8 segundos
Uso de CPU	Menor al 20%	Menor al 25%
Tasa de Error	Menor al 0.05%	Menor al 0.50%

3.3.2 Pruebas de Carga

a. ObtenerAlmacenesSociedad

La prueba de carga con 37 usuarios simultáneos brinda los siguientes resultados con un primer método POST para obtener los almacenes de la sociedad o usuario.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
ObtenerAlma...	37	291	0	372	37.32	0.00%	3.7/sec	6.47	1.06	1786.0
TOTAL	37	291	0	372	37.32	0.00%	3.7/sec	6.47	1.06	1786.0

Figura 86. Resultados de la prueba del primer servicio web.

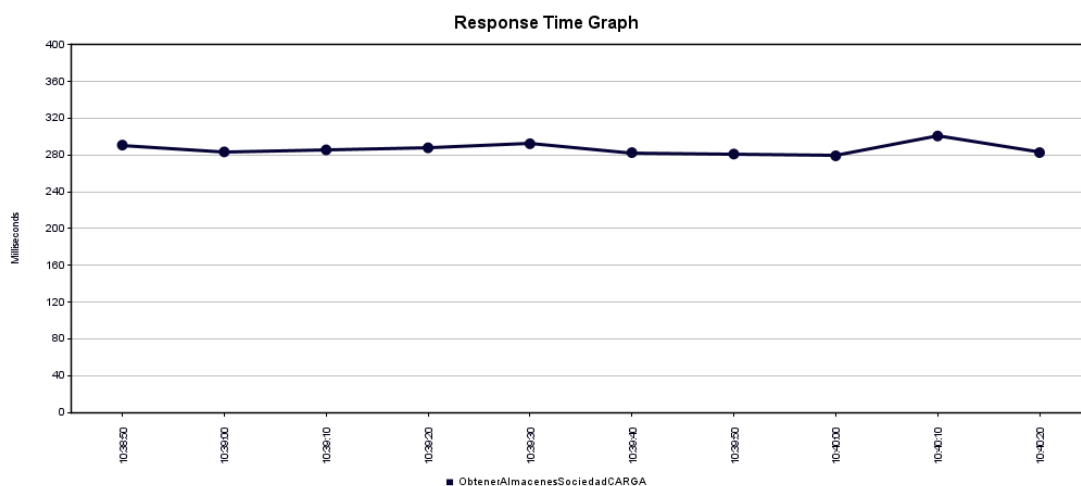


Figura 87. Tiempo de respuesta del primer servicio en un lapso de tiempo.

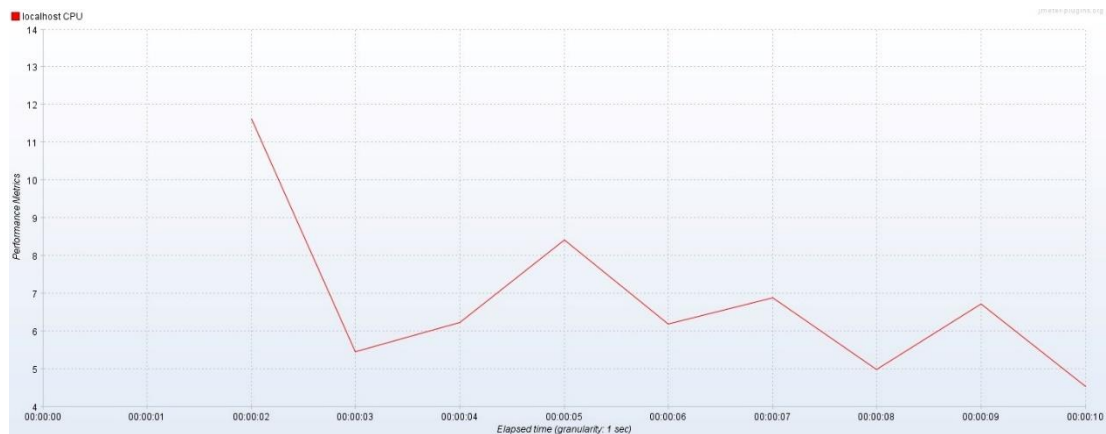


Figura 88. Uso de CPU con el primer servicio en ejecución.

b. ObtenerPortafolios

La prueba de carga con 37 usuarios simultáneos brinda los siguientes resultados con un segundo método POST para obtener los portafolios.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
ObtenerPorta...	37	261	0	277	6.49	0.00%	3.7/sec	58.49	1.05	16153.0
TOTAL	37	261	0	277	6.49	0.00%	3.7/sec	58.49	1.05	16153.0

Figura 89. Resultados de la prueba del segundo servicio web.

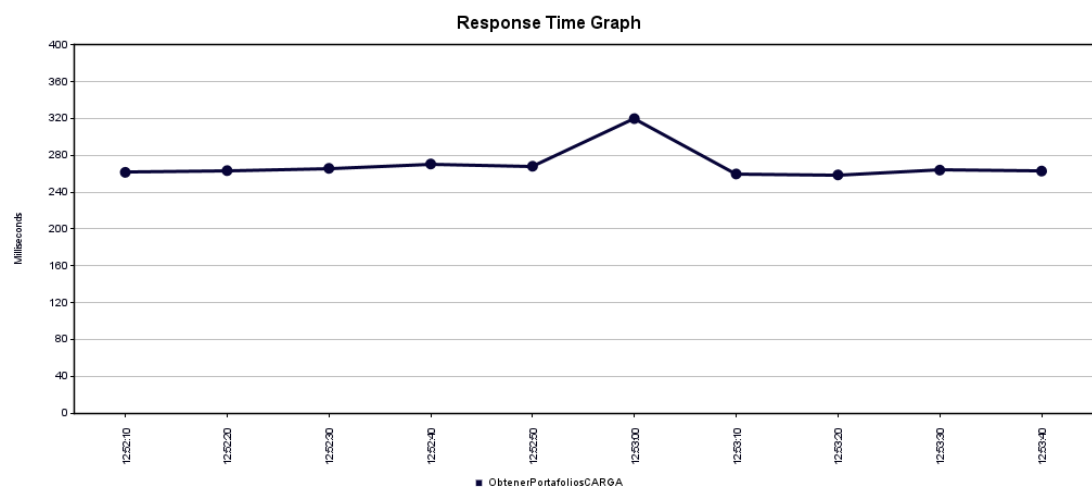


Figura 90. Tiempo de respuesta del segundo servicio en un lapso de tiempo.



Figura 91. Uso de CPU con el segundo servicio en ejecución.

c. CrearXML

La prueba de carga con 37 usuarios simultáneos brinda los siguientes resultados con el tercer método GET para generar el XML.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
CrearXMLCA...	37	1237	0	1617	153.57	0.00%	3.4/sec	0.87	0.55	266.0
TOTAL	37	1237	0	1617	153.57	0.00%	3.4/sec	0.87	0.55	266.0

Figura 92. Resultados de la prueba del tercer servicio web.

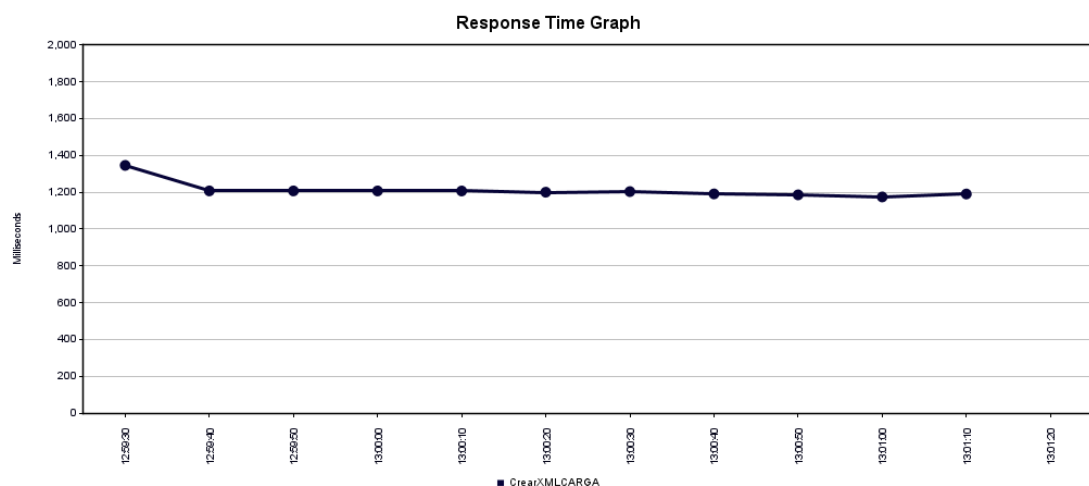


Figura 93. Tiempo de respuesta del tercer servicio en un lapso de tiempo.

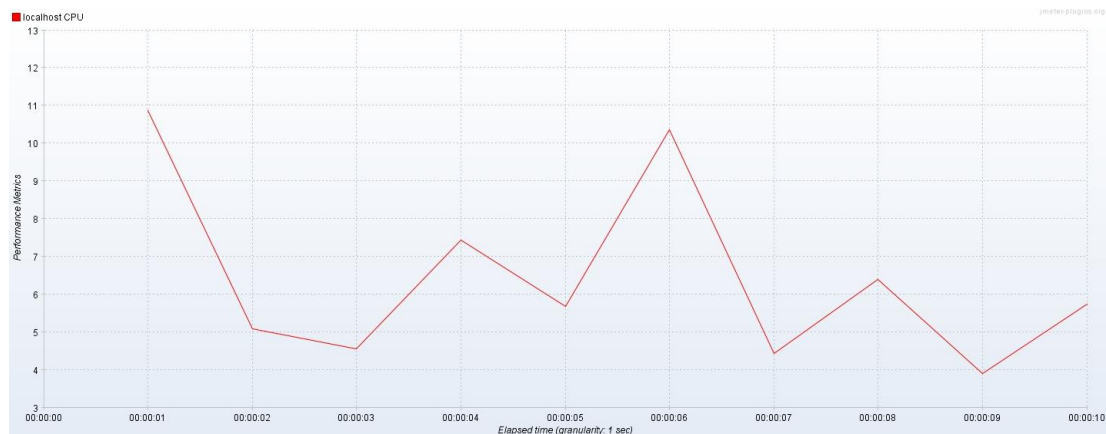


Figura 94. Uso de CPU con el tercer servicio en ejecución.

d. RecibirXMLSri

La prueba de carga con 37 usuarios simultáneos brinda los siguientes resultados con el cuarto método GET para obtener el resultado de recepción del archivo xml en el SRI.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
RecibirXMLSr...	37	986	0	2005	388.16	0.00%	3.5/sec	0.93	0.65	269.0
TOTAL	37	986	0	2005	388.16	0.00%	3.5/sec	0.93	0.65	269.0

Figura 95. Resultados de la prueba del cuarto servicio web.

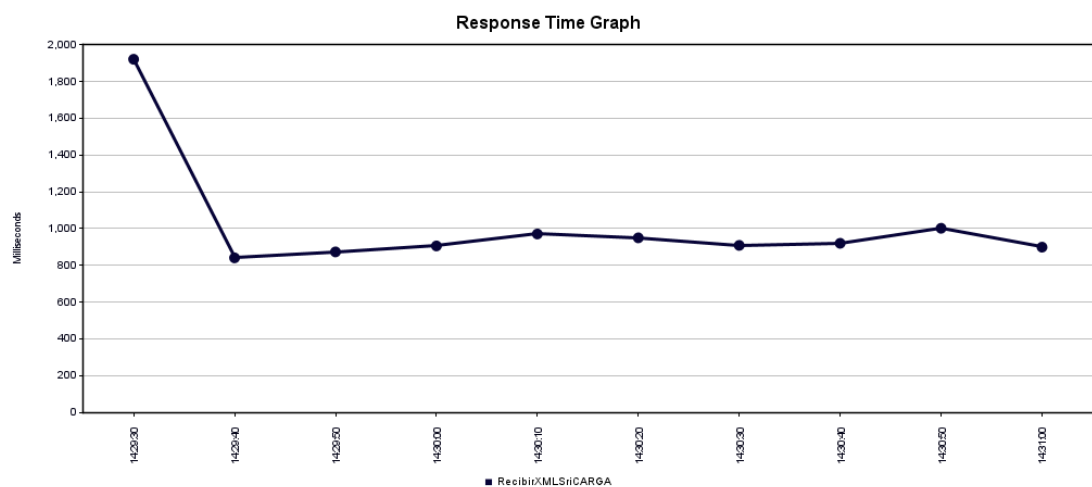


Figura 96. Tiempo de respuesta del cuarto servicio en un lapso de tiempo.

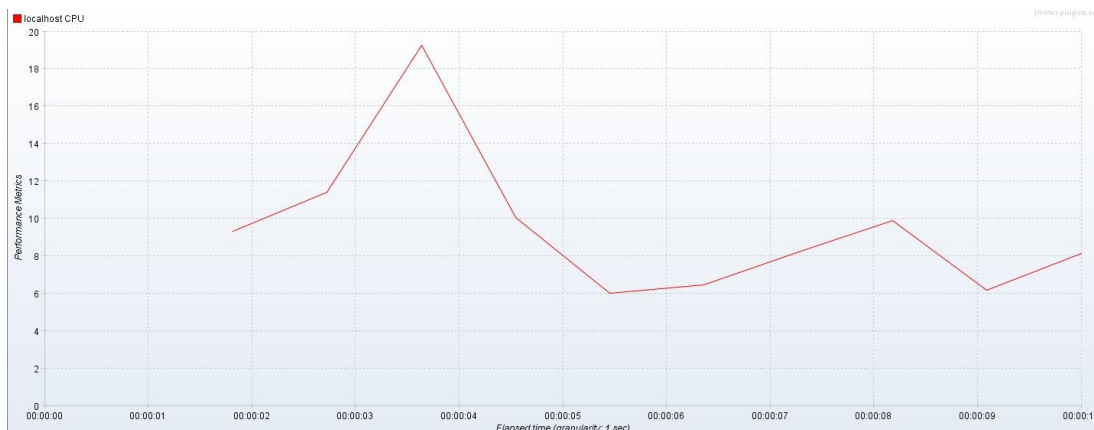


Figura 97. Uso de CPU con el cuarto servicio en ejecución.

e. AutorizarXMLSri

La prueba de carga con 37 usuarios simultáneos brinda los siguientes resultados con el quinto método GET para obtener el resultado de autorización del archivo xml en el SRI.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
AutorizarXML...	37	695	0	1319	239.98	0.00%	3.6/sec	2.49	0.66	710.0
TOTAL	37	695	0	1319	239.98	0.00%	3.6/sec	2.49	0.66	710.0

Figura 98. Resultados de la prueba del quinto servicio web.

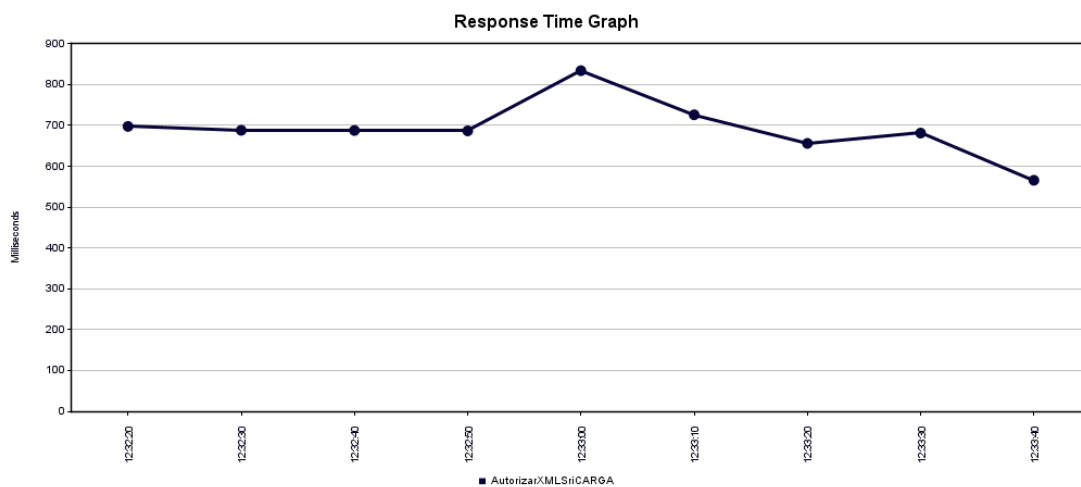


Figura 99. Tiempo de respuesta del quinto servicio en un lapso de tiempo.

