



UNIVERSIDAD TÉCNICA DE AMBATO

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

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

Tema:

APLICACIÓN WEB PROGRESIVA (PWA) UTILIZANDO UNA BASE DE
DATOS NOSQL PARA EL CONTROL DE VENDEDORES Y GESTIÓN DE
PEDIDOS DE LA EMPRESA AMBAGRO S.A.

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo a la
obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de software

AUTOR: Diego Fernando Santamaría Coque

TUTOR: Ing. Leonardo David Torres Valverde, Mg.

Ambato - Ecuador

septiembre – 2022

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: APLICACIÓN WEB PROGRESIVA (PWA) UTILIZANDO UNA BASE DE DATOS NOSQL PARA EL CONTROL DE VENDEDORES Y GESTIÓN DE PEDIDOS DE LA EMPRESA AMBAGRO S.A, desarrollado bajo la modalidad Proyecto de Investigación, por el señor Diego Fernando Santamaría Coque, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, 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 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022.

Ing. Leonardo Torres, Mg.

TUTOR

AUTORÍA

El presente proyecto de investigación titulado: APLICACIÓN WEB PROGRESIVA (PWA) UTILIZANDO UNA BASE DE DATOS NOSQL PARA EL CONTROL DE VENDEDORES Y GESTIÓN DE PEDIDOS DE LA EMPRESA AMBAGRO S.A. es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022.



Diego Fernando Santamaría Coque

C.C. 1805300447

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Diego Fernando Santamaría Coque, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN WEB PROGRESIVA (PWA) UTILIZANDO UNA BASE DE DATOS NOSQL PARA EL CONTROL DE VENDEDORES Y GESTIÓN DE PEDIDOS DE LA EMPRESA AMBAGRO S.A**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, septiembre 2022.

Ing. Pilar Urrutia, Mg.

PRESIDENTA DEL TRIBUNAL

Ing. Félix Fernández, PhD.

PROFESOR CALIFICADOR

Ing. Edison Álvarez, Mg.

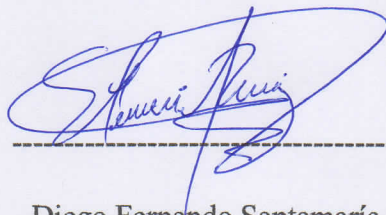
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, septiembre 2022.



Diego Fernando Santamaría Coque

C.C. 1805300447

AUTOR

DEDICATORIA

El presente trabajo de investigación lo dedico principalmente a Dios, por darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.

A mis padres, Fernando y Eugenia quienes con su cariño, esfuerzo, paciencia y apoyo incondicional me han permitido llegar a cumplir un objetivo más en mi vida. Gracias por inculcar en mí el ejemplo de esfuerzo y valentía, de no temer las adversidades porque Dios está conmigo siempre.

A mi hermanita Karen por llenarme de alegría día tras día, por su apoyo incondicional durante todo este proceso, por estar conmigo en todo momento.

A mis abuelitos, Herminia, Alfonso, Fabiola y Belisario por siempre impartirme consejos he inculcar en mí, desde pequeño, buenos valores para llegar a ser un buen profesional y persona.

AGRADECIMIENTO

Primeramente, agradezco a Dios por concederme salud y vida, ya que sin ello sería complicado llevar a cabo el presente proyecto.

A mis padres, a mi hermana y a mi familia por siempre estar ahí y dejar su ejemplo de honradez y dedicación. Gracias por el apoyo que me han ofrecido ya que sin él no tendría la fuerza y energía que me alienta a crecer como persona y como profesional.

A mis amigos Carlos, Ronal y Adrián, que siempre me ofrecieron un gran apoyo moral y humano los cuales fueron de mucha ayuda en los momentos más difíciles.

A los docentes que me acompañaron de inicio a fin en mi carrera universitaria.

A mi tutor, el Ing. Leonardo Torres por el tiempo dedicado y los conocimientos brindados.

A la Universidad Técnica de Ambato, en especial a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial por darme la oportunidad de estudiar la carrera que tanto me gusta y al fin cumplir mi sueño.

ÍNDICE GENERAL DE CONTENIDO

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO.....	iv
DERECHOS DE AUTOR.....	v
DEDICATORIA	vi
AGRADECIMIENTO.....	vii
RESUMEN EJECUTIVO	xx
ABSTRACT	xxi
CAPÍTULO I.....	1
MARCO TEÓRICO.....	1
1.1 Tema de investigación.....	1
1.2 Antecedentes investigativos	1
1.2.1 Contextualización del problema	1
1.2.2 Delimitación.....	3
1.2.2.1 De contenido	3
1.2.2.2 Delimitación espacial	3
1.2.2.3 Delimitación temporal.....	3
1.2.3 Justificación	3
1.2.4 Fundamentación teórica.....	4
1.2.2.1 Desarrollo de software	4
1.2.2.2 Desarrollo web	4
1.2.2.3 Aplicación web progresiva.....	4
1.2.2.4 Gestión administrativa	5
1.2.2.5 Plan de ventas.....	5
1.2.2.6 Control de ventas.....	5
1.2.2.7 Metodologías Ágiles en el Desarrollo de Aplicaciones Web Progresivas	5
1.2.2.7.1 Kanban	6
1.2.2.7.2 Scrum	7
1.2.2.7.3 Xtreme Programming.....	8
1.2.2.8 Herramientas tecnológicas para el desarrollo de la Aplicación Web Progresiva.....	10
1.2.2.8.1 Arquitectura de una Aplicación Web.....	10

1.2.2.8.2 Componentes de una aplicación web	11
1.2.2.8.3 Arquitectura en Capas	11
1.2.2.8.4 Base de datos.....	12
1.2.2.8.5 Tecnología Back-End.....	16
1.2.2.8.6 Tecnologías Front-End y Complementos.....	19
1.2.2.8.7 Typescript.....	21
1.2.2.8.8 Geolocalización.....	21
1.3 Objetivos.....	22
1.3.1 Objetivo general.....	22
1.3.2 Objetivos específicos	22
CAPÍTULO II	23
METODOLOGÍA	23
2.2 Métodos	31
2.2.1 Modalidad de investigación	31
2.2.2 Población y muestra.....	31
2.2.3 Recolección de información	31
2.2.4 Procesamiento y análisis de datos.....	43
CAPÍTULO III.....	45
RESULTADOS Y DISCUSIÓN.....	45
3.1 Análisis y discusión de resultados.....	45
3.1.1 Determinación del framework de desarrollo web.....	45
3.1.2 Determinación de la metodología de desarrollo del proyecto	46
3.2 Desarrollo de la propuesta	49
3.2.1 Fase I: Planificación.....	49
3.2.1.1 Levantamiento de información	49
3.2.1.2 Arquitectura de la aplicación	49
3.2.1.3 Definición de roles	51
3.2.1.4 Diseño de la base de datos	52
3.2.1.5 Diagrama de proceso para el control de los vendedores de la empresa Ambagro S.A.....	60
3.2.1.6 Diagrama de actividades de la aplicación	62
3.2.1.7 Historias de Usuario.....	66
3.2.1.8 Tareas	82
3.2.1.9 Valoración de historias de usuario	99
3.2.1.10 Estimación de historias de usuario.....	99

3.2.1.11 Plan de entrega	101
3.2.1.12 Plan de iteraciones.....	102
3.2.2 Fase II: Diseño	106
3.2.2.1 Metáfora	106
3.2.2.2 Iteraciones	107
Iteración 1.....	107
Iteración 2.....	113
Iteración 3.....	117
Iteración 4.....	122
Iteración 5.....	128
Iteración 6.....	133
3.2.3 Fase III: Codificación	138
3.2.4 Fase IV: Pruebas	165
3.2.4.1 Pruebas de aceptación	165
3.2.4.2 Pruebas de rendimiento	184
CAPÍTULO IV	187
CONCLUSIONES Y RECOMENDACIONES.....	187
4.1. Conclusiones.....	187
4.2. Recomendaciones	188
BIBLIOGRAFÍA.....	189
ANEXOS.....	193

ÍNDICE DE FIGURAS

Figura 1. Arquitectura de una aplicación web	10
Figura 2. Arquitectura de N capas	12
Figura 3. Conformidad con el control actual de procesos de venta	37
Figura 4. Inconvenientes al crear un pedido y enviarlo a su responsable.....	38
Figura 5. Conocimiento de los precios de los productos en venta.....	38
Figura 6. Existe un sistema para gestión de pedidos.....	39
Figura 7. La tecnología ayudaría a mejorar los procesos de compra y venta	40
Figura 8. Acceso a la información de los productos por medio de un dispositivo móvil	41
Figura 9. Notificar al área de bodega cuando se realiza un pedido	42
Figura 10. Implementar una aplicación web en la empresa Ambagro S.A.....	42
Figura 11. Arquitectura de la aplicación.....	50
Figura 12. Colecciones de Base de datos NoSQL	52
Figura 13. Modelo configuración perfil.....	53
Figura 14. Modelo Configuración permitir registro.....	53
Figura 15. Modelo configuración tamaño cliente	53
Figura 16. Modelo configuración número pedido	53
Figura 17. Modelo configuración número visita.....	54
Figura 18. Modelo configuración número cobranza.....	54
Figura 19. Modelo configuración número kilometraje	54
Figura 20. Modelo usuario	55
Figura 21. Modelo cliente	55
Figura 22. Modelo vendedores activos	56
Figura 23. Modelo categoría	56
Figura 24. Modelo nombre producto	56
Figura 25. Modelo empaque	57
Figura 26. Modelo producto.....	57
Figura 27. Modelo kilometraje.....	57
Figura 28. Modelo descripción pedido	58
Figura 29. Modelo pedido.....	58
Figura 30. Modelo visita	59
Figura 31. Modelo cobranza	59
Figura 32. Diagrama de procesos para el registro de actividades del usuario vendedor	60
Figura 33. Diagrama de procesos para el control y reportería de actividades del usuario vendedor.....	61
Figura 34. Inicio de sesión del sistema	62
Figura 35. Registro de usuarios del sistema.....	63
Figura 36. Vistas de los diferentes usuarios del sistema.....	63
Figura 37. Vista y módulos del perfil de usuario	64
Figura 38. Vista y módulos del usuario administrador del sistema	64
Figura 39. Vista y módulos del usuario bodeguero del sistema	65
Figura 40. Vista y módulos del usuario vendedor del sistema.....	65
Figura 41. Vista y módulos del usuario gerente del sistema.....	66

Figura 42. Estructura del sistema web con Angular	108
Figura 43. Interfaz de inicio de sesión	109
Figura 44. Interfaz de registros de usuarios.	109
Figura 45. Interfaz de restablecer contraseña de usuario	110
Figura 46. <i>Interfaz de inicio del usuario Administrador</i>	110
Figura 47. Interfaz de inicio del usuario bodeguero	111
Figura 48. Interfaz de inicio del usuario Vendedor	111
Figura 49. Interfaz de inicio del usuario gerente	112
Figura 50. Interfaz del perfil del usuario.....	112
Figura 51. Interfaz de edición de información del usuario	113
Figura 52. Interfaz de actualización del correo electrónico del usuario	114
Figura 53. Visualización de usuarios existentes en el sistema.....	114
Figura 54. Función de bloqueo de usuario.....	115
Figura 55. Función de desbloqueo de usuario.....	115
Figura 56. Interfaz de configuración de registro de usuarios.....	115
Figura 57. Interfaz de reportaría de usuarios existentes del sistema.....	116
Figura 58. Interfaz de reportaría de usuarios habilitados del sistema.....	116
Figura 59. Interfaz de reportaría de usuarios inhabilitados del sistema.....	117
Figura 60. Interfaz de listado de categorías	118
Figura 61. Añadir categoría	118
Figura 62. Editar categoría.....	119
Figura 63. Eliminar categoría.....	119
Figura 64. Listar nombre producto	120
Figura 65. Añadir nombre producto.....	120
Figura 66. Editar nombre producto	121
Figura 67. Eliminar nombre producto.....	121
Figura 68. Lista de empaques	122
Figura 69. Añadir empaque.....	123
Figura 70. Editar empaque	123
Figura 71. Eliminar empaque.....	124
Figura 72. Listar productos	124
Figura 73. Agregar producto.....	125
Figura 74. Editar producto	126
Figura 75. Eliminar producto.....	126
Figura 76. Ver pedidos realizados.....	127
Figura 77. Despachar pedidos realizados.....	128
Figura 78. Interfaz de ver pedidos marcados como despachados	129
Figura 79. Interfaz de ingreso de kilometraje inicial	129
Figura 80. Interfaz de inicio del módulo de crear pedidos.....	130
Figura 81. Interfaz del módulo Visita	130
Figura 82. Interfaz del módulo Cobranza	131
Figura 83. Interfaz para crear un pedido	131
Figura 84. Editar pedidos	132
Figura 85. Eliminar pedido	132
Figura 86. Interfaz de generar reporte de pedido.....	133
Figura 87. Reporte de pedido.....	134

Figura 88. Interfaz de generar reporte de visitas.....	134
Figura 89. Reporte de Visita	135
Figura 90. Interfaz de generar reporte de cobranzas.....	135
Figura 91. Reporte de Cobranza	136
Figura 92. Interfaz de generar reporte de kilometrajes	136
Figura 93. Reporte de Kilometraje.....	137
Figura 94. Conexión Firebase con Angular	138
Figura 95. Método crear un nuevo usuarios.....	139
Figura 96. Método Guardar usuarios	139
Figura 97. Método de inicio de sesión	140
Figura 98. Método de reestablecer contraseña.....	140
Figura 99. Método de obtener la autenticación de inicio de sesión de un usuario..	140
Figura 100. Método de obtener usuarios del sistema.....	140
Figura 101. Método para obtener un usuario por su ID	141
Figura 102. Método para actualizar un usuario.....	141
Figura 103. Método para obtener los usuarios deshabilitados del sistema	141
Figura 104. Método para obtener los usuarios habilitados en el sistema.....	142
Figura 105. Método para guardar token de usuario	142
Figura 106. Método para obtener la configuración de tipo de perfil de usuario.....	142
Figura 107. Método para cambiar la configuración del tipo de perfil de usuario...	143
Figura 108. Método para obtener la configuración de permitir registro de usuarios	143
Figura 109. Método para cambiar la configuración del permiso de registro de usuarios del sistema.....	143
Figura 110. Método para obtener la configuración de número de clientes del sistema	144
Figura 111. Método para cambiar la configuración del número de clientes del sistema	144
Figura 112. Método para obtener la configuración de token de actualización de clientes.....	144
Figura 113. Método para cambiar la configuración del token de actualización de clientes.....	144
Figura 114. Método para obtener la configuración de numero de pedidos del sistema	145
Figura 115. Método para cambiar la configuración del número de pedidos del sistema	145
Figura 116. Método para obtener la configuración de número de visitas del sistema	145
Figura 117. Método para cambiar la configuración del número de visitas del sistema	145
Figura 118. Método para obtener la configuración de numero de cobranzas del sistema	146
Figura 119. Método para cambiar la configuración del número de cobranza del sistema.....	146
Figura 120. Método para obtener la configuración de numero de kilometrajes del sistema.....	146

Figura 121. Método para cambiar la configuración del número de kilometraje del sistema.....	146
Figura 122. Método obtener categoría	147
Figura 123. Método guardar categoría.....	147
Figura 124. Método actualizar categoría.....	147
Figura 125. Método eliminar categoría.....	147
Figura 126. Método obtener nombre producto	148
Figura 127. Método guardar nombre producto	148
Figura 128. Método actualizar nombre producto.....	148
Figura 129. Método eliminar nombre producto	149
Figura 130. Método obtener empaque	149
Figura 131. Método guardar empaque	149
Figura 132. Método actualizar empaque.....	150
Figura 133. Método eliminar empaque	150
Figura 134. Método obtener producto.....	150
Figura 135. Método obtener producto por categoría.....	151
Figura 136. Método guardar producto	151
Figura 137. Método actualizar producto	151
Figura 138. Método eliminar producto	152
Figura 139. Método obtener kilometraje por ID	152
Figura 140. Método obtener kilometraje por fecha y vendedor.....	152
Figura 141. Método guardar kilometraje	153
Figura 142. Método actualizar kilometraje	153
Figura 143. Método obtener vendedores activos	153
Figura 144. Método guardar ID de vendedores activos.....	154
Figura 145. Método eliminar vendedores activos.....	154
Figura 146. Método obtener cliente	154
Figura 147. Método guardar cliente.....	154
Figura 148. Método actualizar cliente.....	155
Figura 149. Método eliminar cliente.....	155
Figura 150. Método obtener pedido.....	155
Figura 151. Método obtener pedido por su ID.....	156
Figura 152. Método obtener pedido por su cliente	156
Figura 153. Método obtener pedido por su numero.....	156
Figura 154. Método obtener pedido por una fecha y cliente	157
Figura 155. Método guardar pedido.....	157
Figura 156. Método actualizar pedido	157
Figura 157. Método eliminar pedido.....	158
Figura 158. Método cambiar estado de pedido	158
Figura 159. Método obtener descripción pedidos.....	158
Figura 160. Método guardar descripción pedidos.....	159
Figura 161. Método actualizar descripción pedidos	159
Figura 162. Método eliminar descripción pedidos.....	159
Figura 163. Método obtener visita por ID.....	160
Figura 164. Método obtener visita por su número.....	160
Figura 165. Método obtener visita por fecha y cliente	160

Figura 166. Método guardar visita.....	161
Figura 167. Método obtener cobranza	161
Figura 168. Método obtener cobranza por su número	161
Figura 169. Método obtener cobranza por fecha y cliente.....	162
Figura 170. Método guardar cobranza	162
Figura 171. Método obtener coordenadas geográficas	162
Figura 172. Guardianes de rutas de componentes.....	163
Figura 173. Método para imprimir documento	164
Figura 174. Método para imprimir documento	164
Figura 175. Resultados de prueba de rendimiento.....	184
Figura 176. Sugerencia de performance Lighthouse	185
Figura 177. Sugerencia de Best practices Lighthouse	185
Figura 178. Resumen de pruebas de rendimiento en Lighthouse	186

ÍNDICE DE TABLAS

Tabla 1. Precios plan Spark y Blaze para base de datos en tiempo real.....	18
Tabla 2. Precios plan Spark y Blaze para Almacenamiento en Firebase	18
Tabla 3. Precios plan Spark y Blaze para Cloud Firestore.....	19
Tabla 4. Precios plan Spark y Blaze para Autenticación de Usuarios Firebase.....	19
Tabla 5. Modelo entrevista encargado de Bodega	26
Tabla 6. Modelo entrevista Gerente	28
Tabla 7. Población que interviene en el sistema web.....	31
Tabla 8. Entrevista encargado de Bodega	32
Tabla 9. Entrevista Gerente	34
Tabla 10. Comparación de Framework.....	45
Tabla 11. Comparación de metodologías ágiles para el desarrollo de una PWA	46
Tabla 12. Metodología de desarrollo ágil elegida	48
Tabla 13. Definición de roles	51
Tabla 14. Plantilla de Historia de Usuario	66
Tabla 15. Historia de usuario - Establecer la estructura del sistema.....	67
Tabla 16. Historia de usuario - Ingreso al sistema web	68
Tabla 17. Historia de usuario - Restablecer contraseña	68
Tabla 18. Historia de usuario - Visualización de la pantalla de inicio.....	69
Tabla 19. Historia de usuario - Visualización del perfil de usuario.....	69
Tabla 20. Historia de usuario - Edición del perfil de usuario.	70
Tabla 21. Historia de usuario - Visualización de usuarios del sistema.....	70
Tabla 22. Historia de usuario – Funciones de bloqueo de usuario.....	70
Tabla 23. Historia de usuario - Configuración de registro de usuarios.....	71
Tabla 24. Historia de usuario – Reportes	71
Tabla 25. Historia de usuario - Listado de categorías.....	72
Tabla 26. Historia de usuario - Añadir categoría	72
Tabla 27. Historia de usuario - Editar categoría.....	72
Tabla 28. Historia de usuario - Eliminar categoría	73
Tabla 29. Historia de usuario - Listado de nombres productos.....	73
Tabla 30. Historia de usuario - Añadir nombre producto	73
Tabla 31. Historia de usuario - Editar nombre producto.....	74
Tabla 32. Historia de usuario - Eliminar nombre producto	74
Tabla 33. Historia de usuario - Listado de empaques	74
Tabla 34. Historia de usuario - Añadir empaque	75
Tabla 35. Historia de usuario - Editar empaque.....	75
Tabla 36. Historia de usuario - Eliminar empaque.....	75
Tabla 37. Historia de usuario - Listar productos.....	76
Tabla 38. Historia de usuario - Agregar producto.....	76
Tabla 39. Historia de usuario - Editar producto	76
Tabla 40. Historia de usuario - Eliminar producto.....	77
Tabla 41. Historia de usuario - Ver pedidos realizados	77
Tabla 42. Historia de usuario - Despachar pedido realizado.....	78
Tabla 43. Historia de usuario - Ver pedido marcados como despachados.....	78
Tabla 44. Historia de usuario - Ingresar kilometraje.....	79

Tabla 45. Historia de usuario - Vista de crear pedidos	79
Tabla 46. Historia de usuario - Vista de visita al cliente	80
Tabla 47. Historia de usuario - Vista de cobranza	80
Tabla 48. Historia de usuario - Crear pedido	80
Tabla 49. Historia de usuario - Editar pedido	81
Tabla 50. Historia de usuario - Eliminar pedido	81
Tabla 51. Historia de usuario - Crear Visita.....	81
Tabla 52. Historia de usuario - Crear cobranza.....	82
Tabla 53. Historia de usuario – Reportes	82
Tabla 54. Tarea - Establecer la estructura del sistema web.	83
Tabla 55. Tarea - Diseño de interfaz para el ingreso al sistema web.	83
Tabla 56. Tarea - Implementación de ingreso al sistema web.	83
Tabla 57. Tarea - Interfaz de restablecimiento de contraseña.	84
Tabla 58. Tarea - Implementación de restablecimiento de contraseña.	84
Tabla 59. Tarea - Interfaz de inicio del sistema	84
Tabla 60. Tarea - Implementación de la interfaz de inicio del usuario administrador.	85
Tabla 61. Tarea - Interfaz del listado y funciones de usuarios del sistema.	85
Tabla 62. Tarea - Implementación del listado y funciones de usuarios del sistema.	85
Tabla 63. Tarea - Diseño de funciones de bloqueo de usuario.	86
Tabla 64. Tarea - Implementación de funciones de bloqueo de usuario.	86
Tabla 65. Tarea - Interfaz de perfil de usuario.	86
Tabla 66. Tarea - Implementación del perfil de usuario.	87
Tabla 67. Tarea - Interfaz de edición de perfil de usuario.	87
Tabla 68. Tarea - Implementación de edición de perfil de usuario.	87
Tabla 69. Tarea - Interfaz de configuración de registro de usuario.	88
Tabla 70. Tarea - Implementación de configuración de registro de usuario.	88
Tabla 71. Tarea - Interfaz de reportes de usuarios del sistema.	88
Tabla 72. Tarea - Implementación de reportes de usuarios del sistema.	89
Tabla 73. Tarea - Interfaz de inicio del usuario bodeguero.	89
Tabla 74. Tarea - Implementación de pantalla de inicio del usuario bodeguero.	89
Tabla 75. Tarea - Interfaz de categorías de productos.	90
Tabla 76. Tarea - Implementación de categorías de productos.	90
Tabla 77. Tarea - Interfaz de nombres productos.....	90
Tabla 78. Tarea - Implementación de nombres productos.	91
Tabla 79. Tarea - Interfaz de empaques de productos.....	91
Tabla 80. Tarea - Implementación de empaques de productos.	91
Tabla 81. Tarea - Interfaz de productos.	92
Tabla 82. Tarea - Implementación de productos.....	92
Tabla 83. Tarea - Interfaz de los pedidos realizados.....	92
Tabla 84. Tarea - Implementación de pedidos realizados.	93
Tabla 85. Tarea - Interfaz de pedidos despachados.	93
Tabla 86. Tarea - Implementación de pedidos despachados.	93
Tabla 87. Tarea - Interfaz de inicio del usuario vendedor.	94
Tabla 88. Tarea - Implementación de pantalla de inicio del usuario vendedor.	94
Tabla 89. Tarea - Interfaz para crear pedidos.....	94