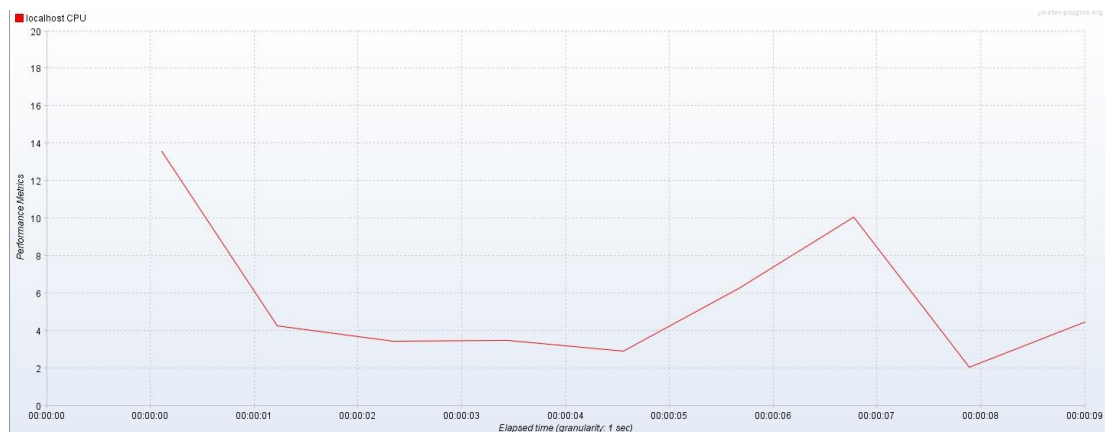


Figura 100. Uso de CPU con el quinto servicio en ejecución.

3.3.3 Pruebas de Estrés.

a. ObtenerAlmacenesSociedad

La prueba de estrés con 74 usuarios simultáneos brinda los siguientes resultados con el primer método POST para obtener los almacenes de la sociedad o usuario.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
ObtenerAlma...	74	252	0	278	8.76	0.00%	7.3/sec	12.76	2.09	1786.0
TOTAL	74	252	0	278	8.76	0.00%	7.3/sec	12.76	2.09	1786.0

Figura 101. Resultados de la prueba del primer servicio web.

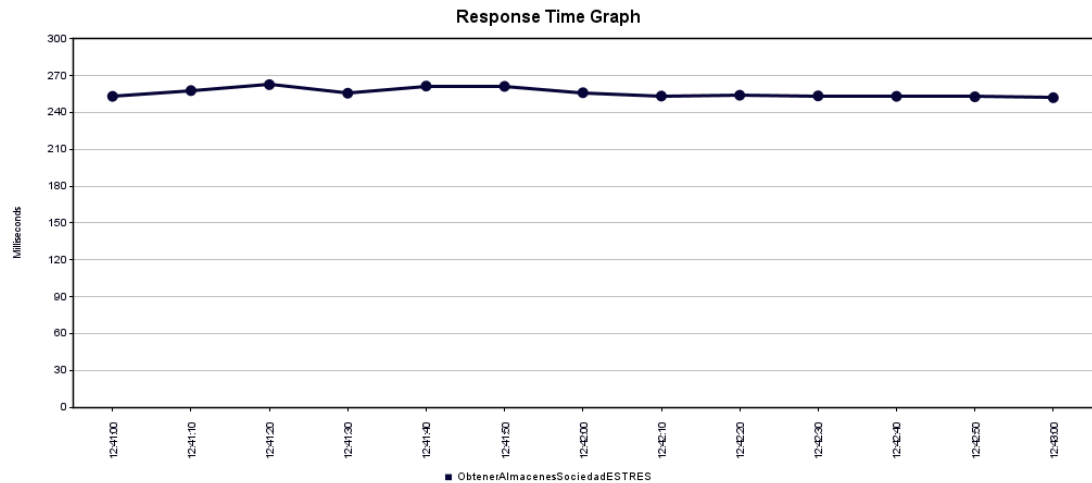


Figura 102. Tiempo de respuesta del primer servicio en un lapso de tiempo.

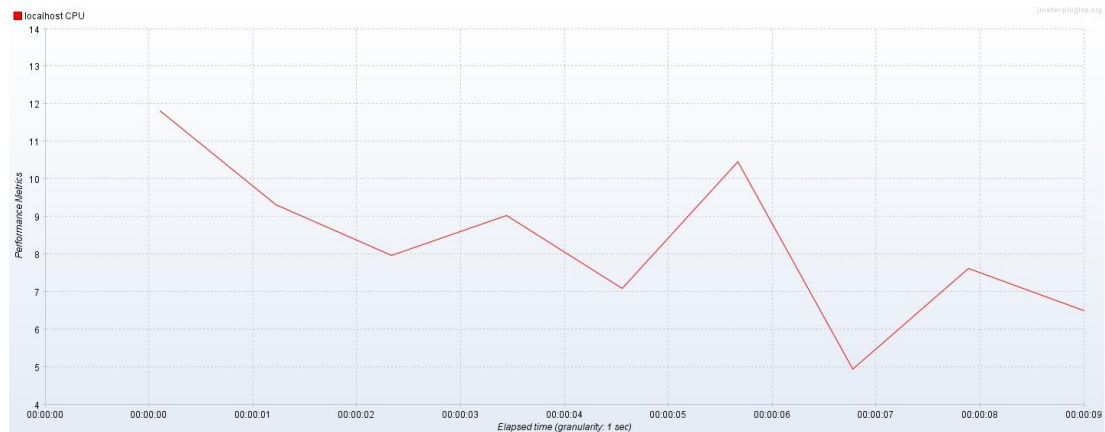


Figura 103. Uso de CPU con el primer servicio en ejecución.

b. ObtenerPortafolios

La prueba de estrés con 74 usuarios simultáneos brinda los siguientes resultados con un segundo método POST para obtener los portafolios.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
ObtenerPorta...	74	284	0	595	79.90	0.00%	7.3/sec	115.40	2.06	16153.0
TOTAL	74	284	0	595	79.90	0.00%	7.3/sec	115.40	2.06	16153.0

Figura 104. Resultados de la prueba del segundo servicio web.

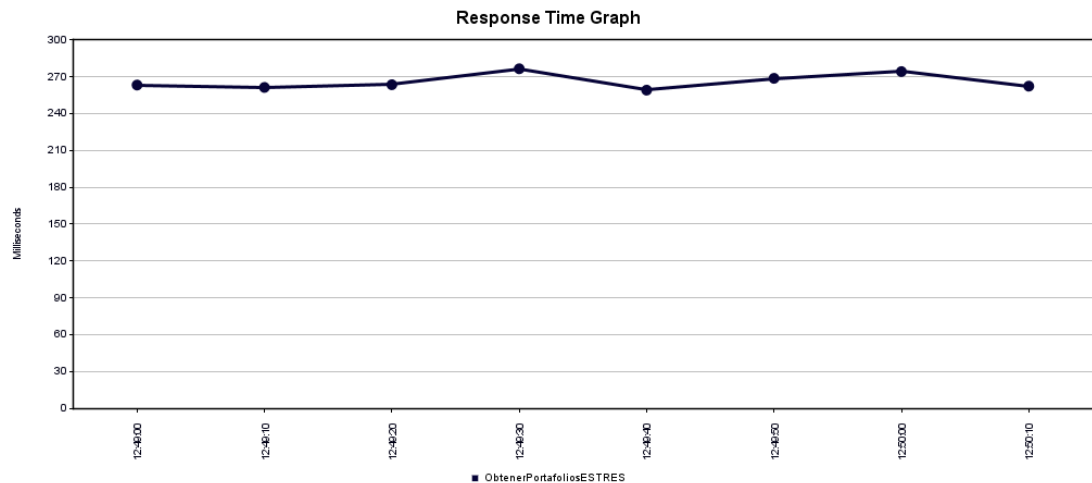


Figura 105. Tiempo de respuesta del segundo servicio en un lapso de tiempo.

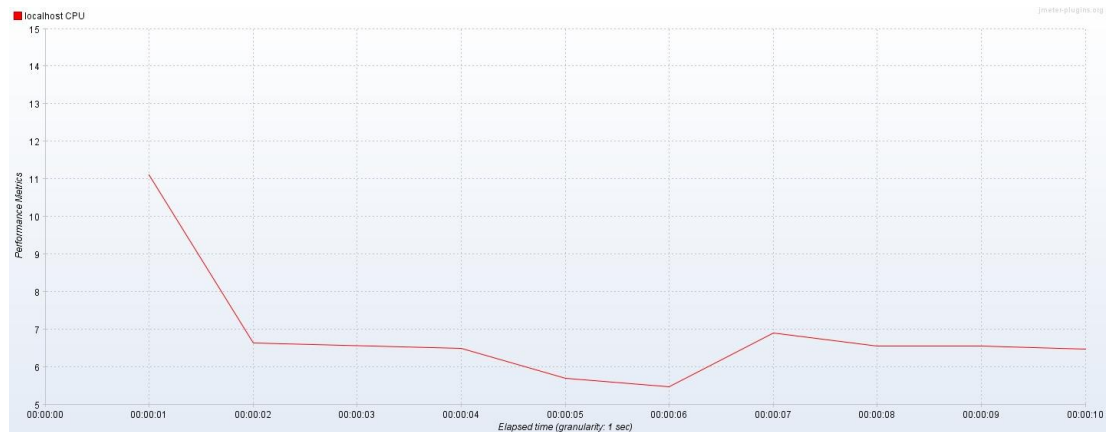


Figura 106. Uso de CPU con el segundo servicio en ejecución.

c. CrearXML

La prueba de estrés con 74 usuarios simultáneos brinda los siguientes resultados con el tercer método GET para generar el XML.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
CrearXMLEST...	74	1272	0	2312	264.14	0.00%	6.7/sec	1.75	1.10	266.0
TOTAL	74	1272	0	2312	264.14	0.00%	6.7/sec	1.75	1.10	266.0

Figura 107. Resultados de la prueba del tercer servicio web.

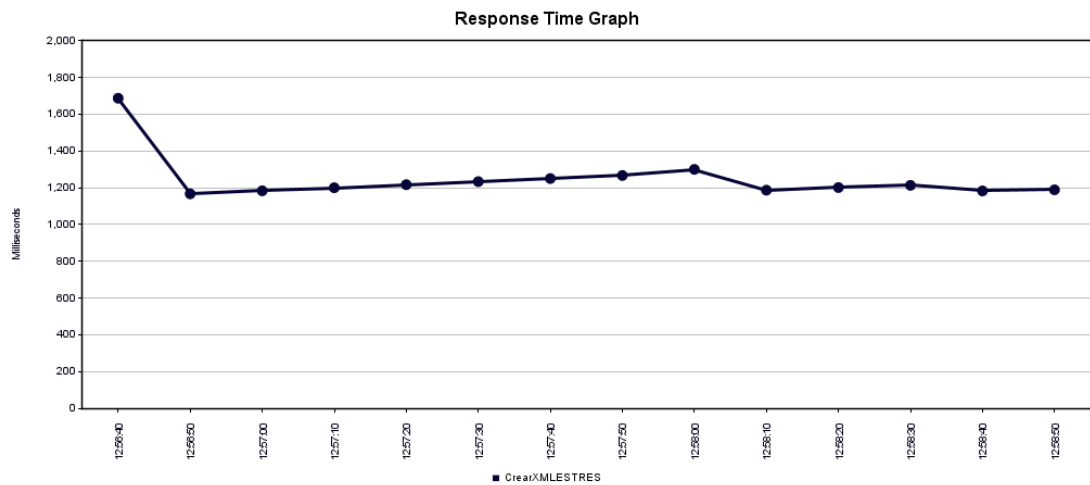


Figura 108. Tiempo de respuesta del tercer servicio en un lapso de tiempo.

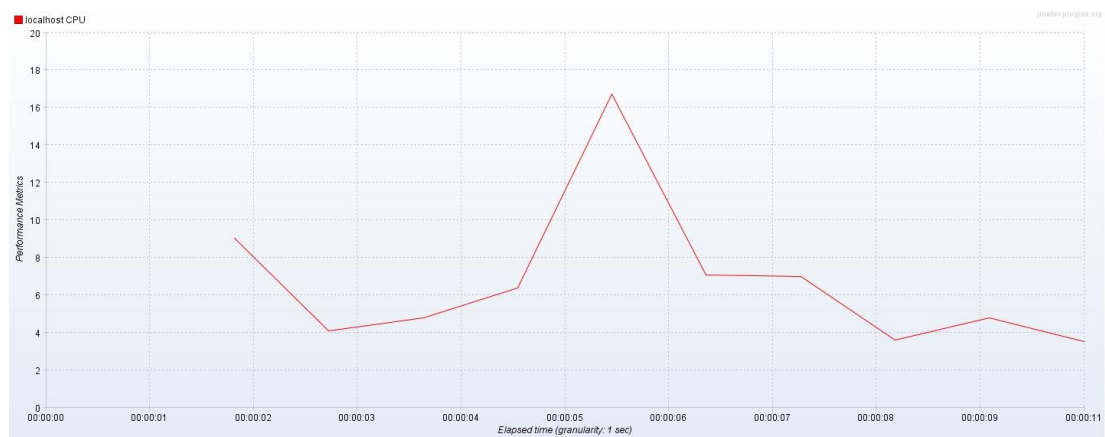


Figura 109. Uso de CPU con el tercer servicio en ejecución.

d. RecibirXMLSri

La prueba de estrés con 74 usuarios simultáneos brinda los siguientes resultados con el cuarto método GET para obtener el resultado de recepción del archivo xml en el SRI.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
RecibirXMLSr...	74	1048	0	2182	429.42	0.00%	6.9/sec	1.82	1.27	269.0
TOTAL	74	1048	0	2182	429.42	0.00%	6.9/sec	1.82	1.27	269.0

Figura 110. Resultados de la prueba del cuarto servicio web.

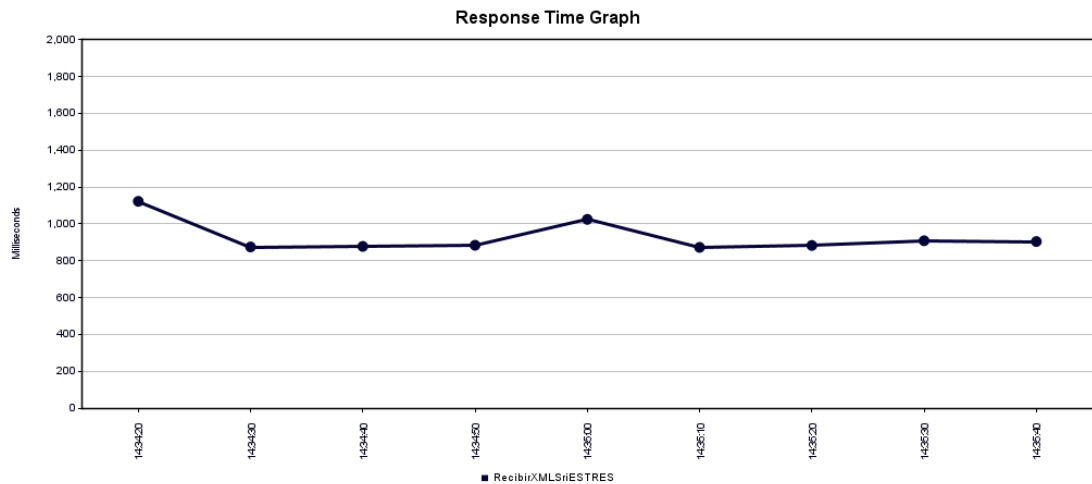


Figura 111. Tiempo de respuesta del cuarto servicio en un lapso de tiempo.