Tabla 90. Tarea - Implementación para crear pedidos.....	95
Tabla 91. Tarea - Interfaz de visita al cliente.....	95
Tabla 92. Tarea - Implementación de visita al cliente.....	95
Tabla 93. Tarea - Interfaz de cobranza.....	96
Tabla 94. Tarea - Implementación de cobranza.....	96
Tabla 95. Tarea - Interfaz de ingreso de kilometraje.....	96
Tabla 96. Tarea - Implementación de ingreso de kilometraje.....	97
Tabla 97. Tarea - Interfaz de inicio del usuario gerente.....	97
Tabla 98. Tarea - Implementación de la pantalla de inicio.....	97
Tabla 99. Tarea - Diseño de reportes de vendedores.....	98
Tabla 100. Tarea - Implementación de reportes de vendedores.....	98
Tabla 101. Estimación iteración 1.....	99
Tabla 102. Estimación iteración 2.....	99
Tabla 103. Estimación iteración 3.....	99
Tabla 104. Estimación iteración 4.....	100
Tabla 105. Estimación iteración 5.....	100
Tabla 106. Estimación iteración 6.....	101
Tabla 107. Plan de entrega.....	101
Tabla 108. Plan de iteraciones.....	103
Tabla 109. Planificación de iteración 1.....	107
Tabla 110. Planificación de iteración 2.....	113
Tabla 111. Planificación de iteración 3.....	117
Tabla 112. Planificación de iteración 4.....	122
Tabla 113. Planificación de iteración 5.....	128
Tabla 114. Planificación de iteración 6.....	133
Tabla 115. Prueba de aceptación historia 002.....	165
Tabla 116. Prueba de aceptación historia 003.....	165
Tabla 117. Prueba de aceptación historia 004.....	166
Tabla 118. Prueba de aceptación historia 005.....	166
Tabla 119. Prueba de aceptación historia 006.....	167
Tabla 120. Prueba de aceptación historia 007.....	167
Tabla 121. Prueba de aceptación historia 008.....	168
Tabla 122. Prueba de aceptación historia 009.....	168
Tabla 123. Prueba de aceptación historia 010.....	169
Tabla 124. Prueba de aceptación historia 011.....	169
Tabla 125. Prueba de aceptación historia 012.....	170
Tabla 126. Prueba de aceptación historia 013.....	170
Tabla 127. Prueba de aceptación historia 014.....	171
Tabla 128. Prueba de aceptación historia 015.....	171
Tabla 129. Prueba de aceptación historia 016.....	172
Tabla 130. Prueba de aceptación historia 017.....	172
Tabla 131. Prueba de aceptación historia 018.....	173
Tabla 132. Prueba de aceptación historia 019.....	173
Tabla 133. Prueba de aceptación historia 020.....	174
Tabla 134. Prueba de aceptación historia 021.....	174
Tabla 135. Prueba de aceptación historia 022.....	175

Tabla 136. Prueba de aceptación historia 023.....	175
Tabla 137. Prueba de aceptación historia 024.....	176
Tabla 138. Prueba de aceptación historia 025.....	176
Tabla 139. Prueba de aceptación historia 026.....	177
Tabla 140. Prueba de aceptación historia 027.....	177
Tabla 141. Prueba de aceptación historia 028.....	178
Tabla 142. Prueba de aceptación historia 029.....	178
Tabla 143. Prueba de aceptación historia 030.....	179
Tabla 144. Prueba de aceptación historia 031.....	179
Tabla 145. Prueba de aceptación historia 032.....	180
Tabla 146. Prueba de aceptación historia 033.....	180
Tabla 147. Prueba de aceptación historia 034.....	181
Tabla 148. Prueba de aceptación historia 035.....	181
Tabla 149. Prueba de aceptación historia 036.....	182
Tabla 150. Prueba de aceptación historia 037.....	182
Tabla 151. Prueba de aceptación historia 038.....	182
Tabla 152. Prueba de aceptación historia 039.....	183

RESUMEN EJECUTIVO

Actualmente, existen varias tecnologías las cuales han brindado numerosos beneficios a distintas empresas con el fin de optimizar y mejorar los procesos que se manejan interna y externamente. Un ejemplo de estas nuevas tecnologías que se han vuelto muy populares por sus múltiples beneficios son las Aplicaciones Web Progresivas. Las PWA (Progressive Web Apps) son una combinación entre una aplicación web y una aplicación nativa, juntando sus beneficios para brindar una experiencia amigable con los usuarios. Con el pasar del tiempo, su popularidad ha ido aumentando puesto que se la puede utilizar en un navegador web o instalarla en una computadora o celular inteligente sin la necesidad de tener un sistema operativo específico, dando así la ventaja de mantener y utilizar los equipos que tiene una empresa.

El presente proyecto ha sido desarrollado con el propósito de ayudar a la empresa Ambagro S.A. en los procesos involucrados en la parte de creación y despacho de pedidos generados por los vendedores de la empresa, así como también el control de los vendedores proponiendo un proceso el cual pueda cumplir con las expectativas de los requerimientos generados. Todo esto de la mano con los beneficios que brindan las Aplicaciones Web Progresivas.

El proyecto ha sido desarrollado en el framework Angular 12, el cual utiliza el lenguaje de programación TypeScript, mismo que cuenta con varias librerías útiles, como es el caso de @angular/pwa, la cual sirve para crear Aplicaciones Web Progresivas de una manera sencilla. La metodología empleada para el desarrollo del proyecto fue Xtreme Programming (XP) dado que por tamaño del proyecto y otros factores fue la metodología más adecuada. Como Backend se utilizó Firebase ya que cuenta con distintos servicios útiles para el desarrollo del sistema, como es el caso de la base de datos NoSQL Firestore.

Palabras clave: PWA, Angular, Firebase, TypeScript, gestión de pedidos, automatización, Service Worker, geolocalización.

ABSTRACT

Currently, there are several technologies which have provided numerous benefits to different companies in order to optimize and improve the processes that are handled internally and externally. An example of these new technologies that have become very popular for their multiple benefits are Progressive Web Applications. PWAs (Progressive Web Apps) are a combination of a web application and a native application, combining their benefits to provide a user-friendly experience. Over time, its popularity has been increasing since it can be used in a web browser or installed on a computer or smart phone without the need for a specific operating system, thus giving the advantage of maintaining and using the equipment that a company has.

This project has been developed with the purpose of helping the company Ambagro S.A. in the processes involved in the creation and dispatch of orders generated by the company's vendors, as well as the control of the vendors by proposing a process which can meet the expectations of the requirements generated. All this hand in hand with the benefits provided by Progressive Web Applications.

The project has been developed in the Angular 12 framework, which uses the TypeScript programming language, which has several useful libraries, such as @angular/pwa, which is used to create Progressive Web Applications in a simple way. The methodology used for the development of the project was Xtreme Programming (XP), since due to the size of the project and other factors it was the most appropriate methodology. Firebase was used as Backend since it has different useful services for the development of the system, as is the case of the NoSQL Firestore database.

Keywords: PWA, Angular, Firebase, TypeScript, orders management, automation, Service Worker, geolocation.

CAPÍTULO I

MARCO TEÓRICO

1.1 Tema de investigación

Aplicación web progresiva (PWA) utilizando una base de datos NoSQL para el control de vendedores y gestión de pedidos de la empresa Ambagro S.A.

1.2 Antecedentes investigativos

1.2.1 Contextualización del problema

En la actualidad, en el mundo se habla de las nuevas preferencias en aplicaciones web las mismas que puedan dar un aporte característico tanto en el ámbito de productividad como de eficacia de las entidades o empresas, aportando herramientas útiles en el trabajo del día a día [1].

Un artículo de la revista Sistematizar procesos en tu empresa expresa que “Las empresas en vías de crecimiento deben luchar cada día por ir de la mano con los avances tecnológicos y adaptarse a ellos, con el fin de acelerar sus procesos y por supuesto, mantener competitividad en el mercado.” [2], ya que las nuevas tecnologías están presentando a los negocios una nueva visión para poder dedicar sus servicios a su personal y mejorar la productividad para generar mayores ingresos.

Un ejemplo para mencionar sobre estos avances tecnológicos son las aplicaciones web progresivas que permiten que un aplicativo web se comporte como una aplicación nativa en un teléfono inteligente, entregando la experiencia del móvil hasta la web [3]. Así mismo, otro caso a mencionar son las bases de datos NoSQL ya que tienen un esquema flexible para crear aplicaciones modernas gracias a su facilidad de desarrollo, funcionalidad y escalabilidad [4].

El objetivo de las Aplicaciones Web Progresivas es hacer un uso óptimo de las capacidades en constante evolución de los navegadores web. Una de esas capacidades avanzadas es almacenar datos en un navegador para la funcionalidad de la aplicación fuera de línea. Los navegadores ahora usan varias tecnologías NoSQL y NewSQL para este propósito [5].

MakeMyTrip una empresa de reservas de viajes originaria de la India menciona que “No todos los recursos tecnológicos son iguales, en especial en la India” [6], es por ello que querían asegurarse de que su sitio web fuera fluido, seguro y confiable para sus clientes, independientemente de la tecnología disponible. Es por ello que lanzaron una aplicación web progresiva ya que esto sería clave para ofrecer experiencias móviles fluidas a los clientes que viven fuera de las principales ciudades o aquellos que usan dispositivos de gama baja.

En el Ecuador, WPA (Progressive Web Apps) se está posicionando de una manera rápida ya que hoy en día las personas buscan tener una aplicación móvil sin la necesidad de entrar a una plataforma de distribución digital con el riesgo de descargar una aplicación nativa llena de muchos términos y condiciones que la mayoría de usuarios no leen [7], para poder utilizarla al 100%, o muchas veces descargar estas aplicaciones sin saber que pertenezcan o no a una entidad que brinda servicios para sus consumidores.

La gestión de pedidos es una de las fases fundamentales en una cadena de suministros ya que allí se garantiza el servicio al cliente. Hay que tomar en cuenta que las etapas que conforman una cadena de suministro están interconectadas, por lo tanto, se deben realizar todas sus fases correctamente [8]. En la gestión de pedidos se cometen varios errores ya que las empresas y sus empleados están confiados que las etapas previas a la gestión de pedidos son suficientes para realizar un buen despacho de sus pedidos. Algunos de los errores más comunes en la gestión de pedidos son: pedidos no preparados por falta de información, etiquetado erróneo de pedidos listos para su expedición, información inconclusa sobre los productos ofertados, falta de comunicación con el encargado de disponer los pedidos de la empresa [9].

En la ciudad de Ambato, en la empresa Ambagro S.A., la gestión de pedidos se la realiza de forma manual, específicamente en un formulario denominado ‘Nota de Pedido’ en el cual, al registrar la información referente al pedido de forma manual, es propenso a errores, además de que una vez tomada la información debe ser enviada por algún medio o entregada físicamente en bodega lo cual provoca que el despacho de productos sea lento, confuso y tardado, provocando que las ventas de un período de tiempo determinado disminuyan o que los compradores dejen de adquirir productos en la empresa.

1.2.2 Delimitación

1.2.2.1 De contenido

Área: Software

Línea de investigación: Desarrollo de Software

Sub línea de investigación: Aplicaciones Web

1.2.2.2 Delimitación espacial

Ambagro S.A. ubicada en el Parque Industrial, Calle 5 y Avenida D.

1.2.2.3 Delimitación temporal

La presente investigación se desarrollará en el periodo académico abril – agosto 2022.

1.2.3 Justificación

El presente proyecto de investigación es importante porque mejorará el control de las ventas de la empresa Ambagro S.A. ya que reducirá el tiempo y los errores de los pedidos creados por los vendedores, así como también aumentará de una manera significativa el proceso mencionado anteriormente y así se tendrá una mayor eficacia y eficiencia en la calidad de los pedidos realizados, además se tendrá una gestión de vendedores la cual permitirá verificar si estos cumplen con sus respectivas obligaciones dentro del horario de trabajo y también se podrá revisar que los recursos de la empresa sean utilizados correctamente por cada vendedor.

Es de interés porque se podrá saber la ubicación exacta de los vendedores al momento de realizar un pedido, hacer una visita a un cliente o cobrar dinero. Es importante porque se investigarán nuevas funcionalidades y beneficios al momento de utilizar las nuevas tecnologías que brinda un navegador web como son los Service workers y las nuevas bases de datos no relacionales que permiten tener una mayor escalabilidad en un sistema desarrollado. Es original porque se investigarán dos nuevas tecnologías que no son tan nombradas ya que no se han implementado masivamente por el temor de que los usuarios más antiguos dejen a un lado las nuevas aplicaciones web al no saberlas utilizar. El presente proyecto de investigación beneficiará a los vendedores y al área de bodega de la empresa ya que son los actores principales del desarrollo de la venta de productos agrícolas.

Factibilidad Técnica: El investigador tiene las habilidades, conocimientos y experiencias necesarias para poder realizar el presente proyecto, así como también los recursos tecnológicos haciendo referencia a las herramientas tecnológicas.

Factibilidad Operativa: El investigador tiene la aprobación y asistencia del gerente general de la institución, así como también del encargado de bodega el cual brindará la información necesaria para realizar el presente proyecto.

Factibilidad Económica: El financiamiento del proyecto será asumido por el investigador refiriéndose al estudio, análisis y tiempo empleado.

1.2.4 Fundamentación teórica

1.2.2.1 Desarrollo de software

El desarrollo de software estudia los componentes necesarios para la creación, gestión, mantenimiento y testeado de software computacional; a la persona encargada de esto se le llama programador y este es un especialista en informática, capaz de elaborar sistemas informáticos (paquetes de software), así como de implementarlos y ponerlos a funcionar, utilizando uno o varios lenguajes de programación [10].

1.2.2.2 Desarrollo web

El desarrollo web es la programación necesaria para la construcción del sitio web. Se divide en dos partes que pueden estar o no conectadas, la parte del cliente y la parte del servidor. En la parte del cliente se refiere a HTML y CSS, código básico para creación de páginas web, y JavaScript y DOM, para la interacción con el usuario. En la parte del servidor se trabaja con código más complejo, como es PHP, ASP.NET, JSP, etc. [10].

1.2.2.3 Aplicación web progresiva

Una aplicación web progresiva está a medio camino de una app 'corriente', aquella que necesita instalarse en el dispositivo, y de una web que se ejecuta dentro del navegador. Las aplicaciones web progresivas ahorran espacio, pueden utilizarse en cualquier Android y facilitan el uso de plataformas que a lo mejor no tienen aplicación 'corriente' [11].

1.2.2.4 Gestión administrativa

“Es un proceso que comprende determinadas funciones y actividades laborales que los gestores deben llevar a cabo a fin de lograr los objetivos de la empresa. Los gestores son las personas que asumen la responsabilidad principal por la realización del trabajo en una organización” [13].

1.2.2.5 Plan de ventas

“El plan de ventas de una empresa es un documento en el que quedan prudentemente reflejadas las ventas de una compañía para un periodo determinado de tiempo. En este texto quedan definidos los principales escenarios en los que puede desarrollarse la interacción comercial en un futuro cercano” [14].

1.2.2.6 Control de ventas

Según la empresa Kyocera, el control de ventas es una tecnología diseñada para alinear las acciones de los vendedores con los objetivos organizacionales. Es una herramienta que ayuda a los gerentes a monitorear el desempeño del equipo o verificar si los empleados están cumpliendo sus actividades propuestas [15].

1.2.2.7 Metodologías Ágiles en el Desarrollo de Aplicaciones Web Progresivas

En las empresas, el desarrollo de Software se ha hecho muy popular por las diferentes ventajas que brinda, una de ellas sería a nivel comercial ya que gracias a esto se ha dado a conocer mundialmente. Debido a que las diferentes tecnologías avanzan a pasos agigantados, el Software debe adaptarse a las necesidades de los usuarios para cumplir con sus expectativas. Debido a estos cambios nacen nuevos conceptos los cuales permiten agilizar los procesos para la creación de Software los cuales buscan mantener y mejorar la interacción entre el desarrollador y el cliente con el objetivo de que el desarrollador intervenga mucho más [16].

Para comprender mejor las metodologías ágiles se tiene un manifiesto ágil el cual es un documento en el que está resumida la filosofía de este enfoque de desarrollo. Aquí se puede observar la ayuda que se va a brindar al desarrollador y al cliente en el proceso de creación del sistema y su comunicación, así como también la relación entre el tiempo y costo del desarrollo de software [16].

A continuación, se detallan las metodologías de desarrollo ágil más conocidas:

1.2.2.7.1 Kanban

Esta metodología es muy popular al momento de aplicar un desarrollo de software. Kanban utiliza tarjetas para administrar visualmente el progreso de cada tarea, todo esto dentro de un tablero dividido en tres columnas. Cada columna ayuda a observar que se está haciendo, que se ha realizado y que se realizará en el futuro [17].

Kanban está conformado por varios fundamentos que deben ser realizados de forma apropiada para obtener el resultado esperado. El objetivo de esta metodología es mantener un balance en el flujo de trabajo para evitar un exceso de responsabilidad en una sola persona. Gracias a esto, algunas ventajas que da Kanban es: gestionar el tiempo de cada miembro de nuestro equipo, identificar los errores y verlos como una oportunidad de mejora, desarrollar un producto de calidad rápidamente, colocar fechas límite para cada tarea asignada a cada persona [17].

Roles en Kanban

Existen dos roles en la metodología Kanban los cuales son:

- **Service request manager**

Es el responsable de administrar la demanda y los requisitos dentro del sistema Kanban, guiando la relación que existe entre los stakeholders y a la vez impulsar la transparencia dentro del sistema respecto a la importancia del trabajo [18].

- **Service delivery manager**

Es el encargado del trabajo que se realiza en el sistema Kanban y en ocasiones, ciertas tareas. También hace más fácil las reuniones que se realizan en esta metodología [18].

Reuniones en Kanban

Esta metodología cuenta con siete reuniones para poder gestionar el trabajo a nivel organizacional. Todas ellas están conectadas para tener una gestión del flujo de trabajo [18].

- Replenishment meeting
- Daily kanban
- Service delivery meeting

- Delivery planning meeting
- Strategy review
- Operations review
- Risk review

1.2.2.7.2 Scrum

Scrum es un proceso el cual permite realizar varias tareas de forma regular con el fin de trabajar cooperativamente con todos los miembros de un equipo. Esta metodología de trabajo ayuda a lograr el mejor resultado de un proyecto determinado. Las experiencias que se aplican con la metodología scrum se retroalimentan las unas con otras y la incorporación de estas tiene su origen en un estudio de cómo hay que coordinar a los equipos para ser potencialmente competitivos [19].

En esta metodología los miembros del equipo deben entregar sus resultados parcial y regularmente al trabajo final por superioridad y demanda del desarrollo, es por esto que la metodología Scrum es adecuada para proyectos extensos y con una cierta complejidad, sus requisitos cambian de manera inesperada y la flexibilidad es una parte importante. La metodología Scrum se la utiliza cuando se quiere ver resultados a corto plazo o cuando algo no está claro y poco definido [19].

Scrum cuenta con diferentes fases las cuales hacen posible el éxito del proyecto, estas fases son:

1. Planificación: Product Backlog
2. Ejecución: Sprint
3. Control: Burn Down

Roles en Scrum

En la metodología Scrum se pueden encontrar tres roles principales los cuales son obligatorios para poder obtener un producto final del proyecto.

- **Product Owner**

Este rol se encarga de comprender las necesidades y prioridades de los stakeholders, incluyendo a los usuarios y clientes. Garantiza la comunicación, y la perspectiva de aceptación y asegurar que se cumplan [19].

- **Scrum Master**

Es el cual debe tomar una actitud de liderazgo ya que facilita todo lo necesario para que se lleve a cabo el proyecto y motiva a los miembros del equipo [19].

- **Scrum Team**

Es el equipo encargado de desarrollar las historias de usuario en la lista de pendientes del Sprint para crear los entregables del proyecto. Este equipo es autogestionado y multifuncional ya que tiene las habilidades necesarias para realizar su trabajo [19].

1.2.2.7.3 Xtreme Programming

La metodología XP (Xtreme Programming) tiene como objetivo centrarse en aumentar las relaciones interpersonales del grupo de desarrollo de un proyecto como clave del éxito mediante el trabajo en equipo, practica y buen ambiente laboral. Esta metodología se enfoca en la retroalimentación continua que existe entre el cliente y el grupo de desarrolladores de Software por eso es adecuada para proyectos con requisitos variantes y ambiguos [20].

Algunas de las características que brinda la metodología XP son: solución inmediata a los cambios repentinos, constante comunicación entre el cliente y el equipo de desarrollo, para tener éxito en el proyecto hay que tener muy en cuenta los requisitos del cliente y el trabajo del equipo, el cronograma de actividades donde se plantean las tareas del equipo debe ser abierto y flexible [20]. Hay que tomar en cuenta que esta metodología es muy arriesgada ya que el éxito o fracaso del proyecto depende únicamente del equipo de desarrolladores, para asegurar el éxito se deben seguir los siguientes principios:

- En el equipo de programación, sus miembros deberán trabajar en parejas para poder ayudarse mutuamente si existiese algún inconveniente.
- El equipo de programación debe tener una buena relación para poder comunicarse sin problemas y llegar a soluciones precisas.
- El cliente debe ser un miembro más del equipo de desarrollo para que brinde los requerimientos directamente al equipo y no a terceras personas [16].