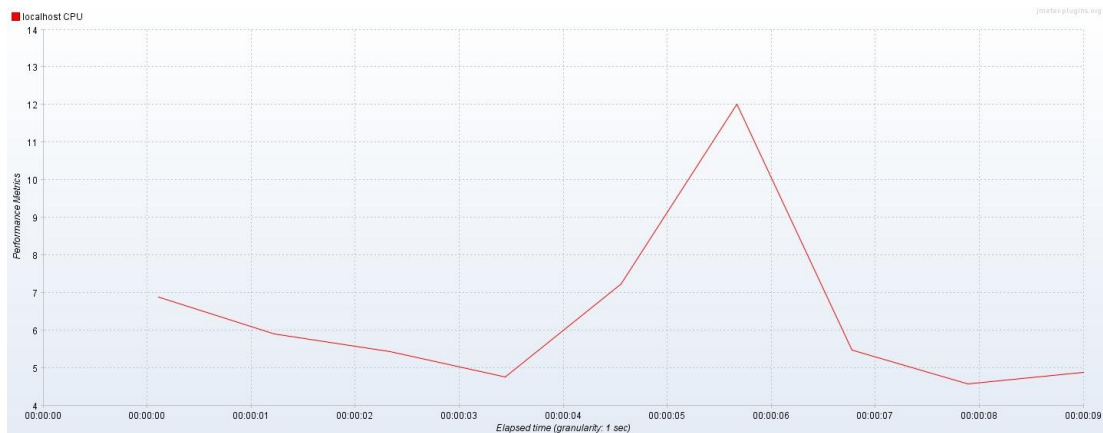


Figura 112. Uso de CPU con el cuarto servicio en ejecución.

e. AutorizarXMLSri

La prueba de estrés con 74 usuarios simultáneos brinda los siguientes resultados con el quinto método GET para obtener el resultado de autorización del archivo xml en el SRI.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/...	Sent KB/sec	Avg. Bytes
AutorizarXML...	74	695	0	1470	274.36	0.00%	6.8/sec	4.75	1.26	710.0
TOTAL	74	695	0	1470	274.36	0.00%	6.8/sec	4.75	1.26	710.0

Figura 113. Resultados de la prueba del quinto servicio web.

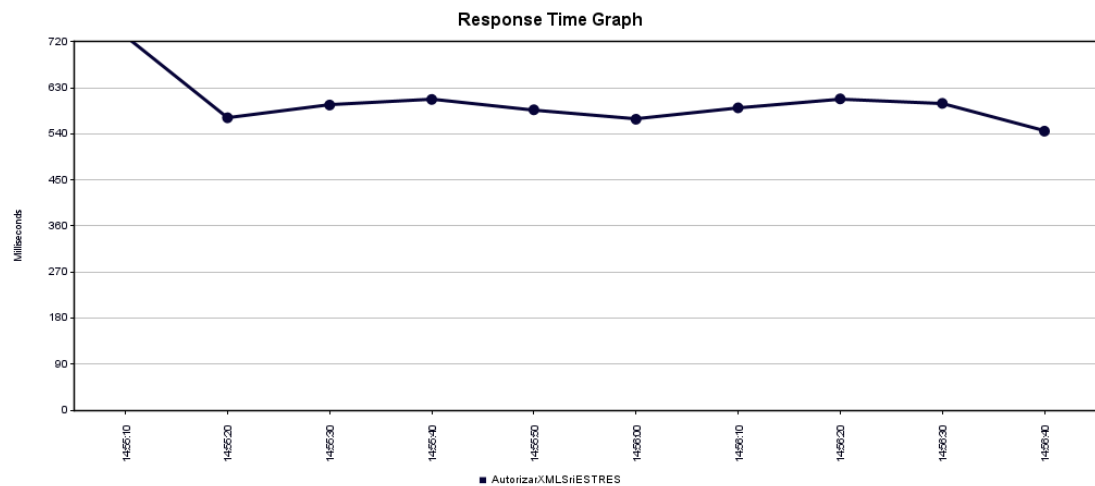


Figura 114. Tiempo de respuesta del quinto servicio en un lapso de tiempo.

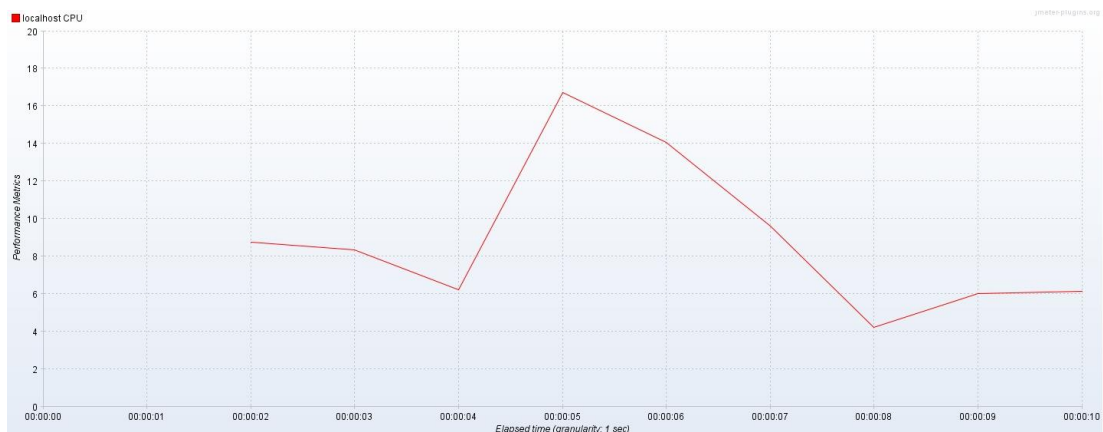


Figura 115. Uso de CPU con el quinto servicio en ejecución.

3.3.4. Análisis de los resultados de la evaluación.

Luego de haber realizado las pruebas de carga y estrés se obtuvo la siguiente tabla para evaluar los resultados obtenidos a partir de las pruebas.

Tabla 16. Evaluación de resultados.

Web Service	Pruebas de Carga			Pruebas de Estrés		
	Tiempo de respuesta promedio (ms)	Uso de CPU	Tasa de Error	Tiempo de respuesta promedio (ms)	Uso de CPU	Tasa de Error
ObtenerAlmacenesSociedad	291	13 %	0.00%	252	12 %	0.00%
ObtenerPortafolios	261	16 %	0.00%	284	11 %	0.00%
CrearXML	1237	11 %	0.00%	1272	17 %	0.00%
RecibirXMLSri	986	18 %	0.00%	1048	12 %	0.00%
AutorizarXMLSri	695	13 %	0.00%	695	17 %	0.00%
PROMEDIO	694	14.2 %	0.00%	710.2	13.8 %	0.00%

Resultados de las Pruebas de Carga: Los servicios web demostraron una sólida capacidad para manejar la carga prevista de 37 solicitudes por segundo. Los tiempos de respuesta se mantuvieron dentro de los límites aceptables, y el sistema mostró una utilización eficiente de los recursos de la CPU. La tasa de error fue mínima, indicando una alta confiabilidad bajo condiciones de carga normal.

Resultados de las Pruebas de Estrés: A medida que la carga se incrementaba hasta 74 solicitudes por segundo, el sistema continuó manejando las solicitudes sin experimentar una degradación significativa en el rendimiento. Aunque se observó un ligero aumento en los tiempos de respuesta, estos se mantuvieron en un rango aceptable, y la tasa de error siguió siendo baja. El sistema demostró una notable capacidad de recuperación y adaptación bajo condiciones de estrés.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

En el presente trabajo se planteó el objetivo principal de implantar el módulo de Facturación Electrónica al sistema “aTiendo” utilizando un modelo de desarrollo de software incremental y evaluando el rendimiento bajo la normativa ISO/IEC 25010 en la empresa Plasticaucho Industrial S.A; se llega a las siguientes conclusiones:

- El resultado principal de la implantación de el módulo de facturación electrónica al sistema “aTiendo” favorece de gran manera a los usuarios que todavía no cuentan con este reglamento obligatorio dentro de sus puntos de venta y que desean trabajar aplicando esta modalidad, además de poder enviar comprobantes por correo.
- En la etapa de recolección de información mediante encuestas se logra evidenciar que solamente el 18.92% nunca realiza un proceso manual en cuanto a la gestión de inventarios y facturación electrónica, mientras que el 81.08% restante sí realiza un proceso manual de alguna forma en su gestión diaria. Con estos resultados se mantuvo una base y contexto claro para arrancar con el desarrollo del módulo de facturación electrónica.
- Utilizar frameworks frontend como Angular, hace que el proceso de desarrollo de software sea adaptado eficaz y eficientemente, además de brindar diferentes complementos de desarrollo de aplicaciones responsivas, lo que brinda al usuario una vista también desde un dispositivo móvil.

Por otro lado, aplicaciones como ASP.Net permiten construir sistemas de servidores backend en los cuales los servicios web sean accesibles en su totalidad y en cualquier momento bajo ciertas circunstancias; para que estos se puedan consumir ágilmente en un framework de front.

- Para la generación de las tablas de base de datos se utilizó MySQL, con el gestor phpMyAdmin brindado en el sistema de cPanel, tecnología que se utiliza en la empresa en donde se encuentra desarrollada la aplicación “aTiendo”. En esta base de datos se generó un esquema con distintas tablas e información almacenada para asegurar la correcta operatividad al momento de utilizar el módulo implantado y el sistema en general.
- Para el manejo de credenciales del certificado digital se manejó un método de encriptación robusto llamado AES, el mismo es conocido por ser empleado a nivel mundial en sistemas de gran alcance. Además, este método de encriptación es muy utilizado dentro de la empresa para otros sistemas.
- El uso de la metodología de desarrollo de software incremental permitió un avance continuo en el proyecto, esto debido a que a través de cada uno de los incrementos se recibieron retroalimentaciones a base de reuniones con el cliente; la realización de prototipos y despliegue de avances fueron factores clave para abordar el desarrollo ya que mediante esto se evaluó el incremento al final.
- La evaluación del rendimiento es una actividad importante en el proceso de desarrollo de software. Un producto software con un buen rendimiento es aquel que puede satisfacer las necesidades de los usuarios en términos de velocidad, fiabilidad, escalabilidad y eficiencia; de tal manera que al culminar con el despliegue del módulo se procedió con la realización de pruebas de rendimientos, las mismas que ayudaron a saber si ciertos servicios web no tenían sobrecarga de datos y así poder corregir y optimizar lo que sea necesario.
- En el análisis de los resultados de la evaluación se observa un buen rango de tiempos de respuesta, presentando un promedio general de 694 milisegundos, siendo un tiempo muy bueno teniendo en cuenta que existe un servicio web que tarda más de un segundo en procesarse.
- La norma ISO/IEC 25010 brindó un alcance ideal para realizar las pruebas de rendimiento sobre el módulo de facturación electrónica. Esta norma permitió

establecer parámetros de prueba con la finalidad de realizar una correcta evaluación. Hay que tener en cuenta que la evaluación del rendimiento es una tarea compleja que requiere una planificación cuidadosa. Sin embargo, la evaluación del rendimiento puede proporcionar información valiosa que puede ayudar a mejorar el rendimiento de los productos software.

4.2 Recomendaciones

- Se recomienda realizar la implementación en los 40 clientes en un lapso de 6 meses para poder analizar factores de adaptación del usuario al nuevo módulo de facturación electrónica desarrollado.
- Revisar documentación técnica acerca del ingreso de notas de débito y crédito hacia el Servicio de Rentas Internas con la finalidad de agregar un módulo de devoluciones a la gestión dentro del sistema.
- Implementar técnicas de CRM dentro de “aTiendo” con la finalidad de recabar, reunir y centralizar la información de clientes dentro del sistema y poder brindar una mejor atención al cliente.
- Se sugiere adaptar tecnologías de desarrollo de aplicaciones móviles para el sistema “aTiendo” que cuente con el módulo offline de la gestión.
- Validar el proceso de facturación electrónica, al igual que el reglamento y variables que afectan a la creación del archivo xml, esto debido a que los procesos se pueden ir actualizando conforme pasa el tiempo.
- Adaptar algoritmos de seguridad más robustos y aplicar en todo el sistema. Esto favorecerá directamente a la gestión de datos, sean estos datos personales o datos públicos.
- Se debe establecer un flujo de optimización basado en el feedback de los usuarios. Los comentarios y sugerencias recopilados a través de distintos instrumentos pueden resultar útiles para editar y optimizar la aplicación web.

MATERIALES DE REFERENCIA

- [1] S. Hussein, «Review of Web Service Technologies: REST over SOAP,» Journal of Al-Qadisiyah for Computer Science and Mathematics., vol. 12, nº 4, pp. 18-30, 2020.
- [2] SRI, «Servicio de Rentas Internas,» Gobierno del Ecuador, 24 Julio 2022. [En línea]. Available: <https://www.sri.gob.ec/herramienta-de-comprobantes-electronicos>. [Último acceso: 18 Septiembre 2023].
- [3] A. García, Firma digital, certificado electrónico y factura electrónica., Madrid, España: RA-MA Editorial, 2020.
- [4] MyTCOE, Performance Test Tools: Introduction to JMeter, Putrajaya, Malasia: MAMPU, 2020.
- [5] R. Pressman, Ingeniería del Software: Un enfoque práctico., Connecticut, EE.UU: Mc.Graw Hill, 2010.
- [6] Y. López y J. Cáceres, «Creación de un FRAMEWORK de desarrollo de aplicacionesweb, basado en JAVASCRIPT, HTML y CSS.,» de CIINATIC, Cúcuta, Colombia, 2019.
- [7] Microsoft, «Desarrollo de aplicaciones .NET,» 15 Mayo 2023. [En línea]. Available: <https://visualstudio.microsoft.com/es/vs/features/net-development/>. [Último acceso: 19 Septiembre 19].
- [8] D. Navarro, «Apuntes de estimación de la fiabilidad de consistencia interna de los ítems de un instrumento de medida,» Universidad de Valencia, Valencia, España, 2022.
- [9] V. Bardales y W. Rojas, «El sistema de Facturación Electrónica y el cumplimiento de las obligaciones tributarias de las empresas del sector comercio del Distrito de Calleria, AÑO 2021,» Universidad Nacional de Ucayal, Pucallpa, Perú, 2022.
- [10] H. Intriago, K. Loor y J. Cedeño, «Facturación electrónica como obligación tributaria para los contribuyentes del régimen general en Portoviejo.,» Ciencia Latina Revista Científica Multidisciplinar, vol. 6, nº 6, pp. 13962-13976, 17 Agosto 2022.
- [11] Abitmedia S.A.S., «FacturaSoft,» Abitmedia S.A.S., 31 Julio 2023. [En línea]. Available: <https://facturasoft.com>. [Último acceso: 2023 Septiembre 18].
- [12] A. Beltrán, «Implementación de facturación electrónica en Colombia,» Universidad Católica de Colombia, Bogotá, Colombia, 2018.

- [13] Contifico, «Facturación electrónica en el Ecuador: Una guía práctica para implementar este sistema.,» Siigo | Contifico, Guayaquil, Ecuador, 2022.
- [14] V. Valverde, N. Portalanza y P. Mora, «Análisis descriptivo de base de datos relaciones y no relacional.,» ESPOCH, Riobamba, Ecuador, 2019.
- [15] L. Valencia, Introducción a SQL., Sevilla, España: Research Group of Natural Computing, 2021.
- [16] ISO, «ISO 25000: Calidad de software y datos.,» ISO/IEC 25010, 2 Julio 2022. [En línea]. Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010>. [Último acceso: 20 Septiembre 2023].
- [17] S. Hernández y D. Duana, «Técnicas e instrumentos de recolección de datos.,» UAEH, vol. 9, nº 17, pp. 51-53, 2020.
- [18] E. G. Maida y J. Pacienza, Metodologías de desarrollo de software, Buenos Aires, Argentina: Universidad Católica de Argentina, 2015.

ANEXOS

Anexo A. Encuesta para obtener nuevos requerimientos aplicada a los usuarios del sistema “aTiendo”.

1. ¿Qué tan importante es para usted que un sistema de facturación electrónica y gestión de inventario tenga las siguientes características o funcionalidades?
 - Generar facturas electrónicas válidas (1 = Muy poco importante, 5 = Muy importante)
 - Gestionar el inventario de manera eficiente (1 = Muy poco importante, 5 = Muy importante)
 - Proporcionar informes útiles para la toma de decisiones (1 = Muy poco importante, 5 = Muy importante)
 - Ser confiable (1 = Muy poco importante, 5 = Muy importante)
2. ¿Cómo utiliza actualmente la facturación electrónica y la gestión de inventario?
 - ¿La realiza manualmente? (1 = Siempre, 5 = Nunca)
 - ¿Qué sistema de software utiliza actualmente?
3. ¿Qué desafíos o problemas enfrenta con la facturación electrónica y la gestión de inventario actualmente?
 - ¿Cuánto tiempo lleva completar una tarea en el sistema actual?
 - 0 – 1 min
 - 1- 5 min
 - 5 – 10 min
 - 10 – 20 min
 - Más de 20 min
 - ¿Cuál es el mayor problema que enfrenta con respecto a la facturación?
4. ¿Qué tipo de empresa tiene?
 - ¿Es una empresa pequeña, mediana o grande?
5. ¿Cuál es su volumen de facturación diario?
 - 0 – 10 facturas
 - 10 – 50 facturas
 - 50 – 100 facturas
 - 100 – 200 facturas

➤ Más de 200 facturas

6. ¿Qué tipo de productos o servicios vende?

- ☐ Primera necesidad
- ☐ Tecnología
- ☐ Calzado
- ☐ Ropa
- ☐ Otros:

7. ¿Tiene algún comentario o sugerencia adicional?

Comentarios adicionales:

Anexo B. Referencias Asp. NET

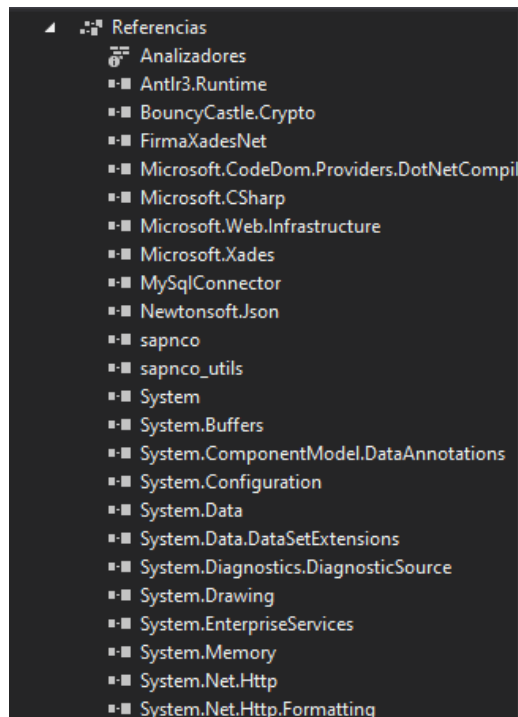


Figura B1. Referencias Backend.

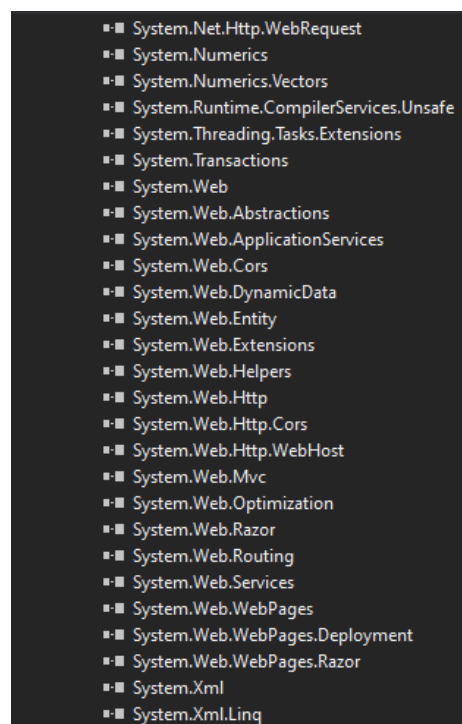


Figura B2. Referencias Backend.

```

<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Antlr" version="3.5.0.2" targetFramework="net461" />
  <package id="bootstrap" version="3.4.1" targetFramework="net461" />
  <package id="BouncyCastle" version="1.7.0" targetFramework="net461" />
  <package id="FirmaxAdesNet" version="2.0.1.1" targetFramework="net461" />
  <package id="jQuery" version="3.4.1" targetFramework="net461" />
  <package id="jQuery.Validation" version="1.17.0" targetFramework="net461" />
  <package id="Microsoft.AspNet.Cors" version="5.2.9" targetFramework="net461" />
  <package id="Microsoft.AspNet.Mvc" version="5.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.Mvc.es" version="5.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.Razor" version="3.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.Razor.es" version="3.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.Web.Optimization" version="1.1.3" targetFramework="net461" />
  <package id="Microsoft.AspNet.Web.Optimization.es" version="1.1.3" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi" version="5.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.Client" version="5.2.9" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.Client.es" version="5.2.9" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.Core" version="5.2.9" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.Core.es" version="5.2.9" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.Cores" version="5.2.9" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.WebHost" version="5.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebApi.WebHost.es" version="5.2.7" targetFramework="net461" />
  <package id="Microsoft.AspNet.WebPages" version="3.2.7" targetFramework="net461" />
  <package id="Microsoft.CodeDom.Providers.DotNetCompilerPlatform" version="2.0.1" targetFramework="net461" />
  <package id="Microsoft.JQuery.Unobtrusive.Validation" version="3.2.11" targetFramework="net461" />
  <package id="Microsoft.Web.Infrastructure" version="1.0.0.0" targetFramework="net461" />
  <package id="Modernizr" version="2.8.3" targetFramework="net461" />
  <package id="MySQLConnector" version="2.1.11" targetFramework="net461" />
  <package id="Newtonsoft.Json" version="12.0.2" targetFramework="net461" />
  <package id="System.Buffers" version="4.5.1" targetFramework="net461" />
  <package id="System.Diagnostics.DiagnosticSource" version="5.0.1" targetFramework="net461" />
  <package id="System.Memory" version="4.5.4" targetFramework="net461" />
  <package id="System.Numerics.Vectors" version="4.5.0" targetFramework="net461" />
  <package id="System.Runtime.CompilerServices.Unsafe" version="5.0.0" targetFramework="net461" />
  <package id="System.Threading.Tasks.Extensions" version="4.3.0" targetFramework="net461" />
  <package id="WebGrease" version="1.6.0" targetFramework="net461" />
</packages>

```

Figura B3. Versiones de referencias.

Anexo C. Entidades del módulo de facturación electrónica.

```
1  export interface Almacenes {
2      codigoError: string;
3      descripcionError: string;
4      lstAlmacenes: AlmacenesEntity[];
5  }
6
7  You, hace 11 meses | 1 author (You)
8  export interface AlmacenesEntity {
9      idAlmacen : string;
10     sociedad_id : string;
11     nombresociedad: string;
12     idfiscal_sociedad?: string;
13     direccion : string;
14     telefono : string;
15     nombre_almacen?: string;
16     codigo : string;
17     pto_emision : string;
18 }
```

Figura C1. Entidad de almacenes en Angular.

```
0 referencias
public AlmacenesEntity(string idAlmacen, string sociedad_id, string nombre_almacen, string nombresociedad, string idfiscal_sociedad, string direccion,
{
    this.idAlmacen = idAlmacen;
    this.sociedad_id = sociedad_id;
    this.nombre_almacen = nombre_almacen;
    this.nombresociedad = nombresociedad;
    this.idfiscal_sociedad = idfiscal_sociedad;
    this.direccion = direccion;
    this.telefono = telefono;
    this.codigo = codigo;
    this.pto_emision = pto_emision;
    this.es_plasticaucho = es_plasticaucho;
    this.es_sincronizado = es_sincronizado;
}
```

Figura C2. Entidad de almacenes en .Net.

```

1 export interface Movimientos {
2   codigoError: string;
3   descripcionError: string;
4   lstMovimientos: MovimientosEntity[];
5 }
6
7 export interface MovimientosEntity {
8   id: string;
9   tercero_id?: string;
10  tipo_id: string;
11  tipo_emision_cod: string;
12  estado_fact_id: string;
13  tipo_comprb_id: string;
14  tipo_comprb?: string;
15  almacen_id: string;
16  estab?: string;
17  cod_doc: string;
18  secuencial: string;
19  clave_acceso?: string;
20  total_si?: string;
21  total_desc?: string;
22  total_imp?: string;
23  detalle_pago?: string;
24  propina?: string;
25  proveedor_id?: string;
26  proveedor?: string;
27  comp_venta?: string;
28  sustento_id?: string;
29  sustento?: string;
30  autorizacion_venta?: string;
31  fecha_emision?: string;
32  importe_total?: string;
33  valor_rete_iva?: string;
34  valor_rete_renta?: string;
35  camp_ad1?: string;
36  camp_ad2?: string;
37  updated_at?: string;
38  tercero?: string;
39  tipo_comprb_cod?: string;
40  id_fiscal_soc?: string;
41  tipo_ambiente?: string;
42  pto_emision?: string;
43  url_factura?: string;
44  comprobante_compra_id?: string;
45  nroFactura?: string;
46 }

```

Figura C3. Entidad de movimientos en Angular.

```

public MovimientosEntity(string id, string tercero_id, string tipo_id, string tipo_emision_cod, string estado_fact_id, string tipo_comprb_id, string id_fiscal_soc, string tipo_comprb, string almacen_id, string estab, string pto_emision, string cod_doc, string
{
    this.id = id;
    this.tercero_id = tercero_id;
    this.tipo_id = tipo_id;
    this.tipo_emision_cod = tipo_emision_cod;
    this.estado_fact_id = estado_fact_id;
    this.tipo_comprb_id = tipo_comprb_id;
    this.id_fiscal_soc = id_fiscal_soc;
    this.tipo_comprb = tipo_comprb;
    this.almacen_id = almacen_id;
    this.estab = estab;
    this.pto_emision = pto_emision;
    this.cod_doc = cod_doc;
    this.secuencial = secuencial;
    this.clave_acceso = clave_acceso;
    this.fecha_emision = fecha_emision;
    this.periodo_fiscal = periodo_fiscal;
    this.total_si = total_si;
    this.total_desc = total_desc;
    this.total_imp = total_imp;
    this.propina = propina;
    this.importe_total = importe_total;
    this.proveedor_id = proveedor_id;
    this.tipo_ambiente = tipo_ambiente;
    this.tipo_comprb_cod = tipo_comprb_cod;
    this.url_factura = url_factura;
    this.proveedor = proveedor;
    this.sustento_id = sustento_id;
    this.sustento = sustento;
    this.comp_venta = comp_venta;
    this.autorizacion_venta = autorizacion_venta;
    this.moneda = moneda;
    this.valor_rete_iva = valor_rete_iva;
    this.valor_rete_renta = valor_rete_renta;
    this.camp_ad1 = camp_ad1;
    this.camp_ad2 = camp_ad2;
    this.tercero = tercero;
    this.es_sincronizado = es_sincronizado;
    this.created_at = created_at;
    this.updated_at = updated_at;
    this.detalle_pago = detalle_pago;
    this.comprobante_compra_id = comprobante_compra_id;
}

```

Figura C4. Entidad de movimientos en .Net.

```

12 export interface InventarioEntity {
13   categoria_id : string;
14   categoria : string;
15   linea_id? : string;
16   linea : string;
17   modelo_id? : string;
18   modelo : string;
19   marca_id : string;
20   marca : string;
21   modelo_producto_id : string;
22   modelo_producto? : string;
23   idProducto : string;
24   Producto : string;
25   productoExistente?: boolean;
26   id : string;
27   dInventario : string;
28   producto_id : string;
29   tarifa_ice_iva?: string;
30   tarifa_ice_iva_id?: string;
31   almacen_id : string;
32   producto_nombre? : string;
33   almacen : string;
34   stock? : string;
35   etiquetas?: string;
36   stock_optimo : string;
37   fav : string;
38   color : string;
39   costo?: string;
40   cantidad?: string;
41   pvp1? : string;
42   pvp2? : string;
43   pvp_sugerido? : string;
44   cod_principal? : string;
45   cod_secundario? : string;
46   unidad_medida? : string;
47   url_image?: string;
48   talla? : string;
49   tarifa_ice_iva1?: string;
50   tarifa_ice_iva_id1?: string;
51   genero?: string;
52   atributo?: string;
53 }

```

Figura C5. Entidad de inventarios en Angular.

```

1 public InventarioEntity(string categoria_id, string categoria, string linea_id, string linea, string modelo_id, string marca_id, string marca, string atributo, str
2 {
3   this.categoria_id = categoria_id;
4   this.categoria = categoria;
5   this.linea_id = linea_id;
6   this.linea = linea;
7   this.modelo_id = modelo_id;
8   this.marca_id = marca_id;
9   this.marca = marca;
10  this.atributo = atributo;
11  this.atributo_id = atributo_id;
12  this.genero = genero;
13  this.genero_id = genero_id;
14  this.modelo_producto_id = modelo_producto_id;
15  this.idProducto = idProducto;
16  Producto = producto;
17  this.id = id;
18  this.idInventario = idInventario;
19  this.producto_id = producto_id;
20  this.almacen_id = almacen_id;
21  this.tarifa_ice_iva = tarifa_ice_iva;
22  this.tarifa_ice_iva_id = tarifa_ice_iva_id;
23  this.tarifa_ice_iva_id1 = tarifa_ice_iva_id1;
24  this.tarifa_ice_iva1 = tarifa_ice_iva1;
25  this.almacen = almacen;
26  this.stock = stock;
27  this.stock_optimo = stock_optimo;
28  this.es_sincronizado = es_sincronizado;
29  this.fav = fav;
30  this.color_id = color_id;
31  this.color = color;
32  this.modelo = modelo;
33  this.etiquetas = etiquetas;
34  this.modelo_producto = modelo_producto;
35  this.producto_nombre = producto_nombre;
36  this.costo = costo;
37  this.pvp1 = pvp1;
38  this.pvp2 = pvp2;
39  this.pvp_sugerido = pvp_sugerido;
40  this.cod_principal = cod_principal;
41  this.cod_secundario = cod_secundario;
42  this.talla = talla;
43  this.unidad_medida = unidad_medida;
44  this.url_image = url_image;
45 }

```

Figura C6. Entidad de inventarios en .Net.

```

1  export interface DetallesMovimiento {
2      codigoError: string;
3      descripcionError: string;
4      lstDetalleMovimientos: DetallesMovimientoEntity[];
5  }
6
7  EdisonLopezEC, el mes pasado | 3 authors (You and others)
8  export interface DetallesMovimientoEntity {
9      id: string;
10     producto_id: string;
11     producto_nombre: string;
12     modelo_producto_nombre?: string;
13     tamano?: string;
14     color?: string;
15     tarifa?: string;
16     pto_emision?: string;
17     inventario_id: string;
18     movimiento_id: string;
19     cod_tarifa?: string;
20     desc_add?: string;
21     cantidad: string;
22     costo: string;
23     unidad_medida?: string;
24     precio: string;
25     url_image?: string;
26     created_at?: string;
27     tipo_movimiento?: string;
28     ice?: string;
29 }