Roles en XP

Los roles en la metodología XP son:

- **Tracker**
Es el encargado del seguimiento del proyecto, además, proporciona retroalimentación al equipo de trabajo [21].
- **Ciente**
Es el encargado de diseñar las historias de usuario y las pruebas de funcionalidad que se llevan a cabo en el proyecto [21].
- **Programador**
Es el miembro más importante ya que es el encargado de escribir el código de programación del producto final de la investigación [21].
- **Coach**
Es el encargado de ver que todos los miembros del equipo realicen sus tareas correctamente [21].
- **Manager**
Es el encargado de crear y organizar las reuniones internas. También proporciona información importante para que el equipo realice sus actividades sin ningún problema [21].
- **Tester**
Es el encargado de ejecutar las pruebas funcionales sobre el producto final para poder evaluar el nivel de funcionalidad del producto final [21].

La metodología XP cuenta con diferentes fases las cuales hacen posible que el proyecto sea un éxito.

- **Fase de exploración**
En esta primera fase se busca entender lo que el cliente necesita, es decir recaudar lo necesario para poder desarrollar el producto.
- **Fase de planificación**
En esta fase se estima el esfuerzo empleado para la creación del Software.
- **Fase de iteraciones**
Aquí se crea la solución propuesta del proyecto.

- **Fase de puesta en producción**

Esta última fase se hace la entrega del producto final al cliente.

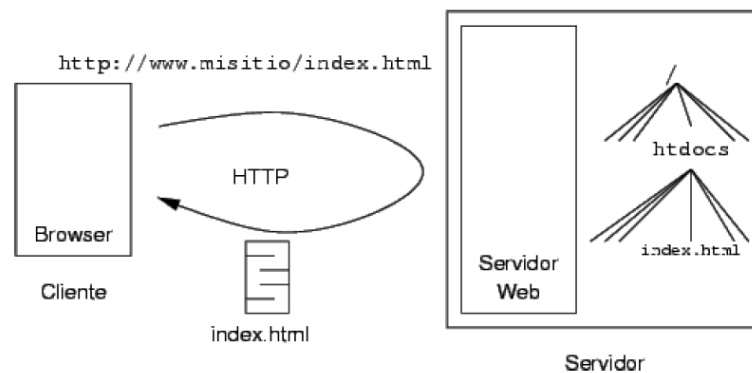
1.2.2.8 Herramientas tecnológicas para el desarrollo de la Aplicación Web

Progresiva

1.2.2.8.1 Arquitectura de una Aplicación Web

Una aplicación web es proporcionada por un servidor web y usada por usuarios que se conectan desde cualquier parte a través de un navegador web o browser. Este tipo de aplicaciones se basan en una arquitectura cliente/servidor la cual está conformada por tres componentes principales: uno o más clientes, conexión de red y un servidor web [22].

Figura 1. Arquitectura de una aplicación web



Elaborado por: [22]

Un servidor web es el encargado de repartir páginas de información formateada a los clientes que lo soliciten. Las peticiones son realizadas a través de una conexión a internet, es por ello por lo que se usa el protocolo HTTP. Solicitada dicha petición al servidor web este busca la página web en su sistema de archivos y la envía de vuelta al navegador web o browser que la solicito.

Las páginas web son elementos principales de una aplicación web. Los navegadores web solicitan estas páginas webs llenas de información a los servidores web en los que se encuentran alojadas. En la mayoría de los casos, las páginas web abarcan código HTML y scripts dinámicos los cuales son ejecutados por el servidor web antes de ser

enviados al navegador que solicito la página web. Entregada la página web al navegador web la conexión existente entre éste y el servidor web se perderá. Muchas de las páginas webs son dinámicas (PHP, ASP, etc.) y están asociadas lógicamente para dar un servicio al usuario [23].

1.2.2.8.2 Componentes de una aplicación web

Los componentes de una aplicación web son tres:

1. Lógica de negocio

Esta es la parte más importante de la aplicación web ya que aquí se establece un conjunto de operaciones y procesos los cuales sirven para proveer servicios.

2. Administración de los datos

Aquí se realiza la manipulación de la base de datos y archivos utilizados.

3. Interfaz

Los usuarios acceden y visualizan un sistema web mediante un browser, celular, etc.

1.2.2.8.3 Arquitectura en Capas

Las aplicaciones web están conformadas por modelos de capas, una capa simboliza un elemento que procesa o trata información. Estas capas separan la funcionalidad del sistema en factores diferentes. Las capas se pueden encontrar en un mismo computador o en un conjunto de computadores, cada nivel que conforma esta arquitectura tiene diferentes responsabilidades y abstracciones de la capa inferior [24].

Los tipos más comunes de arquitecturas en capas son:

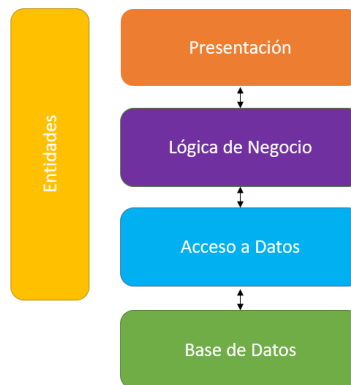
1.2.2.8.3.1 Modelo de dos capas

Toda la información pasa por dos capas entre la interfaz y la administración de los datos [24].

1.2.2.8.3.2 Modelo de N capas

Toda la información pasa por varias capas, dependiendo de los requerimientos del sistema. El más habitual es el modelo de tres capas [24].

Figura 2. Arquitectura de N capas



Elaborado por: [24]

Capa entidades: esta capa es la encargada de almacenar las entidades del negocio, es decir las clases que representaran los objetos. Estas entidades también sirven para comunicarnos entre las diferentes capas [24].

Capa Acceso a datos: en esta capa se encontrarán todos los métodos que se utilizan para leer y escribir datos de la base de datos [24].

Capa Lógica de Negocio: aquí se encuentran las validaciones del negocio, es decir los requerimientos y reglas del usuario [24].

Capa Presentación: en esta capa se crea la interacción entre el usuario y el sistema web [24].

1.2.2.8.4 Base de datos

1.2.2.8.4.1 Base de datos SQL

Una base de datos SQL es una base de datos relacional la cual utiliza lenguaje SQL el cual sirve para realizar consultas de forma estructurada, ejecutar tareas con los datos y recuperar datos de una base de datos. Los comandos SQL más comunes para realizar dichas tareas son: Select, Insert, Update, Delete y Create. Un sistema de base de datos relacional tiene una o más tablas las cuales se identifican de forma única por sus nombres y están formadas por columnas y filas. Las columnas tienen el nombre de la columna, tipos de datos y los atributos que se coloquen, en las filas se encuentran los datos de las columnas [25].

Al momento de crear una base de datos se crean tablas las cuales están relacionadas para tener consistencia y una estructura adecuada en los datos. Estas relaciones deben estar representadas en la base de datos, hay que tomar en cuenta que para obtener nuevamente los datos almacenados se tendrá que implementar consultas JOIN para obtener la información requerida. Existen distintos tipos de relaciones en una base de datos, algunos de ellos son: relación uno a uno, una a muchas y muchas a una, muchas a muchas y relación de autorreferencia.

Ventajas:

- **Estándares bien definidos:** Todos los procesos que se ejecuten deben permanecer en el margen de los estándares planteados por SQL [25].
- **Atomicidad:** Si se va a realizar un cambio en la base de datos se tiene que seguir la estructura adecuada y verificar si se necesita realizar modificaciones totales o parciales para que se apliquen los cambios, caso contrario no se guardara ningún cambio [25].
- **Estructura simple:** el lenguaje que se implementa en este tipo de base de datos es algo sencillo ya que son palabras que los humanos empleamos día a día [25].
- **Portabilidad:** SQL se lo puede utilizar en computadoras, laptops, celulares y servidores [25].
- **Experiencia y madurez:** El gran tiempo que se ha utilizado e implementados las bases de datos SQL han generado la aceptación de varios usuarios, así como herramientas que facilitan aún más su uso [25].

Desventajas:

- **Dificultad de crecimiento:** Cuando estas bases de datos comienzan a crecer en volumen, el almacenamiento y costo crecen bastante, debido a que se necesita mayor capacidad en el Hardware lo que conlleva un mayor gasto [25].
- **Dificultad en la interfaz:** Las interfaces de los sistemas que utilizan una base de datos SQL son un poco más complejas al momento de aumentar o disminuir información [25].

- **Cambios en la estructura:** Es más complicado alterar la base de datos previamente creada ya que se debe mantener la propiedad ACID [25].
- **Complejidad en la instalación:** este tipo de Base de datos requiere ciertos requisitos de Hardware para poder ser instaladas, así como también de Sistema Operativo [25].

1.2.2.8.4.2 Base de datos NoSQL

Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumple con el diseño entidad-relación. Este tipo de bases de datos son adaptables al momento de guardar datos, es decir, no contiene un esquema de tablas fijo, no se utilizan operaciones de tipo JOIN y tampoco se mantiene la propiedad ACID (atomicidad, consistencia, aislamiento y durabilidad), es por ello que su rendimiento es excelente en escalado horizontal. Los modelos no relacionales son un diseño de gestión de base de datos que varía del modelo clásico relacional en múltiples elementos, es por esto que no existe un lenguaje de consultas [26].

Las bases de datos NoSQL aparecieron como consecuencia del aumento del volumen de los datos e información, suceso que llegaría gracias a la llegada de la web 2.0, hasta antes de esto, solo existía información reducida de un grupo de empresas las cuales podían subir su información a la nube. Con la llegada de las redes sociales, tales como Facebook, Instagram, WhatsApp, Twitter, YouTube, etc., más personas tenían la posibilidad de compartir información a través de la nube, provocando así un aumento de los datos [26].

Este tipo de base de datos nace con el fin de dar una solución al gran incremento de los datos que se han ido generando estos años ya que por consecuencia de ello se exigen mayores requerimientos de procesamiento y análisis a gran escala de estas inmensas cantidades de datos.

Ventajas:

- **Flexibilidad en sus datos:** La flexibilidad en los datos brinda mayor libertad para poder modelar los datos existentes ya que no existen esquemas estrictos [27].

- **Escalabilidad Horizontal:** Se puede agregar nodos a los sistemas para mejorar su rendimiento sin la necesidad de reestructurar nuevamente la base de datos [27].
- **Sencillo para realizar consultas:** Las consultas se las realiza de una forma normal, sin la necesidad de saber códigos amplios y tediosos [27].
- **Usa pocos recursos:** Las bases de datos NoSQL se las puede implementar en equipos con poco Hardware [27].
- **Modelo de concurrencia débil:** Como no se mantiene la propiedad ACID, las instrucciones son consideradas una transacción [27].
- **Uso eficaz de los recursos:** Usa al máximo las nuevas tecnologías como los discos sólidos, memorias RAM y los sistemas distribuidos [27].
- **Gestión de grandes cantidades de datos:** Los gestores NoSQL se manejan muy bien con grandes cantidades de datos por que trabajan con una estructura distribuida [27].

Desventajas:

- **Herramientas GUI:** La gran mayoría de bases de datos NoSQL no contienen una interfaz gráfica lo que hace más complejo su uso [27].
- **Poca documentación:** No existe mucha documentación con respecto a este tipo de base de datos ya que son nuevas [27].
- **Atomicidad:** a veces la información no es consistente, puede ser diferente en cada uno de los nodos existentes en la base de datos NoSQL [27].
- **Estándares en el lenguaje:** Cada motor de base de datos NoSQL tiene un criterio definido, es por ello que el lenguaje varía [27].