```

Figura C7. Entidad de detalle movimientos en Angular.

```

public DetalleMovimientosEntity(string id, string producto_id, string producto_nombre, string modelo_producto_nombre, string inventario_id, string pto_emision)
{
    this.id = id;
    this.producto_id = producto_id;
    this.producto_nombre = producto_nombre;
    this.modelo_producto_nombre = modelo_producto_nombre;
    this.inventario_id = inventario_id;
    this.pto_emision = pto_emision;
    this.movimiento_id = movimiento_id;
    this.tamano = tamano;
    this.color = color;
    this.cantidad = cantidad;
    this.tarifa = tarifa;
    this.tipo_movimiento = tipo_movimiento;
    this.desc_add = desc_add;
    this.costo = costo;
    this.precio = precio;
    this.url_image = url_image;
    this.created_at = created_at;
    this.update_at = update_at;
    this.unidad_medida = unidad_medida;
    this.cod_tarifa = cod_tarifa;
    this.ice = null;
}

```

Figura C8. Entidad de detalle movimientos en .Net.

```

1  export interface FormasPago {
2      codigoError: string;
3      descripcionError: string;
4      lstFormasPago: FormasPagoEntity[];
5  }
6
7  export interface FormasPagoEntity {
8      id: string;
9      nombre: string;
10     codigo: string;
11     fecha_inicio: string;
12     fecha_fin: string;
13     created_at: string;
14     updated_at: string;
15 }

```

Figura C9. Entidad de formas de pago en Angular.

```

public FormasPagoEntity(string id, string nombre, string codigo, string fecha_inicio, string fecha_fin, string created_at, string updated_at)
{
    this.id = id;
    this.nombre = nombre;
    this.codigo = codigo;
    this.fecha_inicio = fecha_inicio;
    this.fecha_fin = fecha_fin;
    this.created_at = created_at;
    this.updated_at = updated_at;
}

```

Figura C10. Entidad de formas de pago en .Net.

```

1  export interface DetalleImpuestos {
2      codigoError: string;
3      descripcionError: string;
4      lstDetalleImpuestos: DetalleImpuestosEntity[];
5  }
6
7  export interface DetalleImpuestosEntity {
8      id: string;
9      detalle_movimiento_id: string;
10     cod_impuesto: string;
11     porcentaje: string;
12     base_imponible: string;
13     valor: string;
14     created_at: string;
15     updated_at: string;
16     movimiento_id?: string;
17 }

```

Figura C11. Entidad de detalle impuestos en Angular.

```

public DetalleImpuestosEntity(string id, string detalle_movimiento_id, string cod_impuesto, string porcentaje, string base_imponible, string valor, string movimiento_id)
{
    this.id = id;
    this.detalle_movimiento_id = detalle_movimiento_id;
    this.cod_impuesto = cod_impuesto;
    this.porcentaje = porcentaje;
    this.base_imponible = base_imponible;
    this.valor = valor;
    this.movimiento_id = movimiento_id;
    this.created_at = created_at;
    this.updated_at = updated_at;
}

```

Figura C12. Entidad de detalle impuestos en .Net.

```

1  export interface Sociedades {
2      codigoError: string;
3      descripcionError: string;
4      lstSociedades: SociedadesEntity[];
5  }
6
7  EdisonLopezEC, el mes pasado | 2 authors (You and one other)
8  export interface SociedadesEntity {
9      idGrupo: string;
10     idSociedad: string;
11     nombreGrupo?: string;
12     razon_social: string;
13     nombre_comercial: string;
14     id_fiscal: string;
15     id_fiscal_grupo?: string;
16     email: string;
17     telefono: string;
18     password: string;
19     funcion : string;
20     tipo_ambienteid: string;
21     url_certificado?: string;
22     clave_certificado?: string;
23     dir1?: string;
24     direccion?: string;
25     ambiente?: string;
26     email_certificado?: string;
27     pass_certificado?: string;
28     sociedad_pertenece?: string;
29     almacen_personal_id?: string;
30     emite_retencion?: string;
31     obligado_contabilidad?: string;
32     url_logo?: string;
33 }

```

Figura C13. Entidad de sociedades en Angular.

```

public SociedadesEntity(string idSociedad, string idGrupo, string nombreGrupo, string razon_social, string nombre_comercial, string id_fiscal_grupo, string i
{
    this.idSociedad = idSociedad;
    this.idGrupo = idGrupo;
    this.nombreGrupo = nombreGrupo;
    this.razon_social = razon_social;
    this.nombre_comercial = nombre_comercial;
    this.id_fiscal_grupo = id_fiscal_grupo;
    this.id_fiscal = id_fiscal;
    this.password = password;
    this.email = email;
    this.telefono = telefono;
    this.gven = gven;
    this.tipologia = tipologia;
    this.cod_sap = cod_sap;
    this.funcion = funcion;
    this.paginacion = paginacion;
    this.tipo_ambienteId = tipo_ambienteId;
    this.diri = diri;
    this.direccion = direccion;
    this.ambiente = ambiente;
    this.url_certificado = url_certificado;
    this.clave_certificado = clave_certificado;
    this.email_certificado = email_certificado;
    this.pass_certificado = pass_certificado;
    this.sociedad_pertenece = null;
    this.nombre_personal = null;
    this.almacen_personal_id = null;
    this.emite_retencion = null;
    this.obligado_contabilidad = null;
    this.url_logo = null;
}

```

Figura C14. Entidad de sociedades en .Net.

Anexo D. Controladores y servicios web.

```
[EnableCors(origins: "*", headers: "*", methods: "*")]
[RoutePrefix("api/almacenes")]
0 referencias
public class AlmacenesController : ApiController
{
    [HttpGet]
    [Route("ObtenerAlmacenes")]
    0 referencias
    public Almacenes ObtenerAlmacenes()
    {
        return DALAlmacenes.ObtenerAlmacenes();
    }

    [HttpPost]
    [Route("ObtenerAlmacenesSociedad")]
    0 referencias
    public Almacenes ObtenerAlmacenesSociedad(SociedadesEntity sociedad)
    {
        return DALAlmacenes.ObtenerAlmacenesSociedad(sociedad.idSociedad);
    }

    [HttpPost]
    [Route("ObtenerAlmacenesEsp")]
    0 referencias
    public Almacenes ObtenerAlmacenesEsp(AlmacenesEntity almacen)
    {
        return DALAlmacenes.ObtenerAlmacenesEsp(almacen.sociedad_id, almacen.idAlmacen);
    }

    [HttpPost]
    [Route("ObtenerAlmacenID")]
    0 referencias
    public Almacenes ObtenerAlmacenID(AlmacenesEntity almacen)
    {
        return DALAlmacenes.ObtenerAlmacenID(almacen.idAlmacen);
    }

    [HttpPost]
    [Route("InsertarAlmacen")]
    0 referencias
    public Almacenes InsertarAlmacen(AlmacenesEntity almacen)
    {
        return DALAlmacenes.InsertarAlmacen(almacen);
    }

    [HttpPost]
    [Route("ModificarAlmacen")]
    0 referencias
    public Almacenes ModificarAlmacen(AlmacenesEntity almacen)
    {
        return DALAlmacenes.ModificarAlmacen(almacen);
    }

    [HttpPost]
    [Route("EliminarAlmacen")]
}
```

Figura D1. Controlador de almacenes.

```

23 export class AlmacenesService {
24
25   get obtenerAlmacen$(): Observable<AlmacenesEntity> {
26     return this.almacen$.asObservable();
27   }
28
29   asignarAlmacen(almacen: AlmacenesEntity) {
30     this.almacen$.next(almacen);
31   }
32
33   obtenerAlmacenes(): Observable<Almacenes> {
34     return this.http.get<Almacenes>(`${environment.apiUrl}almacenes/ObtenerAlmacenes`);
35   }
36
37   obtenerAlmacenId(almacen: AlmacenesEntity): Observable<Almacenes> {
38     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/ObtenerAlmacenId`,almacen);
39   }
40
41   obtenerAlmacenesSociedad(almacen: SociedadesEntity): Observable<Almacenes> {
42     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/ObtenerAlmacenesSociedad`, almacen );
43   }
44
45   obtenerAlmacenesEsp(almacen: AlmacenesEntity): Observable<Almacenes> {
46     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/ObtenerAlmacenesEsp`, almacen );
47   }
48
49   obtenerAlmacenID(almacen: AlmacenesEntity): Observable<Almacenes> {
50     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/ObtenerAlmacenID`, almacen );
51   }
52
53   agregarAlmacen(almacen: AlmacenesEntity): Observable<Almacenes> {
54     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/InsertarAlmacen`, almacen);
55   }
56
57   eliminarAlmacen(almacen: AlmacenesEntity): Observable<Almacenes> {
58     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/EliminarAlmacen`, almacen);
59   }
60
61   actualizarAlmacen(almacen: AlmacenesEntity): Observable<Almacenes> {
62     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/ModificarAlmacen`, almacen);
63   }
64
65   obtenerAlmacenesS(almacen: SociedadesEntity): Observable<Almacenes> {
66     return this.http.post<Almacenes>(`${environment.apiUrl}almacenes/ObtenerAlmacenesS`, almacen);
67   }
68
69   obtenerAlmacenN(nombre_almacen: AlmacenesEntity):Observable<Almacenes>{
70     return this http post/Almacenes/${environment.apiUrl}almacenes/ObtenerAlmacenN` nombre almacen`;

```

Figura D2. Servicios web de almacenes.

```

namespace WsAtiendo.Controllers
{
    [EnableCors(origins: "*", headers: "*", methods: "**")]
    [RoutePrefix("api/detalleimpuestos")]
    public class DetalleImpuestosController : ApiController
    {
        [HttpPost]
        [Route("ObtenerDetalleImpuesto")]
        public DetalleImpuestos ObtenerDetalleImpuesto(DetalleImpuestosEntity detalle)
        {
            return DALDetalleImpuestos.ObtenerDetalleImpuesto(detalle.detalle_movimiento_id);
        }

        [HttpPost]
        [Route("InsertarDetalleImpuestos")]
        public DetalleImpuestos InsertarDetalleImpuestos(DetalleImpuestosEntity detalle)
        {
            return DALDetalleImpuestos.InsertarDetalleImpuestos(detalle);
        }

        [HttpPost]
        [Route("ModificarMovimientoBP")]
        public DetalleImpuestos ModificarMovimientoBP(DetalleImpuestosEntity detalle)
        {
            return DALDetalleImpuestos.ModificarMovimientoBP(detalle);
        }

        [HttpPost]
        [Route("ModificarDetalleImpuestosBP")]
        public DetalleImpuestos ModificarDetalleImpuestosBP(DetalleImpuestosEntity detalle)
        {
            return DALDetalleImpuestos.ModificarDetalleImpuestosBP(detalle);
        }

        [HttpPost]
        [Route("ModificarDetalleImpuestosVal")]
        public DetalleImpuestos ModificarDetalleImpuestosVal(DetalleImpuestosEntity detalle)
        {
            return DALDetalleImpuestos.ModificarDetalleImpuestosVal(detalle);
        }
    }
}

```

Figura D3. Controlador de detalle impuestos.

```

You, hace 10 meses | 1 author (You)
@Injectable({
    providedIn: 'root'
})
export class DetalleImpuestosService {
    private detail$ = new BehaviorSubject<DetalleImpuestosEntity>(initDetalle);

    constructor(private readonly http: HttpClient) { }

    get obtenerDetalleImpuesto$: Observable<DetalleImpuestosEntity> {
        return this.detail$.asObservable();
    }

    obtenerDetalleImpuesto(detalle: DetalleImpuestosEntity): Observable<DetalleImpuestos> {
        return this.http.post<DetalleImpuestos>(`${environment.apiUrl}detalleimpuestos/ObtenerDetalleImpuesto`, detalle);
    }

    asignarDetalleImpuesto(detalle: DetalleImpuestosEntity) {
        this.detail$.next(detalle);
    }

    agregarDetalleImpuestos(detalle: DetalleImpuestosEntity): Observable<DetalleImpuestos> {
        return this.http.post<DetalleImpuestos>(`${environment.apiUrl}detalleimpuestos/InsertarDetalleImpuestos`, detalle);
    }

    modificarMovimientoBP(detalle: DetalleImpuestosEntity): Observable<DetalleImpuestos> {
        return this.http.post<DetalleImpuestos>(`${environment.apiUrl}detalleimpuestos/ModificarMovimientoBP`, detalle);
    }

    modificarDetalleImpuestosBP(detalle: DetalleImpuestosEntity): Observable<DetalleImpuestos> {
        return this.http.post<DetalleImpuestos>(`${environment.apiUrl}detalleimpuestos/ModificarDetalleImpuestosBP`, detalle);
    }

    modificarDetalleImpuestosVal(detalle: DetalleImpuestosEntity): Observable<DetalleImpuestos> {
        return this.http.post<DetalleImpuestos>(`${environment.apiUrl}detalleimpuestos/ModificarDetalleImpuestosVal`, detalle);
    }
}

```

Figura D4. Servicios web de detalle impuestos.

```

[HttpPost]
[Route("ObtenerDetalleMovimiento")]
public DetalleMovimientos ObtenerDetalleMovimiento(DetalleMovimientosEntity detalle)
{
    return DALDetalleMovimiento.ObtenerDetalleMovimiento(detalle.movimiento_id);
}

[HttpPost]
[Route("ObtenerDetalleMovimientoEx")]
public DetalleMovimientos ObtenerDetalleMovimientoEx(DetalleMovimientosEntity detalle)
{
    return DALDetalleMovimiento.ObtenerDetalleMovimientoEx(detalle.movimiento_id, detalle.producto_id);
}

[HttpPost]
[Route("ObtenerUltDetalleMovimiento")]
public DetalleMovimientos ObtenerUltDetalleMovimiento(DetalleMovimientosEntity detalle)
{
    return DALDetalleMovimiento.ObtenerUltDetalleMovimiento(detalle.movimiento_id);
}

[HttpPost]
[Route("ObtenerDetalleMovimientoSociedad")]
public DetalleMovimientos ObtenerDetalleMovimientoSociedad(SociedadesEntity sociedad)
{
    return DALDetalleMovimiento.ObtenerDetalleMovimientoSociedad(sociedad.idSociedad);
}

[HttpPost]
[Route("ObtenerDetalleMovimientoAlm")]
public DetalleMovimientos ObtenerDetalleMovimientoAlm(AlmacenesEntity almacen)
{
    return DALDetalleMovimiento.ObtenerDetalleMovimientoAlm(almacen.nombre_almacen);
}

[HttpPost]
[Route("Test")]
public String Test()
{
    return "1";
}

[HttpGet]
[Route("ObtenerDetalleMovimientoAlmF")]
public DetalleMovimientos ObtenerDetalleMovimientoAlmF(string almacen, string fechadesde, string fechahasta)
{
    return DALDetalleMovimiento.ObtenerDetalleMovimientoAlmF(almacen, fechadesde, fechahasta);
}

[HttpGet]
[Route("ObtenerDetalleMovimientoAlmFtipo")]