1.2.2.8.4.3 Diferencias entre bases de datos SQL y NoSQL

La mayor diferencia que existe entre estos dos tipos de bases de datos es que en la base de datos NoSQL no se utilizan estructuras de datos en forma de tablas y tener que relacionarlas entre ellas, simplemente brindan un diseño mucho más adaptable y fácil de manipular [26].

1.2.2.8.5 Tecnología Back-End

1.2.2.8.5.1 Firebase

Firebase es una plataforma digital la cual está a cargo la empresa Google desde el año 2014. Esta plataforma contribuye con el desarrollo de aplicaciones de distintos tipos gracias a su facilidad de uso para las diferentes plataformas como son iOS, Android y Web. Firebase tiene diferentes funcionalidades las cuales contribuyen a que los desarrolladores realicen su trabajo de una manera más fácil y segura [28].

1.2.2.8.5.2 Características de Firebase

Algunas de las características más relevantes de esta plataforma digital son:

- **Desarrollo:** mediante las funciones que tiene Firebase se pueden crear aplicaciones en un corto tiempo optimizando su desarrollo y mejorando su calidad [28].
- **Analítica:** se puede monitorear el rendimiento de las aplicaciones mediante métricas analíticas en las cuales se puede observar usuarios activos, ingresos monetarios, uso de la base de datos en diferentes tiempos, etc. [28].
- **Poder de crecimiento:** se puede administrar de una mejor manera los usuarios de las aplicaciones, así como también invitar a nuevos usuarios por medio de invitaciones o notificaciones personalizadas [28].
- **Monetización:** Firebase implemento AdMob la cual es una empresa de publicidad móvil la cual permite generar ingresos mediante las aplicaciones desarrolladas [28].
- **Rapidez:** utilizar Firebase es muy fácil y rápido ya que con la ayuda de su API que esta sostenida en un solo SDK se pueden realizar todas las tareas con sencillos comandos [28].
- **Agilidad:** se puede crear varias aplicaciones que sean para distintas plataformas sin la necesidad de abandonar Firebase ya que éste ofrece aplicaciones multiplataforma gracias a sus APIs incorporadas a SDK individuales para iOS, Android y JavaScript [28].

1.2.2.8.5.3 Servicios de Firebase

Base de datos

Firestore trabaja con una base de datos NoSQL la cual es la encargada de almacenar todos los datos de los sistemas desarrollados. Cuando se realiza una operación CRUD se tiene la ventaja de notificar a los usuarios de la aplicación que se ha realizado un cambio a los datos del sistema, también se puede realizar actualizaciones en tiempo real [28].

Realtime Database

Es la primera base de datos que implemento Firebase. Esta base de datos está clasificada como eficiente y de baja latencia para el desarrollo de aplicaciones web y móviles las cuales usan la sincronización de datos [28].

Almacenamiento de archivos

Servicio de Storage que permite almacenar archivos los cuales serán utilizados para mostrar a los usuarios finales [28].

Cloud Firestore

Es la segunda versión de la base de datos de Firebase la cual está compuesta por las mejores características de las bases de datos en tiempo real más un nuevo realce en los modelos de datos para poder mejorar las velocidades de consulta [28].

Autenticación

Firestore propone servicios de inicio de sesión y registro para los usuarios de los sistemas desarrollados de diferentes formas incluidas direcciones de correo electrónico, número de celular, redes sociales como Facebook, Twitter, Google, GitHub y Apple, de forma anónima, etc. [28].

Valores de pago

Firestore cuenta con dos planes, el primero que es gratis, llamado Spark, el cual tiene limitaciones en varios aspectos, este plan es usado por empresas que son pequeñas y empiezan a utilizar los servicios de Firestore. El segundo plan llamado Blaze cuenta con todas las características que ofrece la plataforma digital, el pago de este plan es

acorde al uso que le demos a Firebase, dependiendo de que característica se estén utilizando, los valores irán variando.

A continuación, se detallan algunas características de Firebase con sus respectivos precios de acuerdo con el plan seleccionado [29].

Tabla 1. Precios plan Spark y Blaze para base de datos en tiempo real

Base de datos en tiempo real	Spark	Blaze
Conexiones simultáneas	100	200k/base de datos
GB almacenados	1 GB	5 \$/GB
GB descargados	10 GB/mois	1 \$/GB
Bases de datos por proyecto	1	Multiple

Elaborado por: [29]

Tabla 2. Precios plan Spark y Blaze para Almacenamiento en Firebase

Almacenamiento	Spark	Blaze
GB almacenados	5 GB	0,026 \$/GB
GB descargados	1 GB/día	0.12 \$/GB
Operaciones de carga	20K/día	0.05 \$/10k
Operaciones de descarga	50K/día	0.004 \$/10k
Cubos por proyecto	1	Múltiple

Elaborado por: [29]

Tabla 3. Precios plan Spark y Blaze para Cloud Firestore

Cloud Firestore	Spark	Blaze
GB almacenados (base de datos)	1 Go au total	0.18 \$/GB
Escrituras	20K/día	\$0,18/100K operaciones
Lecturas	50K/día	\$0,06/100K operaciones
Eliminaciones	20K/día	\$0,02/100K operaciones

Elaborado por: [29]

Tabla 4. Precios plan Spark y Blaze para Autenticación de Usuarios Firebase

Autenticación	Spark	Blaze
Primeras 10,000 autenticaciones	Gratis	Gratis
Más de 10.000 autenticaciones por mes en EE. UU., Canadá e India	N/A	\$0,01 por autenticación
Más de 10.000 autenticaciones por mes para otros países.	N/A	\$0,06 por autenticación

Elaborado por: [29]

1.2.2.8.6 Tecnologías Front-End y Complementos

Existen varias tecnologías las cuales se pueden emplear para el desarrollo del Front-End, las más populares y utilizadas en los últimos tiempos son tres: React, Angular y Vue.js. Cada una de estas tecnologías brinda diferentes herramientas que facilitan el desarrollo de los sistemas, pero una de las características a resaltar es que facilitan considerablemente la creación de Aplicaciones Web Progresivas.

1.2.2.8.6.1 React

Es una librería de código abierto de JavaScript para el desarrollo de tecnologías Front-End. Fue desarrollada por Facebook quien mantiene una sociedad de desarrolladores independientes y compañías que utilizan esta tecnología. Hoy por hoy, existen muchas empresas que utilizan React para desarrollar sus aplicaciones, entre ellas se puede mencionar a Facebook, Instagram, el cliente web de WhatsApp, AirBnb, Uber, Netflix, Twitter, Reddit, etc. [30].

Como se mencionó anteriormente, React es una librería y al tratarse de esto se necesitaría implementar otras librerías para poder completar las necesidades de nuestra aplicación web, esto puede ser una ventaja o desventaja para la persona que lo utilice ya que un Framework como Angular brinda todas las herramientas para poder crear nuestras aplicaciones web, pero no ofrece muchas librerías como React, mientras que React es una librería que cubre necesidades específicas [30].

1.2.2.8.6.2 Angular

Es un Framework de código abierto desarrollado por Google para desarrollar aplicaciones de una sola página, SPA (Single Page Application). Algo que caracteriza a Angular es que divide el Backend del Frontend en la aplicación lo que permite tener una estructura adecuada al momento de programar, además, ayuda a evitar escribir código repetitivo y mantiene todo estructurado gracias a su patrón MVC (Modelo Vista Controlador) ayudando así a realizar actualizaciones o cambios sin ningún problema [31].

Una de las ventajas más relevantes a destacar de Angular es que es modular y escalable, acoplándose así a las necesidades y al estar justificado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML y reutilizarlas en nuevos componentes.

1.2.2.8.6.3 Vue.js

Es un Framework progresivo JavaScript el cual se utiliza para crear interfaces de usuario de una forma rápida, agradable y práctica. En los últimos años Vue.js se ha popularizado para la creación de aplicaciones de una sola página (Single Page Application) ya que posee una modularización de bibliotecas al igual que Angular.

Vue fue creado por Evan You ex trabajador de Google quien fue uno de los desarrolladores de Angular. Este Framework fue lanzado en el año 2014, aunque al principio se pensó en que sería una biblioteca personal, con el apoyo de la comunidad y la gran acogida que tuvo se convirtió en uno de los Frameworks web más populares [32].

Algo a mencionar de Vue.js es que es un Framework reactivo, esto quiere decir que los sistemas desarrollados reaccionan al cambio, dependiendo de algún evento interno dentro del sistema se podría realizar cambios de estado y con ello cambiar el comportamiento de nuestros sistemas.

1.2.2.8.7 Typescript

Typescript es un superset de JavaScript ya que se pueden ejecutar programas creados con JavaScript en esta nueva tecnología. Este lenguaje de programación fue establecido en el año 2012 por Microsoft, desde entonces ha tenido una gran acogida por los desarrolladores y más aún cuando Google decidió tomarlo como lenguaje predeterminado para desarrollar con Angular. Actualmente Typescript se lo puede aplicar en cualquiera de los Frameworks o librerías más populares como es el caso de React para Frontend y Node para Backend [33].

1.2.2.8.8 Geolocalización

Es la cualidad que tienen los dispositivos digitales para obtener la ubicación geográfica en tiempo real con una conexión web. La geolocalización de estos dispositivos digitales no solo es posible en celulares inteligentes o tablets, también se aplica para laptops, computadoras de escritorio, televisores inteligentes, radios para autos, etc., siempre y cuando estén conectados al Internet. Hoy en día la geolocalización no solo puede ubicar coordenadas geográficas, sino que también es capaz de ubicar direcciones específicas en un mapa, trazar rutas de un punto a otro, entre otras cosas [34].

1.2.2.8.8.1 GPS en dispositivos móviles

Hoy por hoy los teléfonos inteligentes se pueden rastrear por GPS ya que cuentan con microsensors que permiten enviar y recibir señales satelitales las cuales son dirigidas a satélites que orbitan la tierra. Esto también se aplica para los radios inteligentes de autos ya que cuentan con estos microsensors los cuales no solo ayudan a localizar el

vehículo, sino que también permite ver la velocidad en la cual se está movilizandoo el vehículo [34].

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar una Aplicación Web Progresiva para automatizar el control de los vendedores y la gestión de pedidos de la empresa Ambagro S.A.

1.3.2 Objetivos específicos

- Analizar los procesos utilizados para la gestión de pedidos de la empresa Ambagro S.A.
- Proponer un proceso para el control de los vendedores de la empresa Ambagro S.A.
- Identificar las herramientas de software que se utilizarán para el desarrollo del sistema web.
- Implementar una Aplicación Web Progresiva para mejorar el control de los vendedores y la gestión de pedidos de la empresa Ambagro S.A.

CAPÍTULO II

METODOLOGÍA

2.1 Materiales

Para realizar el presente proyecto, se establecieron dos entrevistas las cuales fueron aplicadas al bodeguero y al gerente general de la empresa Ambagro S.A. con el propósito de entender y comprender los procesos que se ejecutan al momento de crear un pedido y despacharlo, y a su vez proponer un proceso de control de vendedores de la empresa.

De igual forma, se realizó una encuesta para los vendedores de la empresa con la finalidad de recaudar información destacada la cual permitirá conocer sus necesidades y así poder desarrollar el sistema web progresivo.

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMA COMPUTACIONALES E INFORMÁTICOS

Encuesta dirigida a los vendedores de la empresa AMBAGRO S.A.

OBJETIVO:

Determinar los requerimientos y la factibilidad para el desarrollo de una Aplicación Web Progresiva que automatice la gestión de pedidos generados por los vendedores de la empresa.

Instrucciones

- Lea detenidamente cada pregunta y responda con toda libertad y sinceridad.
- Marque con una "X" la respuesta que usted considere apropiada.

1. ¿Se siente conforme con el control actual de los procesos de ventas?
 - Si
 - No
2. ¿Alguna vez ha tenido inconvenientes al momento de crear un pedido y enviarlo a su responsable?
 - Si
 - No
3. ¿Conoce todos los precios de los productos que se encuentra a la venta?
 - Si
 - No
4. ¿Existe actualmente en la empresa Ambagro S.A. un sistema para gestión de pedidos?
 - Si
 - No
5. ¿Cree usted que la tecnología ayuda a mejorar los procesos de compra y venta?
 - Si
 - No
6. ¿Desearía tener acceso a la información de todos los productos que ofrece la empresa mediante una aplicación en tiempo real a través de un dispositivo móvil?
 - Si
 - No
7. ¿Le gustaría que luego de haber realizado un pedido, se notifique automáticamente al área de bodega para realizar el despacho correspondiente?
 - Si
 - No

8. ¿Considera que la empresa AMBAGRO S.A. debería implementar una aplicación web para facilitar los procesos de creación y despacho de sus pedidos?
- Si
 - No

Tabla 5. Modelo entrevista encargado de Bodega

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMA COMPUTACIONALES E INFORMÁTICOS

Entrevista dirigida al encargado de bodega de la empresa AMBAGRO S.A.

PROCESO	ACTOR	PREGUNTA	DESCRIPCIÓN	CONCLUSIÓN
Proceso de creación y despacho de pedidos	Bodeguero	1. ¿Cree usted que los vendedores de la empresa Ambagro S.A. llenan correctamente las notas de pedido?		
		2. ¿Los pedidos generados por los vendedores son enviados sin ningún inconveniente hacia usted?		
		3. ¿Cree que se ha aprovechado al máximo los beneficios que nos brindan las nuevas tecnologías en la empresa?		
		4. ¿Considera que incluir nuevas tecnologías		

		para las ventas de una empresa sería algo que impulsaría el desarrollo de esta?	
		5. ¿Considera que llevar un control digital de todos los pedidos generados por los vendedores de la empresa sería algo útil?	
		6. ¿Cómo considera que un sistema web le ayudaría a mejorar el rendimiento de las ventas realizadas en la empresa?	

Elaborado por: Diego Santamaria

Tabla 6. Modelo entrevista Gerente

UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMA COMPUTACIONALES E INFORMÁTICOS

Entrevista dirigida al gerente general de la empresa AMBAGRO S.A.

PROCESO	ACTOR	PREGUNTA	DESCRIPCIÓN	CONCLUSIÓN
Proceso de control de vendedores	Gerente	1. ¿Considera que los vendedores de la empresa Ambagro S.A. utilizan correctamente los recursos brindados por la empresa? ¿Por qué?		
		2. ¿Cree que los vendedores de la empresa tienen un buen rendimiento al momento de realizar sus actividades? ¿Por qué?		
		3. ¿Cree que es apropiado tener información adicional como hora, lugar, fecha y tipo de servicio que realiza		

		<p>cada vendedor de la empresa al momento de visitar a un cliente? ¿Por qué?</p>		
		<p>4. ¿Estaría de acuerdo en generar reportes de cada vendedor de la empresa, si es así, que tipo de reportes le gustaría generar?</p>		
		<p>5. ¿Considera que usted debería administrar las cuentas de usuario, perfiles y acceso a los recursos del sistema web de cada empleado que utilice el sistema? ¿Por qué?</p>		
		<p>6. ¿Considera que sería apropiado enviar un llamado de atención al vendedor si realiza alguna falta o error en su trabajo, si considera que sería apropiado, porque medio de comunicación le gustaría hacerlo (redes</p>		

		sociales, llamada personal, etc.)?)		
		7. ¿Estaría de acuerdo en llevar un registro de incidentes de cada vendedor de la empresa? ¿Por qué?		
		8. ¿Considera que sería adecuado tener un control con respecto a los kilometrajes de los vehículos de cada vendedor de la empresa? ¿Por qué?		
		9. ¿Cómo considera que un sistema web ayudaría a mejorar el rendimiento en el área de ventas de la empresa Ambagro S.A.?		

Elaborado por: Diego Santamaria

2.2 Métodos

2.2.1 Modalidad de investigación

Investigación bibliográfica

Porque se realizó varias revisiones de diferentes fuentes bibliográficas, documentos, folletos, manuales, etc., los cuales permitieron redactar el presente proyecto de investigación, así como también entender y saber cómo se utilizan las PWA y cuando son útiles.

Investigación de campo

Porque se realizó las observaciones adecuadas de las propiedades del problema en el espacio donde se desarrolló la investigación y se interactuó específicamente con los empleados involucrados en el proceso, todo esto para proponer una solución adecuada para la empresa y sus empleados.

2.2.2 Población y muestra

Población

La población para la presente investigación está constituida así:

Tabla 7. Población que interviene en el sistema web

Nº	Tipo de población	Descripción	Número
1	Personal de ventas	Vendedores	15
2	Bodega	Encargado de bodega	1
3	Gerente	Gerente general	1
Total			17

Elaborado por: Diego Santamaria

Muestra

En este caso, se trabajará con la población total considerando que es un grupo pequeño.

2.2.3 Recolección de información

Tras aplicar la entrevista al encargado de bodega y al gerente general de la empresa Ambagro S.A. se consiguieron los siguientes resultados.

Tabla 8. Entrevista encargado de Bodega

PROCESO	ACTOR	PREGUNTA	DESCRIPCIÓN	CONCLUSIÓN
Proceso de creación y despacho de pedidos	Bodeguero	1. ¿Cree usted que los vendedores de la empresa Ambagro S.A. llenan correctamente las notas de pedido?	Si, porque se llena en un formulario diseñado para ese fin.	Tras analizar las respuestas de la entrevista se pudo deducir que los procesos involucrados en la creación y despacho de pedidos generados por los vendedores de la empresa Ambagro S.A. tienen cierto déficit puesto que los pedidos creados no llegan a su responsable dando como resultado inconsistencias en el flujo de trabajo lo cual puede conllevar pérdidas de clientes. A si mismo, no se hace uso de las nuevas tecnologías lo cual dificulta estos procesos dentro de la empresa.
		2. ¿Los pedidos generados por los vendedores son enviados sin ningún inconveniente hacia usted?	No, los pedidos los trae el vendedor y si son importantes son enviados mediante una fotografía al WhatsApp.	
		3. ¿Cree que se ha aprovechado al máximo los beneficios que nos brindan las nuevas tecnologías en la empresa?	No, porque aún se utiliza papel.	
		4. ¿Considera que incluir nuevas tecnologías para las ventas de una empresa sería algo que	Si, porque facilitaría la transmisión de datos de toda índole.	

		impulsaría el desarrollo de esta?	
		5. ¿Considera que llevar un control digital de todos los pedidos generados por los vendedores de la empresa sería algo útil?	Si, porque tendríamos una base de datos que nos facilitaría la búsqueda de información.
		6. ¿Cómo considera que un sistema web le ayudaría a mejorar el rendimiento de las ventas realizadas en la empresa?	<p>Mas efectividad en las ventas.</p> <p>Mayor comunicación con los clientes.</p> <p>Facilidad de organización de pedidos para su despacho y procesos.</p>

Elaborado por: Diego Santamaria

Tabla 9. Entrevista Gerente

PROCESO	ACTOR	PREGUNTA	DESCRIPCIÓN	CONCLUSIÓN
Proceso de control de vendedores	Gerente	1. ¿Considera que los vendedores de la empresa Ambagro S.A. utilizan correctamente los recursos brindados por la empresa? ¿Por qué?	No. Falta conocimientos.	Al examinar las respuestas de la encuesta realizada se pudo deducir que los recursos que ofrece la empresa a sus vendedores no son utilizados adecuadamente ya que no se cuenta con un proceso para el control de vendedores lo que conlleva un bajo rendimiento en ellos y un desperdicio en los recursos brindados.
		2. ¿Cree que los vendedores de la empresa tienen un buen rendimiento al momento de realizar sus actividades? ¿Por qué?	No. Falta de manejo tecnológico.	
		3. ¿Cree que es apropiado tener información adicional como hora, lugar, fecha y tipo de servicio que realiza cada vendedor de la empresa al momento de visitar a un cliente? ¿Por qué?	Si. Mayor control y eficiencia.	

		4. ¿Estaría de acuerdo en generar reportes de cada vendedor de la empresa, si es así, que tipo de reportes le gustaría generar?	Si. Control de ventas y cobranzas. Tipo de atención. Uso correcto de vehículos.	
		5. ¿Considera que usted debería administrar las cuentas de usuario, perfiles y acceso a los recursos del sistema web de cada empleado que utilice el sistema? ¿Por qué?	Si. Para garantizar información completa y veraz.	
		6. ¿Considera que sería apropiado enviar un llamado de atención al vendedor si realiza alguna falta o error en su trabajo, si considera que sería apropiado, porque medio de comunicación le gustaría hacerlo (redes sociales, llamada personal, etc.)?	Si. Vía correo electrónico personal.	
		7. ¿Estaría de acuerdo en llevar un registro de incidentes de cada	Si. Es obligatorio por ley.	

		vendedor de la empresa? ¿Por qué?		
		8. ¿Considera que sería adecuado tener un control con respecto a los kilometrajes de los vehículos de cada vendedor de la empresa? ¿Por qué?	Si. Para optimizar el uso de recursos.	
		9. ¿Cómo considera que un sistema web ayudaría a mejorar el rendimiento en el área de ventas de la empresa Ambagro S.A.?	Crecimiento en ventas. Optimización de recursos. Mayor control en cartera y cobranza.	

Elaborado por: Diego Santamaria

Tras aplicar la encuesta a los 15 vendedores de la empresa Ambagro S.A. se obtuvieron resultados adecuados para poder avanzar con la presente investigación. A continuación, se muestran dichos resultados con su respectivo análisis e interpretación.

Pregunta N°1: ¿Se siente conforme con el control actual de los procesos de ventas?

Figura 3. Conformidad con el control actual de procesos de venta



Elaborado por: Diego Santamaria

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 3, se puede apreciar que el 93% de los vendedores de la empresa Ambagro S.A no están conformes con el control actual de los procesos de ventas de la empresa, mientras que un 7% si lo está, por lo que se puede determinar que los procesos de ventas internos deben ser más estructurados y controlados para evitar inconvenientes.

Pregunta N°2: ¿Alguna vez ha tenido inconvenientes al momento de crear un pedido y enviarlo a su responsable?

Figura 4. Inconvenientes al crear un pedido y enviarlo a su responsable



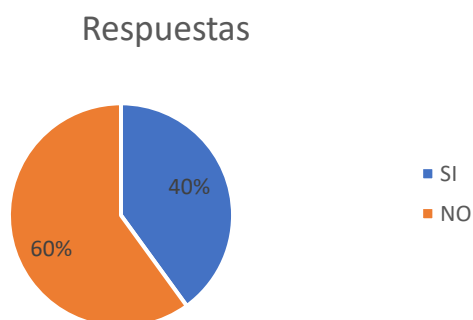
*Elaborado por: Diego Santamaria
Fuente: Encuesta aplicada*

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 4, se puede apreciar que el 100% de los vendedores de la empresa Ambagro S.A. han tenido inconvenientes al momento de crear o enviar un pedido al encargado de bodega, lo que demuestra que existen inconsistencias en el flujo de trabajo ya que no se innovan los procesos actuales.