```

Figura D5. Controlador de detalle movimientos.

```

43 asignarDetalle(detallePedido: DetallesMovimientoEntity) {
44     this.details.next(detallePedido);
45 }
46
47 obtenerDetalleMovimiento(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
48     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimiento`, detalle);
49 }
50
51 obtenerDetalleMovimientoEx(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
52     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimientoEx`, detalle);
53 }
54
55 obtenerUltDetalleMovimiento(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
56     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerUltDetalleMovimiento`, detalle);
57 }
58
59 obtenerDetalleMovimientoSociedad(sociedad: SociedadesEntity): Observable<DetalleMovimiento> {
60     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimientoSociedad`, sociedad);
61 }
62
63 obtenerDetalleMovimientoAlm(almacen: AlmacenesEntity): Observable<DetalleMovimiento> {
64     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimientoAlm`, almacen);
65 }
66
67 obtenerDetalleMovimientoAlmF(almacen: string, fechadesde: string, fechahasta: string): Observable<DetalleMovimiento> {
68     return this.http.get<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimientoAlmF?almacen=${almacen}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
69 }
70
71 obtenerDetalleMovimientoAlmFtipo(almacen: string, fechadesde: string, fechahasta: string, tipo: string): Observable<DetalleMovimiento> {
72     return this.http.get<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimientoAlmFtipo?almacen=${almacen}&fechadesde=${fechadesde}&fechahasta=${fechahasta}&tipo=${tipo}`);
73 }
74
75 obtenerDetalleMovimientoAlmFtipo(almacen: string, tipo: string): Observable<DetalleMovimiento> {
76     return this.http.get<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ObtenerDetalleMovimientoAlmFtipo?almacen=${almacen}&tipo=${tipo}`);
77 }
78
79
80 agregarDetallePedido(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
81     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/InsertarDetallePedido`, detalle);
82 }
83
84 agregarDetalleCompra(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
85     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/InsertarDetalleCompra`, detalle);
86 }
87
88 agregarDetalleCompra(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
89     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/InsertarDetalleCompra`, detalle);
90 }
91
92 modificarDetallePedido(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
93     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ModificarDetallePedido`, detalle);
94 }
95
96 modificarDetallePedidoventa(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
97     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ModificarDetallePedidoventa`, detalle);
98 }
99
100 modificarDetalleCompra(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
101     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/ModificarDetalleCompra`, detalle);
102 }
103
104 eliminarDetallePedido(detalle: DetallesMovimientoEntity): Observable<DetalleMovimiento> {
105     return this.http.post<DetalleMovimiento>(`${environment.apiUrl}detallemovimiento/EliminarDetallePedido`, detalle);
106 }

```

Figura D6. Servicios web de detalle movimientos.

```

[HttpPost]
[Route("InsertarDetallePago")]
public DetallePagos InsertarDetallePago(DetallePagosEntity detalle)
{
    return DALDetallesPagos.InsertarDetallePago(detalle);
}

[HttpPost]
[Route("ObtenerDetallePagoMov")]
public DetallePagos ObtenerDetallePagoMov(MovimientosEntity movimiento)
{
    return DALDetallesPagos.ObtenerDetallePagoMov(movimiento.id);
}

[HttpPost]
[Route("ObtenerDetallePagoMovimiento")]
public DetallePagos ObtenerDetallePagoMovimiento(MovimientosEntity movimiento)
{
    return DALDetallesPagos.ObtenerDetallePagoMovimiento(movimiento.id);
}

[HttpPost]
[Route("ObtenerDetallePagoE")]
public DetallePagos ObtenerDetallePagoE(AlmacenesEntity almacen)
{
    return DALDetallesPagos.ObtenerDetallePagoE(almacen.sociedad_id);
}

[HttpPost]
[Route("ObtenerDetallePagoTD")]
public DetallePagos ObtenerDetallePagoTD(AlmacenesEntity almacen)
{
    return DALDetallesPagos.ObtenerDetallePagoTD(almacen.sociedad_id);
}

[HttpPost]
[Route("ObtenerDetallePagoTC")]
public DetallePagos ObtenerDetallePagoTC(AlmacenesEntity almacen)
{
    return DALDetallesPagos.ObtenerDetallePagoTC(almacen.sociedad_id);
}

[HttpGet]
[Route("ObtenerDetallePagoAla")]
public DetallePagos ObtenerDetallePagoAla(string almacen, string forma)
{
    return DALDetallesPagos.ObtenerDetallePagoAla(almacen, forma);
}

[HttpGet]
[Route("ObtenerDetallePagoAlaF")]
public DetallePagos ObtenerDetallePagoAlaF(string almacen, string forma, string fechadesde, string fechahasta)
{

```

✓ No se encontraron problemas.

Figura D7. Controlador de detalle pagos.

```

28 export class DetallesPagoService {
29   private detalle$ = new BehaviorSubject<DetallesPagoEntity>(initDetalle);
30
31   constructor(private readonly http: HttpClient) { }
32
33   get obtenerDetallePagos(): Observable<DetallesPagoEntity> {
34     return this.detalle$.asObservable();
35   }
36
37   asignarDetallePago(detalle: DetallesPagoEntity) {
38     this.detalle$.next(detalle);
39   }
40
41   agregarDetallePago(detalle: DetallesPagoEntity): Observable<DetallesPago> {
42     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/insertarDetallePago`, detalle);
43   }
44   obtenerDetallePagoMov(movimiento: MovimientosEntity): Observable<DetallesPago> {
45     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoMov`, movimiento);
46   }
47   obtenerDetallePagoMovimiento(movimiento: MovimientosEntity): Observable<DetallesPago> {
48     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoMovimiento`, movimiento);
49   }
50   obtenerDetallePagoI(almacen: AlmacenesEntity): Observable<DetallesPago> {
51     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoI`, almacen);
52   }
53   obtenerDetallePagoID(almacen: AlmacenesEntity): Observable<DetallesPago> {
54     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoID`, almacen);
55   }
56   obtenerDetallePagoC(almacen: AlmacenesEntity): Observable<DetallesPago> {
57     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoC`, almacen);
58   }
59
60   obtenerDetallePagoAla(almacen: AlmacenesEntity, forma: FormasPagoEntity): Observable<DetallesPago> {
61     return this.http.get<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoAla?almacen=${almacen.nombre_almacen}&forma=${forma.id}`);
62   }
63
64   obtenerDetallePagoAlaF(almacen: string, forma: string, fechadesde: string, fechahasta: string): Observable<DetallesPago> {
65     return this.http.get<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoAlaF?almacen=${almacen}&forma=${forma}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
66   }
67
68   obtenerDetallePagoF(sociedad: string, forma: string, fechadesde: string, fechahasta: string): Observable<DetallesPago> {
69     return this.http.get<DetallesPago>(`${environment.apiUrl}detallesPago/obtenerDetallePagoF?sociedad=${sociedad}&forma=${forma}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
70   }
71
72   modificarDetallePago(detalle: DetallesPagoEntity): Observable<DetallesPago> {
73     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/modificarDetallePago`, detalle);
74   }
75
76   eliminarDetallePago(detalle: DetallesPagoEntity): Observable<DetallesPago> {
77     return this.http.post<DetallesPago>(`${environment.apiUrl}detallesPago/eliminarDetallePago`, detalle);
78   }

```

Figura D8. Servicios web de detalle pagos.

```

10 namespace WsAtiendo.Controllers
11 {
12   [EnableCors(origins: "*", headers: "*", methods: "*")]
13   [RoutePrefix("api/formaspago")]
14   [References]
15   public class FormasPagoController : ApiController
16   {
17     [HttpGet]
18     [Route("ObtenerFormasPago")]
19     [References]
20     public FormasPago ObtenerFormasPago()
21     {
22       return DALFormasPago.ObtenerFormasPago();
23     }
24
25     [HttpPost]
26     [Route("ObtenerFormasPagoH")]
27     [References]
28     public FormasPago ObtenerFormasPagoH(FormasPagoEntity name)
29     {
30       return DALFormasPago.ObtenerFormasPagoH(name.nombre);
31     }
32   }

```

Figura D9. Controlador de formas de pago.

```

20 export class FormasPagoService {
21   private form$ = new BehaviorSubject<FormasPagoEntity>(initForm);
22
23   constructor(private readonly http: HttpClient) { }
24
25   get obtenerFormas$(): Observable<FormasPagoEntity> {
26     return this.form$.asObservable();
27   }
28
29   asignarForma(formaPago: FormasPagoEntity) {
30     this.form$.next(formaPago);
31   }
32
33   obtenerFormasPago(): Observable<FormasPago> {
34     return this.http.get<FormasPago>(`${environment.apiUrl}formasPago/ObtenerFormasPago`);
35   }
36
37   obtenerFormasPagoN(forma: FormasPagoEntity): Observable<FormasPago> {
38     return this.http.post<FormasPago>(`${environment.apiUrl}formasPago/ObtenerFormasPagoN`, forma);
39   }
40 }
41

```

Figura D10. Servicios web de formas de pago.

```

[HttpPost]
[Route("ObtenerInventarios")]
public IActionResult ObtenerInventarios()
{
    return Ok(Inventario.ObtenerInventarios());
}

[HttpPost]
[Route("ObtenerPortafolios")]
public IActionResult ObtenerPortafolios(almacenEntity almacen)
{
    return Ok(Inventario.ObtenerPortafolios(almacen.idalmacen));
}

[HttpPost]
[Route("ObtenerInventariosExisten")]
public IActionResult ObtenerInventariosExisten(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerInventariosExisten(inventario.producto_id, inventario.almacen_id));
}

[HttpPost]
[Route("ObtenerInventariosAlm")]
public IActionResult ObtenerInventariosAlm(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerInventariosAlm(inventario.producto_id, inventario.almacen_id));
}

[HttpPost]
[Route("ObtenerPortafoliosCategorias")]
public IActionResult ObtenerPortafoliosCategorias(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerPortafoliosCategorias(inventario.categoria_id, inventario.almacen_id));
}

[HttpPost]
[Route("ObtenerPortafoliosLineas")]
public IActionResult ObtenerPortafoliosLineas(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerPortafoliosLineas(inventario.almacen_id, inventario.linea, inventario.categoria_id));
}

[HttpPost]
[Route("ObtenerPortafoliosModelos")]
public IActionResult ObtenerPortafoliosModelos(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerPortafoliosModelos(inventario.almacen_id, inventario.modelo, inventario.linea, inventario.categoria_id));
}

[HttpPost]
[Route("ObtenerPortafoliosModelosColores")]
public IActionResult ObtenerPortafoliosModelosColores(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerPortafoliosModelosColores(inventario.almacen_id, inventario.modelo, inventario.color, inventario.categoria_id, inventario.linea));
}

[HttpPost]
[Route("ObtenerPortafoliosLineasCodigo")]
public IActionResult ObtenerPortafoliosLineasCodigo(InventarioEntity inventario)
{
    return Ok(Inventario.ObtenerPortafoliosLineasCodigo(inventario.almacen_id, inventario.linea, inventario.categoria_id));
}

[HttpPost]
[Route("ObtenerPortafoliosCategoriasCodigo")]
public IActionResult ObtenerPortafoliosCategoriasCodigo(InventarioEntity inventario)

```

Figura D11. Controlador de inventarios.

```

export class InventarioService {
  obtenerModelosLineas(lineas: LineasEntity): Observable<Inventarios> {
    return this.http.get<Inventarios>(`${environment.apiUrl}categorias/ObtenerCategorias`);
  }

  obtenerCategoria(): Observable<Inventarios> {
    return this.http.get<Inventarios>(`${environment.apiUrl}categorias/ObtenerCategorias`);
  }

  obtenerCategoriaid(categoria: CategoriasEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}categorias/ObtenerCategoriaid`, categoria);
  }

  // Obtencion de datos para las tablas

  obtenerInventario(): Observable<Inventarios> {
    return this.http.get<Inventarios>(`${environment.apiUrl}inventarios/ObtenerInventarios`);
  }

  obtenerPortafolios(almacen: AlmacenesEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafolios`, almacen);
  }

  obtenerPortafoliosCategoria(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafoliosCategoria`, inventario);
  }

  obtenerPortafoliosCategoriaSugerido(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafoliosCategoriaSugerido`, inventario);
  }

  obtenerPortafoliosLineas(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafoliosLineas`, inventario);
  }

  obtenerPortafoliosLineasSugerido(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafoliosLineasSugerido`, inventario);
  }

  obtenerPortafoliosModelos(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafoliosModelos`, inventario);
  }

  obtenerPortafoliosColores(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerPortafoliosModelosColores`, inventario);
  }

  obtenerInventariosExiste(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerInventarioExiste`, inventario);
  }

  obtenerInventariosAlm(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ObtenerInventarioAlm`, inventario);
  }

  agregarInventario(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/InsertarInventarios`, inventario);
  }

  agregarInventarioUltimo(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/InsertarInventarioUltimo`, inventario);
  }

  actualizarInventario(inventario: InventariosEntity): Observable<Inventarios> {
    return this.http.post<Inventarios>(`${environment.apiUrl}inventarios/ModificarInventarios`, inventario);
  }
}

```

Figura D12. Servicios web de inventarios.

```

[HttpPost]
[Route("ObtenerMovimientosAlmacen")]
a referencia
public Movimientos ObtenerMovimientosAlmacen(MovimientosEntity movimiento)
{
    return DALMovimientos.ObtenerMovimientosAlmacen(movimiento.almacen_id);
}

[HttpPost]
[Route("ObtenerMovimientosAlmacenV")]
a referencia
public Movimientos ObtenerMovimientosAlmacenV(MovimientosEntity movimiento)
{
    return DALMovimientos.ObtenerMovimientosAlmacenV(movimiento.almacen_id);
}

[HttpGet]
[Route("ObtenerMovimientosAlmacenFecha")]
a referencia
public Movimientos ObtenerMovimientosAlmacenFecha(string almacen, string fechadesde, string fechahasta)
{
    return DALMovimientos.ObtenerMovimientosAlmacenFecha(almacen, fechadesde, fechahasta);
}

[HttpGet]
[Route("ObtenerMovimientosAlmacenNombre")]
a referencia
public Movimientos ObtenerMovimientosAlmacenNombre(string almacen)
{
    return DALMovimientos.ObtenerMovimientosAlmacenNombre(almacen);
}

[HttpGet]
[Route("ObtenerMovimientosFechas")]
a referencia
public Movimientos ObtenerMovimientosFechas(string sociedad, string fechadesde, string fechahasta)
{
    return DALMovimientos.ObtenerMovimientosFechas(sociedad, fechadesde, fechahasta);
}

[HttpGet]
[Route("ObtenerMovimientosTodos")]
a referencia
public Movimientos ObtenerMovimientosTodos(string sociedad)
{
    return DALMovimientos.ObtenerMovimientosTodos(sociedad);
}

[HttpGet]
[Route("ObtenerMovimientosAlmacenFechaPAGO")]
a referencia
public Movimientos ObtenerMovimientosAlmacenFechaPAGO(string almacen, string fechadesde, string fechahasta, string pago)
{
    return DALMovimientos.ObtenerMovimientosAlmacenFechaPAGO(almacen, fechadesde, fechahasta, pago);
}

[HttpGet]
[Route("ObtenerMovimientosAlmacenNombrePAGO")]
a referencia
public Movimientos ObtenerMovimientosAlmacenNombrePAGO(string almacen, string pago)
{
    return DALMovimientos.ObtenerMovimientosAlmacenNombrePAGO(almacen, pago);
}

[HttpGet]
[Route("ObtenerMovimientosFechasPAGO")]
a referencia

```

Figura D13. Controlador de movimientos.

```

53 asignarMovimiento(movimiento: MovimientosEntity) {}
54 this.movements$.next(movimiento);
55
56
57 agregarMovimiento(movimiento: MovimientosEntity): Observable<Movimientos> {
58   return this.http.post<Movimientos>(`${environment.apiUrl}movimientos/insertarMovimiento`, movimiento);
59 }
60
61 obtenerUltimoSecuencial(almacen_id: string): Observable<Movimientos> {
62   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerUltimoSecuencial?almacen_id=${almacen_id}`);
63 }
64
65 obtenerMovimientoUno(movimiento: MovimientosEntity): Observable<Movimientos> {
66   return this.http.post<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientoUno`, movimiento);
67 }
68
69 obtenerMovimientoID(movimiento: MovimientosEntity): Observable<Movimientos> {
70   return this.http.post<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientoID`, movimiento);
71 }
72
73 obtenerMovimientosAlmacen(movimiento: MovimientosEntity): Observable<Movimientos> {
74   return this.http.post<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosAlmacen`, movimiento);
75 }
76
77 obtenerMovimientosAlmacenFecha(almacen: string, fechadesde: string, fechahasta: string): Observable<Movimientos> {
78   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosAlmacenFecha?almacen=${almacen}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
79 }
80
81 obtenerMovimientosCompF(almacen: string, fechadesde: string, fechahasta: string): Observable<Movimientos> {
82   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosCompF?almacen=${almacen}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
83 }
84
85 obtenerMovimientosVentF(almacen: string, fechadesde: string, fechahasta: string): Observable<Movimientos> {
86   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosVentF?almacen=${almacen}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
87 }
88
89 obtenerMovimientosAlmacenNombre(almacen: string): Observable<Movimientos> {
90   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosAlmacenNombre?almacen=${almacen}`);
91 }
92
93 obtenerMovimientosFechas(sociedad: string, fechadesde: string, fechahasta: string): Observable<Movimientos> {
94   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosFechas?sociedad=${sociedad}&fechadesde=${fechadesde}&fechahasta=${fechahasta}`);
95 }
96
97 obtenerMovimientoCLAVEACCESO(movimiento: MovimientosEntity): Observable<Movimientos> {
98   return this.http.post<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientoCLAVEACCESO`, movimiento);
99 }
100
101 obtenerMovimientosTodos(sociedad: string): Observable<Movimientos> {
102   return this.http.get<Movimientos>(`${environment.apiUrl}movimientos/ObtenerMovimientosTodos?sociedad=${sociedad}`);
103 }
104

```

Figura D14. Servicios web de movimientos.

```

[HttpGet]
[Route("ObtenerSociedades")]
// referencia
public Sociedades ObtenerGrupos()
{
    return DAL.Sociedades.ObtenerSociedades();
}

[HttpPost]
[Route("ObtenerSociedadEmail")]
// referencia
public string ObtenerSociedadEmail(SociedadesEntity sociedad)
{
    return DAL.Sociedades.ObtenerEmailSociedadPorId(sociedad);
}

[HttpPost]
[Route("EliminarSociedad")]
// referencia
public Sociedades EliminarSociedad(SociedadesEntity sociedad)
{
    return DAL.Sociedades.EliminarSociedad(sociedad);
}

[HttpPost]
[Route("InsertarSociedad")]
// referencia
public Sociedades InsertarSociedad(SociedadesEntity sociedad)
{
    if (sociedad.IdGrupo.Equals("1"))
    {
        sociedad = DAL.Clientes.ObtenerSociedadesSAP(sociedad);
    }
    return DAL.Sociedades.InsertarSociedad(sociedad);
}

[HttpPost]
[Route("ModificarSociedad")]
// referencia
public Sociedades ModificarSociedad(SociedadesEntity sociedad)
{
    if (sociedad.IdGrupo.Equals("1"))
    {
        sociedad = DAL.Clientes.ObtenerSociedadesSAP(sociedad);
    }
    return DAL.Sociedades.ModificarSociedad(sociedad);
}

[HttpPost]
[Route("ModificarSociedadImagen")]
// referencia
public Sociedades ModificarSociedadImagen(SociedadesEntity sociedad)
{
    return DAL.Sociedades.ModificarSociedadLogo(sociedad);
}

[HttpPost]
[Route("ModificarPersonal")]
// referencia
public Sociedades ModificarPersonal(SociedadesEntity sociedad)
{
    return DAL.Sociedades.ModificarPersonal(sociedad);
}

```

Figura D15. Controlador de sociedades.

```

42 | asignarSociedad(sociedad: SociedadesEntity) {
43 |   this.sociedad$.next(sociedad);
44 | }
45 | agregarSociedad(sociedad: SociedadesEntity): Observable<Sociedades> {
46 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/InsertarSociedad`, sociedad);
47 | }
48 |
49 | obtenerAlmacenPertenece(sociedadId: string): Observable<string> {
50 |   return this.http.get<string>(`${environment.apiUrl}sociedades/ObtenerAlmacenPersonalIdPorIdPersonal?IdSociedad=${sociedadId}`);
51 | }
52 |
53 | obtenerSociedades(): Observable<Sociedades> {
54 |   return this.http.get<Sociedades>(`${environment.apiUrl}sociedades/ObtenerSociedades`);
55 | }
56 | eliminarSociedad(sociedad: SociedadesEntity): Observable<Sociedades> {
57 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/EliminarSociedad`, sociedad);
58 | }
59 | actualizarSociedad(sociedad: SociedadesEntity): Observable<Sociedades> {
60 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ModificarSociedad`, sociedad);
61 | }
62 |
63 | actualizarSociedadImagen(sociedad: SociedadesEntity): Observable<Sociedades> {
64 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ModificarSociedadImagen`, sociedad);
65 | }
66 |
67 |
68 |
69 | actualSociedad(sociedad: SociedadesEntity): Observable<Sociedades> {
70 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ActualizarSociedad`, sociedad);
71 | }
72 | actualizarPass(sociedad: SociedadesEntity): Observable<Sociedades> {
73 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ActualizarPass`, sociedad);
74 | }
75 | actualizarCertificado(sociedad: SociedadesEntity): Observable<Sociedades> {
76 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ActualizarCertificado`, sociedad);
77 | }
78 | actualizarClaveCorreo(sociedad: SociedadesEntity): Observable<Sociedades> {
79 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ActualizarClaveCorreo`, sociedad);
80 | }
81 | obtenerSociedadI(sociedad: SociedadesEntity): Observable<Sociedades> {
82 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ObtenerSociedad`, sociedad);
83 | }
84 | obtenerUsuario(sociedad: SociedadesEntity): Observable<Sociedades> {
85 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ObtenerUsuario`, sociedad);
86 | }
87 | obtenerUser(sociedad: SociedadesEntity): Observable<Sociedades> {
88 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ObtenerUser`, sociedad);
89 | }
90 | obtenerSociedadDatos(sociedad: SociedadesEntity): Observable<Sociedades> {
91 |   return this.http.post<Sociedades>(`${environment.apiUrl}sociedades/ObtenerSociedadDatos`, sociedad);
92 | }
93 | obtenerSociedadesN(nombre: SociedadesEntity): Observable<Sociedades>{

```

Figura D16. Servicios web de sociedades.

Anexo E. Funcionalidades para servicios web del SRI.

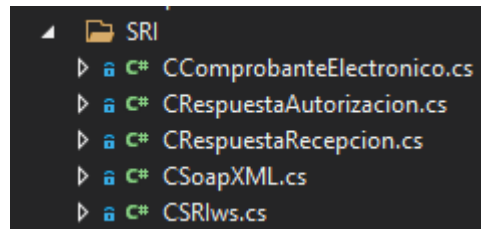


Figura E1. Estructura de clases SRI.

```
1 referencia
public CRespuestaRecepcion RecepcionComprobantesOnlinePrueba(string xml)
{
    String respuesta = "";
    CRespuestaRecepcion RespuestaRecepcionPrueba = new CRespuestaRecepcion();
    var soapEnvelopeXml = new XmlDocument();

    HttpWebRequest webRequest = (HttpWebRequest)HttpWebRequest.Create("https://celcer.sri.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantesOffline?wsdl");
    webRequest.ContentType = "text/xml; charset=utf-8";
    webRequest.Accept = "text/xml";
    webRequest.Method = "POST";

    soapEnvelopeXml.LoadXml(new CSoapXML().RecepcionComprobanteSoap(xml));
    using (Stream requestStream = webRequest.GetRequestStream())
    {
        soapEnvelopeXml.Save(requestStream);
    }

    using (HttpWebResponse response = (HttpWebResponse)webRequest.GetResponse())
    {
        using (StreamReader rd = new StreamReader(response.GetResponseStream()))
        {
            respuesta = rd.ReadToEnd();
            var soapResult = XmlDocument.Parse(respuesta);
            var responseXml = soapResult.Descendants("RespuestaRecepcionComprobante").ToList();
            foreach (var xmlDoc in responseXml)
            {
                RespuestaRecepcionPrueba = (CRespuestaRecepcion)DeserializeFromXmlElement(xmlDoc, typeof(CRespuestaRecepcion));
            }
        }
    }
    return RespuestaRecepcionPrueba;
}
```

Figura E2. Método de recepción de comprobante.

```
1 referencia
public CRespuestaAutorizacion AutorizacionComprobanteOnlinePrueba(string claveAcceso)
{
    CRespuestaAutorizacion RespuestaAutorizacion = new CRespuestaAutorizacion();
    try
    {
        String respuesta = "";
        var soapEnvelopeXml = new XmlDocument();

        HttpWebRequest webRequest = (HttpWebRequest)HttpWebRequest.Create("https://celcer.sri.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantesOffline?wsdl");
        webRequest.ContentType = "text/xml; charset=utf-8";
        webRequest.Accept = "text/xml";
        webRequest.Method = "POST";

        soapEnvelopeXml.LoadXml(new CSoapXML().AutorizacionComprobanteSoap(claveAcceso));
        using (Stream requestStream = webRequest.GetRequestStream())
        {
            soapEnvelopeXml.Save(requestStream);
        }

        using (HttpWebResponse response = (HttpWebResponse)webRequest.GetResponse())
        {
            using (StreamReader rd = new StreamReader(response.GetResponseStream()))
            {
                respuesta = rd.ReadToEnd();
                var soapResult = XmlDocument.Parse(respuesta);
                var responseXml = soapResult.Descendants("RespuestaAutorizacionComprobante").ToList();
                foreach (var xmlDoc in responseXml)
                {
                    RespuestaAutorizacion = (CRespuestaAutorizacion)DeserializeFromXmlElement(xmlDoc, typeof(CRespuestaAutorizacion));
                }
            }
        }
        return RespuestaAutorizacion;
    }
    catch (Exception)
    {
        return null;
    }
}
```

Figura E3. Método de autorización de comprobante.

```

1 referencia
public static String RecibirXMLSrI(string movimiento_id)
{
    try
    {
        using (MySQLConnection mConnection = new MySQLConnection(ConnectionString))
        {
            mConnection.Open();
            string queryS = @"SELECT *
                                FROM movimientos
                                WHERE id=?pMovimiento_id";
            using (MySQLCommand myCmd = new MySQLCommand(queryS, mConnection))
            {
                myCmd.Parameters.AddWithValue("?pMovimiento_id", movimiento_id);

                using (MySQLDataAdapter adapter = new MySQLDataAdapter(myCmd))
                {
                    DataSet dataSet = new DataSet();
                    adapter.Fill(dataSet);

                    DataTable dataTable = dataSet.Tables[0];
                    foreach (DataRow row in dataTable.Rows)
                    {
                        String url_factura = row["url_factura"].ToString();

                        CComprobanteElectronico cComprobante = new CComprobanteElectronico();

                        var recepcion = cComprobante.RecepcionComprobantePrueba(url_factura);
                        var est = recepcion.Estado;

                        return est;
                    }
                }
            }
        }
    }
    catch (Exception ex)
    {
    }
}

```

Figura E4. DAL de recepción de comprobante.

```

1 referencia
public static String AutorizarXMLSrI(string movimiento_id)
{
    try
    {
        using (MySQLConnection mConnection = new MySQLConnection(ConnectionString))
        {
            mConnection.Open();
            string queryS = @"SELECT *
                                FROM movimientos
                                WHERE id=?pMovimiento_id";
            using (MySQLCommand myCmd = new MySQLCommand(queryS, mConnection))
            {
                myCmd.Parameters.AddWithValue("?pMovimiento_id", movimiento_id);

                using (MySQLDataAdapter adapter = new MySQLDataAdapter(myCmd))
                {
                    DataSet dataSet = new DataSet();
                    adapter.Fill(dataSet);

                    DataTable dataTable = dataSet.Tables[0];
                    foreach (DataRow row in dataTable.Rows)
                    {
                        String clave_acceso = row["clave_acceso"].ToString();

                        CComprobanteElectronico cComprobante = new CComprobanteElectronico();

                        var respuestaComprobanteProduccion = cComprobante.AutorizacionComprobantePrueba(clave_acceso);
                        var est = respuestaComprobanteProduccion.Comprobantes[0].Estado;
                        var hora = respuestaComprobanteProduccion.Comprobantes[0].FechaAutorizacion;

                        return est + hora;
                    }
                }
            }
        }
    }
}

```

Figura E5. DAL de autorización de comprobante.

```

namespace WsAtiendo.Controllers
{
    [EnableCors(origins: "*", headers: "*", methods: "*")]
    [RoutePrefix("api/sriws")]
    public class SriController : ApiController
    {
        [HttpGet]
        [Route("RecibirXMLSri")]
        public string RecibirXMLSri(string movimiento_id)
        {
            return DALsri.RecibirXMLSri(movimiento_id);
        }

        [HttpGet]
        [Route("AutorizarXMLSri")]
        public string AutorizarXMLSri(string movimiento_id)
        {
            return DALsri.AutorizarXMLSri(movimiento_id);
        }

        [HttpGet]
        [Route("EnviarComprobanteCorreo")]
        public string EnviarComprobanteCorreo(string movimiento_id)
        {
            return DALEnviarCom.EnviarComprobanteCorreo(movimiento_id);
        }
    }
}

```

Figura E6. Controlador de autorización, recepción y envío de comprobantes.

```

6  @Injectable({
7      providedIn: 'root'
8  })
9  export class SriwsService {
10
11      constructor(private readonly http: HttpClient) { }
12
13      recibirXMLSri(movimiento_id: string): Observable<String> {
14          return this.http.get<String>(`${environment.apiUrl}sriws/RecibirXMLSri?movimiento_id=${movimiento_id}`);
15      }
16
17      autorizarXMLSri(movimiento_id: string): Observable<String> {
18          return this.http.get<String>(`${environment.apiUrl}sriws/AutorizarXMLSri?movimiento_id=${movimiento_id}`);
19      }
20
21      enviarComprobanteCorreo(movimiento_id: string): Observable<String> {
22          return this.http.get<String>(`${environment.apiUrl}sriws/EnviarComprobanteCorreo?movimiento_id=${movimiento_id}`);
23      }
24

```

Figura E7. Consumo de servicios web en Angular.

Anexo F. Manual de usuario.

En las siguientes páginas se encuentra el manual de usuario para el cliente o usuario del sistema “aTiendo”, con un enfoque en el módulo de facturación electrónica.

MANUAL DE USUARIO PARA EL CLIENTE DEL SISTEMA “ATIENDO” CON
EL ENFOQUE EN LA FACTURACIÓN ELECTRÓNICA.

Junio 2024.

1. Funcionalidades generales

1.1. Inicio de sesión

Como usuario, se debe ingresar al sistema con las credenciales brindadas o se puede solicitar al administrador nuevas credenciales. Se verificará el rol de usuario con el que se ingrese.

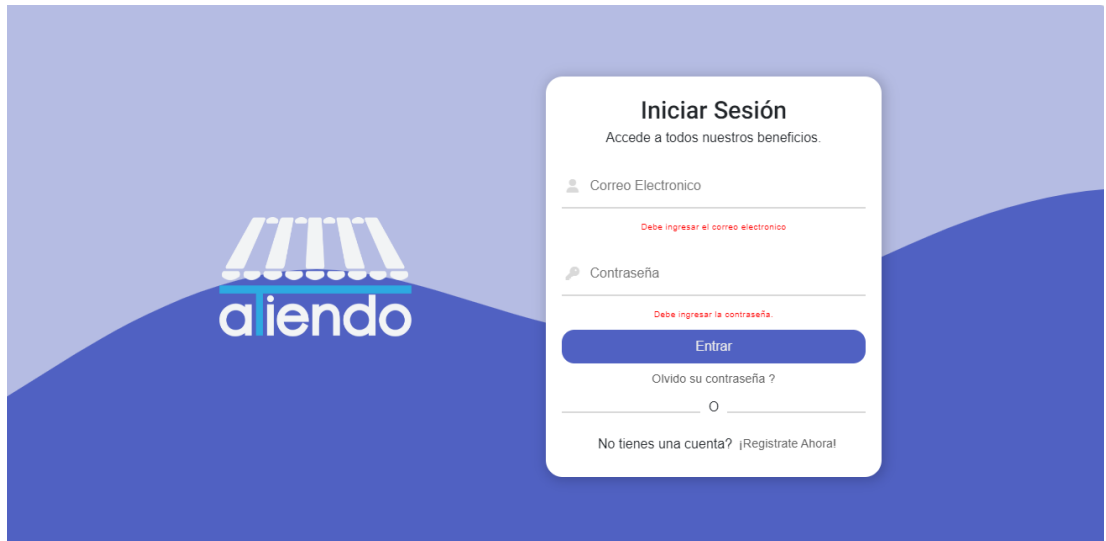


Figura 1. Pantalla de inicio de sesión.

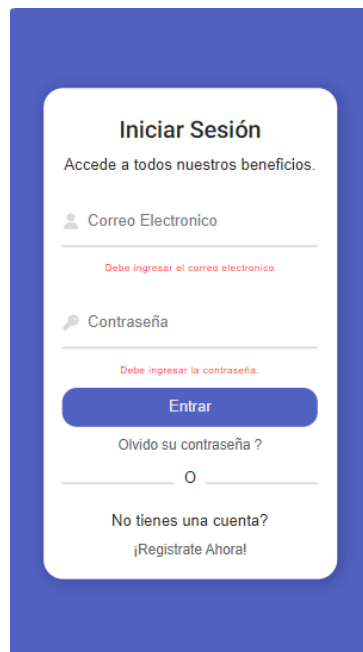


Figura 2. Pantalla de inicio de sesión en dispositivo móvil.

En la figura 1 y 2 se pueden observar los siguientes componentes:

- Campo de texto para el correo electrónico.
- Campo de texto para la contraseña.
- Botón para ingresar al sistema.

Para ingresar al sistema se debe colocar tanto correo electrónico, así como la contraseña, en el caso de que alguno de estos sea incorrecto, el sistema mostrará un mensaje de error.

1.2. Dashboard y barra lateral.

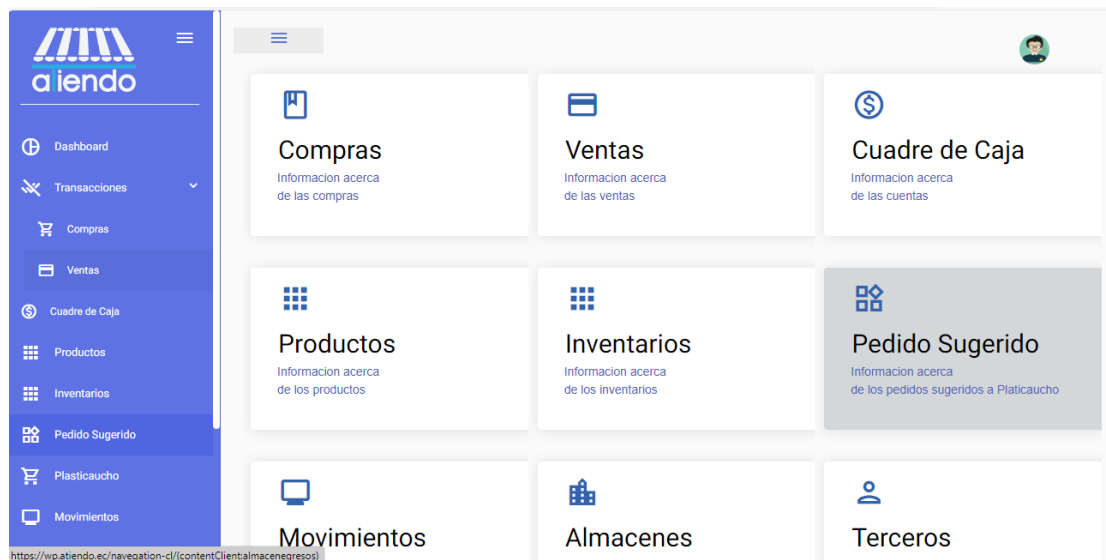


Figura 3. Pantalla de dashboard y barra lateral.

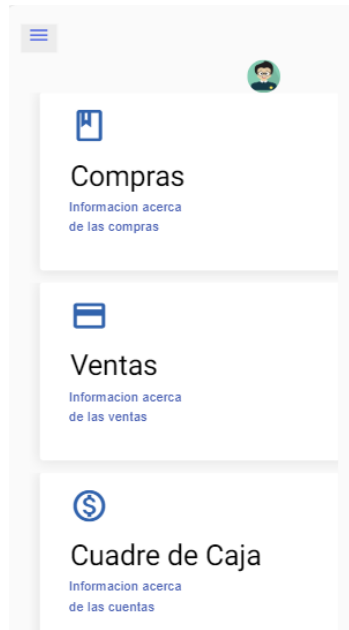


Figura 4. Pantalla de dashboard en dispositivo móvil.

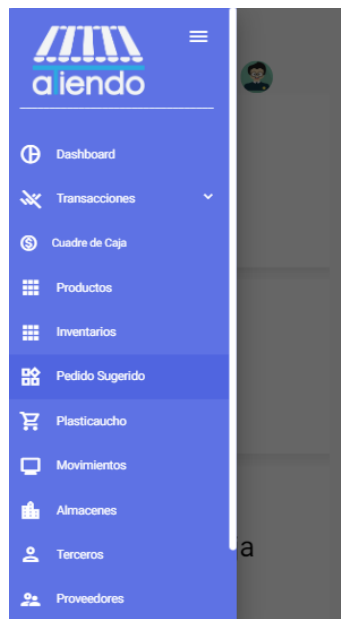


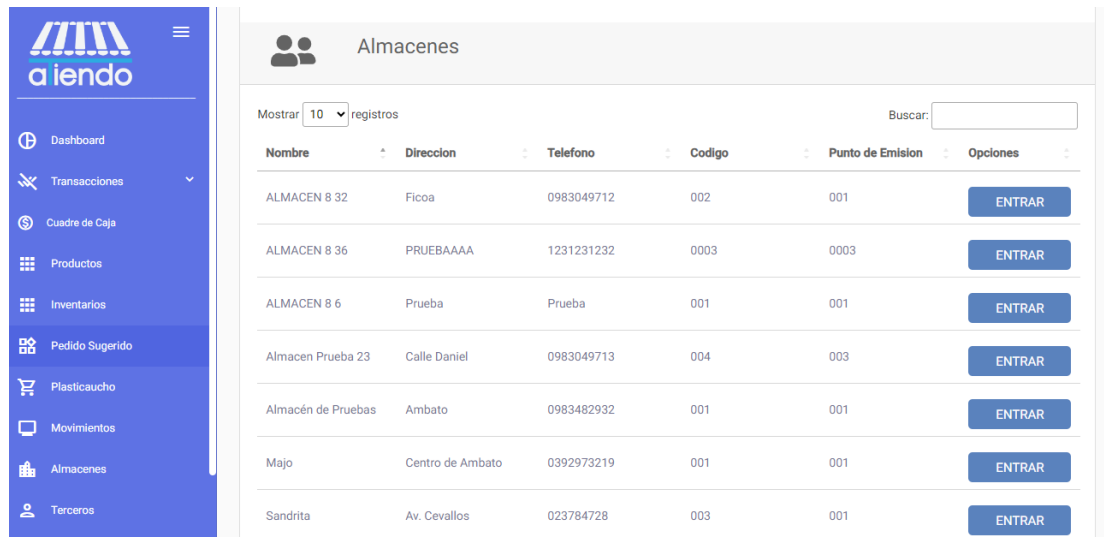
Figura 5. Barra lateral en dispositivo móvil.

En el dashboard y barra lateral se encuentra el módulo de ventas, en el cual se puede realizar el proceso de facturación electrónica.

A partir de ahora, y luego de haber validado la vista responsiva, se colocarán imágenes desde una vista web de computador; teniendo en cuenta que todo el funcionamiento es aplicable para dispositivos móviles.

2. Funcionalidades específicas.

2.1. Lista de almacenes.



Nombre	Direccion	Telefono	Codigo	Punto de Emision	Opciones
ALMACEN 8 32	Ficoa	0983049712	002	001	ENTRAR
ALMACEN 8 36	PRUEBAAAA	1231231232	0003	0003	ENTRAR
ALMACEN 8 6	Prueba	Prueba	001	001	ENTRAR
Almacen Prueba 23	Calle Daniel	0983049713	004	003	ENTRAR
Almacén de Pruebas	Ambato	0983482932	001	001	ENTRAR
Majo	Centro de Ambato	0392973219	001	001	ENTRAR
Sandrita	Av. Cevallos	023784728	003	001	ENTRAR

Figura 6. Pantalla de lista de almacenes.

Al ingresar a la venta se observa la lista de almacenes que posee el usuario. Se deberá ingresar al almacén del cual se realizará la venta respectiva.

2.2. Generar nueva venta.

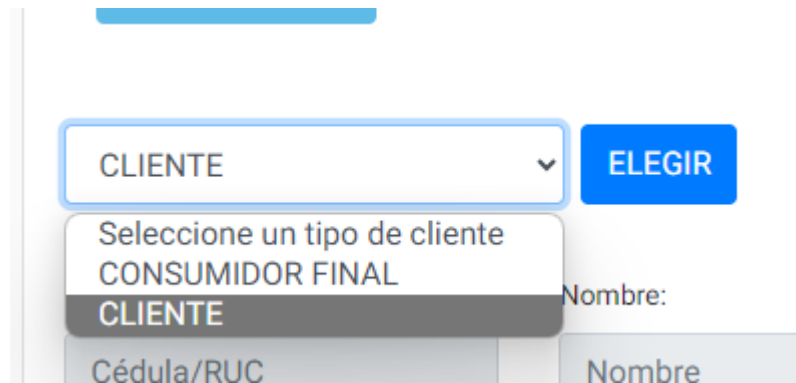


CÓDIGO	TERCERO	FECHA	TOTAL FACTURA
2321	Daniel Eduardo Montero López	25/06/2024	431.25
2240	Daniel Eduardo Montero López	19/06/2024	359.50

Figura 7. Pantalla de generación de nueva venta.

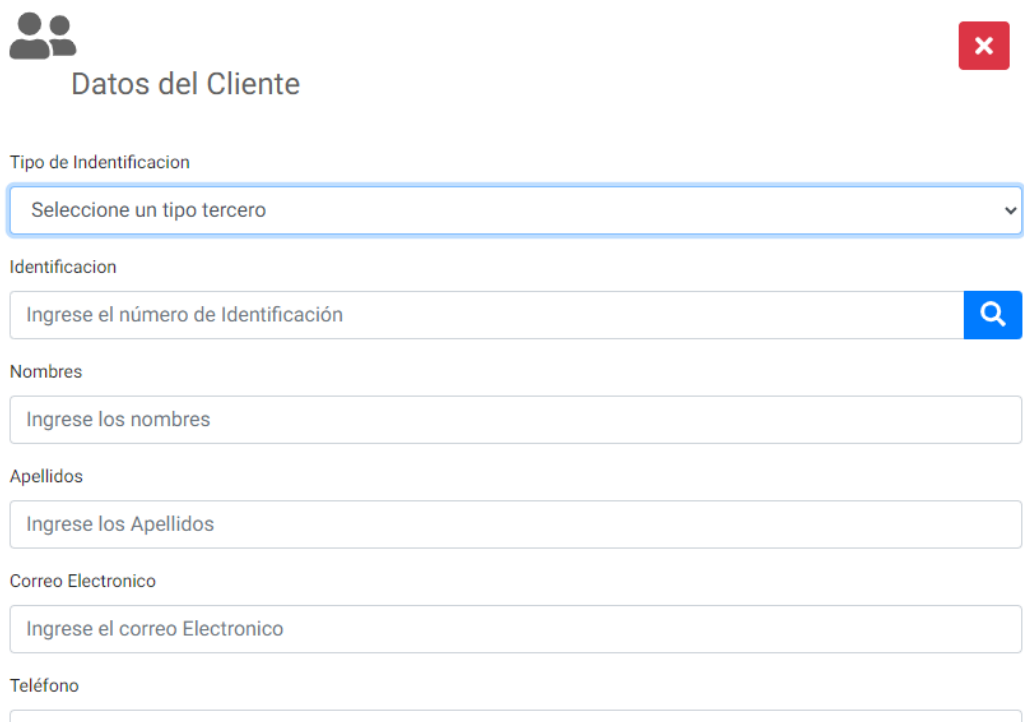
En la figura 7 se visualiza la pantalla de lista de ventas en el almacén escogido y el modo de generación de una nueva venta, mediante el botón “NUEVO” en la esquina superior derecha.

2.3. Datos del cliente de facturación.



A screenshot of a web application showing a dropdown menu for selecting a client type. The dropdown is open, displaying two options: "CONSUMIDOR FINAL" and "CLIENTE", with "CLIENTE" currently selected and highlighted. To the right of the dropdown is a blue button labeled "ELEGIR". Below the dropdown, there are input fields for "Cédula/RUC" and "Nombre".

Figura 8. Selección de cliente.



A screenshot of a web application titled "Datos del Cliente" (Client Data). The form contains several input fields for client information: "Tipo de Identificación" (Type of Identification) with a dropdown menu showing "Seleccione un tipo tercero"; "Identificación" (Identification) with a text input field and a search icon; "Nombres" (Names) with a text input field; "Apellidos" (Surnames) with a text input field; "Correo Electronico" (Email) with a text input field; and "Teléfono" (Phone) with a text input field. A red close button is visible in the top right corner.

Figura 9. Datos del cliente.

Una vez se haya escogido si el cliente desea que la factura sea emitida con sus datos, se debe llenar dichos datos como se observa en la figura 9. En el caso de que el cliente ya haya comprado en nuestra sociedad, se puede cargar sus datos automáticamente con el botón de búsqueda.

En el caso de que el cliente no quiera que la factura sea emitida con sus datos, no se pedirá ningún dato y se colocará como “consumidor final”.

2.4. Añadir productos.



Figura 10. Botón añadir.

Mostrar registros

Buscar:

IMAGEN	PRODUCTO	CATEGORIA	TIPO	MARCA	MODELO	COLOR	TAMAÑO	CARACTERISTICA	GENERO
	ANDINA AZUL GRIS SIN FORRO FEMENINO 34	CALZADO	BOTAS	VENUS	ANDINA	AZUL GRIS	34	SIN FORRO	FEMENINO

CANTIDAD:

PVP: 9.00

IVA: 15%

ICE: No aplica ICE

STOCK: 0

:

:

Figura 11. Añadir productos a la venta.

Para añadir productos a la venta respectiva se debe hacer clic en el botón “Añadir” y colocar la cantidad de cada producto que se vaya añadiendo. Además, se cuenta con un botón de búsqueda para filtrar fácilmente el producto correspondiente.

CONTINUAR

Una vez se haya completado la venta y validado los subtotales, se debe proceder con el pago de esta. Para continuar se debe presionar el botón “CONTINUAR”.

Seleccionar método de pagc

Moneto

ABONAR

RESTANTE: 16.10

Seleccionar método de pago

EFFECTIVO

TARJETA DE DEBITO

TARJETA DE CREDITO

VALOR

FINALIZAR VENTA

Para proceder con la forma de pago, se debe abonar el monto en dependencia de la forma de pago que se maneje en la sociedad. El botón “FINALIZAR VENTA” se activará únicamente cuando se haya abonado todo el restante.

FORMA DE PAGO	FORMA DE PAGO	VALOR	
350	EFFECTIVO	16.10	 

En el caso de que se haya abonado un monto incorrectamente, se puede eliminar o editar este monto abonado.

[illegible]

Finalmente, una vez se haya completado la venta, se genera la factura electrónica, la misma que se puede descargar en .pdf y enviar al consumidor final por correo electrónico.