Pregunta N°3: ¿Conoce todos los precios de los productos que se encuentra a la venta?

Figura 5. Conocimiento de los precios de los productos en venta



*Elaborado por: Diego Santamaria
Fuente: Encuesta aplicada*

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 5, se puede apreciar que el 60% de los vendedores de la empresa Ambagro S.A. no conocen todos los precios de los productos que se encuentran a la venta, mientras que el 40% si conocen dichos precios, lo que demuestra que sería útil tener un registro de los productos con sus respectivos precios en un sistema móvil para evitar inconvenientes.

Pregunta N°4: ¿Existe actualmente en la empresa Ambagro S.A. un sistema para gestión de pedidos?

Figura 6. Existe un sistema para gestión de pedidos



Elaborado por: Diego Santamaria

Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 6, se puede apreciar que el 100% de los vendedores de la empresa Ambagro S.A. indicaron que no existe un sistema para gestionar pedidos, lo que demuestra que aún no se utilizan tecnologías las cuales brindan múltiples beneficios para la empresa y sus integrantes.

Pregunta N°5: ¿Cree usted que la tecnología ayuda a mejorar los procesos de compra y venta?

Figura 7. La tecnología ayudaría a mejorar los procesos de compra y venta



Elaborado por: Diego Santamaria
Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 7, se puede apreciar que el 100% de los vendedores de la empresa Ambagro S.A. están de acuerdo en que la tecnología ayuda a mejorar los procesos de compra y venta, por lo que se demuestra que sería de mucha ayuda implementar tecnologías al flujo de trabajo de la gestión de pedidos.

Pregunta N°6: ¿Desearía tener acceso a la información de todos los productos que ofrece la empresa mediante una aplicación en tiempo real a través de un dispositivo móvil?

Figura 8. Acceso a la información de los productos por medio de un dispositivo móvil



Elaborado por: Diego Santamaria
Fuente: Encuesta aplicada

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 8, se puede apreciar que el 100% de los vendedores de la empresa Ambagro S.A. quisieran tener acceso a la información de todos los productos que ofrece la empresa, lo que demuestra que sería conveniente tener almacenada la información de los productos ofertados por la empresa en un sistema móvil.

Pregunta N°7: ¿Le gustaría que luego de haber realizado un pedido, se notifique automáticamente al área de bodega para realizar el despacho correspondiente?

Figura 9. Notificar al área de bodega cuando se realiza un pedido



*Elaborado por: Diego Santamaria
Fuente: Encuesta aplicada*

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 9, se puede apreciar que el 100% de los vendedores de la empresa Ambagro S.A. están de acuerdo en que se notifique al encargado de bodega cuando exista un nuevo pedido, dando así una mayor dinamismo a este proceso.

Pregunta N°8: ¿Considera que la empresa AMBAGRO S.A. debería implementar una aplicación web para facilitar los procesos de creación y despacho de sus pedidos?

Figura 10. Implementar una aplicación web en la empresa Ambagro S.A.



*Elaborado por: Diego Santamaria
Fuente: Encuesta aplicada*

Análisis e interpretación de resultados:

Acorde con los resultados representados en la Figura 10, se puede apreciar que el 100% de los vendedores de la empresa Ambagro S.A. están de acuerdo en implementar una aplicación web para facilitar los procesos de creación y despacho de pedidos, por lo que se demuestra que las nuevas tecnologías traen varias ventajas que los usuarios pueden identificar a simple vista, es por ello que las hacen muy populares.

2.2.4 Procesamiento y análisis de datos

De acuerdo con la entrevista realizada al encargado de bodega de la empresa Ambagro S.A. se pudo identificar que:

- Es indispensable llevar una organización con los pedidos realizados por los vendedores para evitar confusiones y retrasos al momento de despacharlos.
- Es importante tener un registro de todos los pedidos despachados por el encargado de bodega ya que se tendrá una mayor organización y control al momento de manejar esta información.
- Es importante tener actualizada la información de los productos que se estén ofertando en la empresa para ofrecer un servicio de calidad.
- Tener un formulario estructurado al momento de crear un pedido es de suma importancia ya que con ello se pueden realizar los despachos correctamente.

De acuerdo con la entrevista realizada al gerente general de la empresa Ambagro S.A. se pudo identificar que:

- Es necesario contar con un seguimiento de los vendedores ya que no se utilizan los recursos brindados por la empresa adecuadamente.
- Es importante tener información adicional cuando el vendedor realice un pedido ya que con ello se puede verificar el rendimiento de este al momento de realizar una venta.
- Es importante tener un registro de las actividades que realiza un vendedor ya que con ello se podrá verificar si los recursos de la empresa son utilizados correctamente.
- Es indispensable mantenerse en constante comunicación con un vendedor de la empresa vía correo electrónico para informarle sobre cualquier novedad.

- Es necesario llevar un registro de incidentes de los vendedores de la empresa ya que es obligatorio por ley.

De acuerdo con la encuesta realizada a los vendedores de la empresa Ambagro S.A. se pudo identificar que:

- Los vendedores no están conformes con el control actual de los procesos de venta ya que existen varios problemas que dificultan su desempeño al momento de realizar una venta.
- A los vendedores de la empresa les gustaría acceder a la información de los productos ofertados por la empresa fácilmente, sin la necesidad de llevar un cuaderno o revista que cuente con esta información.
- Es indispensable acortar los tiempos de creación y despacho de los pedidos para garantizar una atención apropiada para los clientes.
- Es importante que los procesos que intervienen en la creación y despacho de pedidos sean rápidos y directos para evitar pérdidas a largo plazo en la empresa.
- Es indispensable innovar la gestión de pedidos de la empresa para asegurar la satisfacción de los clientes.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1 Análisis y discusión de resultados

3.1.1 Determinación del framework de desarrollo web

Comparación de Frameworks

Tabla 10. Comparación de Framework.

	React	Angular	Vue.js
Creadores	Facebook	Google	Evan You
Documentación	Si	Si	Si
Aprendizaje	Medio	Medio	Fácil
Modelo	Virtual DOM	MVC	Virtual Dom
Enlace de datos	Unidireccional	Bidireccional	Bidireccional
Aplicaciones Web	Necesita estar incorporado con varios módulos y herramientas.	Usar independientemente.	Requiere herramientas de terceros.
Reutilización de código	No	Si	Si
Compañías que lo usan	Atlassian, KhanAcademy, Facebook, Yahoo, Netflix.	TheGuardian, Netflix, Upwork, Paypal.	Adobe, Grammarly, Facebook, Alibaba.
Descargas NPM semanales	15366533	20278177	3164568

Elaborado por: Diego Santamaria

Framework elegido

Al realizar el cuadro comparativo de los 3 Frameworks más populares para desarrollar Aplicaciones Web Progresiva se llegó a la conclusión de que el Framework Angular es el más adecuado para realizar el presente proyecto porque utiliza el lenguaje TypeScript el cual es más explícito y fiable que su predecesor JavaScript. Angular cuenta con elementos de diseño pre planteados que facilitan el desarrollo de sistemas web, también trabaja con componentes los cuales se pueden reutilizar en varias vistas. Una particularidad de este Framework es que es altamente compatible con las plataformas iOS y Android al momento de crear aplicaciones web progresivas.

A pesar de que su curva de aprendizaje es elevada por su gran estructura, características y herramientas, es una excelente opción para obtener un producto final de calidad en un tiempo determinado.

3.1.2 Determinación de la metodología de desarrollo del proyecto

Comparación de metodologías ágiles

Tabla 11. Comparación de metodologías ágiles para el desarrollo de una PWA

	KANBAN	SCRUM	XP
Tamaño de equipos de trabajo	Indefinido	5 a 9 personas	2 a 10 personas
Tamaño de proyecto	Todos	Todos	Pequeños
Documentación	No definida	Si	Si
Reuniones diarias	Si	Si	Si
Principios	<ul style="list-style-type: none">• Visualización.• Priorización.• Mejora continua.	<ul style="list-style-type: none">• Auto organización.• Colaboración.	<ul style="list-style-type: none">• Retroalimentación rápida.• Simplicidad.• Cambios incrementales.• Aceptar cambios.

	<ul style="list-style-type: none"> • Liderazgo en todos los niveles. • Calidad garantizada. 	<ul style="list-style-type: none"> • Priorización basada en el valor. • Desarrollo iterativo. 	<ul style="list-style-type: none"> • Trabajo de calidad.
Ventajas	<ul style="list-style-type: none"> • Equipos de trabajo organizados • Cumplimiento • Asignación de tareas • Calidad del producto final • Flexibilidad 	<ul style="list-style-type: none"> • Gestión de las expectativas del usuario. • Resultados anticipados • Flexibilidad y adaptación a los diferentes entornos. 	<ul style="list-style-type: none"> • Relación estrecha con el cliente. • Ausencia de trabajos de programación innecesarios • Software estable debido a continuas pruebas • Menos errores gracias a la programación en pareja • Aplicación rápida de cambios
Desventajas	<ul style="list-style-type: none"> • Costoso • Limitación de tareas • No es adecuado para todos los proyectos 	<ul style="list-style-type: none"> • Sus equipos de trabajo son muy reducidos. • Exhaustiva definición de las tareas y sus plazos. • Alta cualificación o formación para los que la utilicen. 	<ul style="list-style-type: none"> • Mayor esfuerzo de trabajo • El cliente se implica en el proceso • Requiere mucho tiempo • Cara • Requiere control de versiones • Requiere autodisciplina en la aplicación

Complejidad	Alta	Alta	Baja
-------------	------	------	------

Elaborado por: Diego Santamaria

Metodología elegida

Para elegir una metodología de desarrollo ágil la cual se adapte mejor al desarrollo del sistema web se tomará como referencia la tabla anterior ya que se realizó las comparaciones de las metodologías ágiles más populares.

Para poder seleccionar una metodología adecuada se utilizará una escala numérica con números del 1 al 5 los cuales tendrán las siguientes valoraciones [35]:

1: muy malo

4: bueno

2: malo

5: excelente

3: regular

Tabla 12. Metodología de desarrollo ágil elegida

	KANBAN	SCRUM	XP
Retroalimentación	3	4	5
Colaboración	3	4	3
Simplicidad	3	2	4
Costo	1	2	3
Mejor técnica	3	3	1
Adaptable al proyecto propuesto	1	2	4
Total	14	17	20
Porcentaje	70%	85%	100%

Elaborado por: Diego Santamaria

Una vez evaluados y calificados cada uno de los ítems propuestos para cada metodología de desarrollo ágil se pudo determinar que la metodología XP (Xtreme Programming) con un porcentaje del 100% es la más adecuada para desarrollar el presente sistema web ya que muchas de sus características y ventajas coinciden con este trabajo.

3.2 Desarrollo de la propuesta

3.2.1 Fase I: Planificación

En esta fase se define el levantamiento de la información de la empresa, con ello se pueden generar las historias de usuario, determinación de los roles de usuario, tareas, valoración y estimación de las historias de usuario, plan de entrega y el plan de iteraciones.

3.2.1.1 Levantamiento de información

Para el levantamiento de información se utilizaron las encuestas dirigidas al encargado de bodega y gerente general, además de las entrevistas aplicadas a los vendedores de la empresa Ambagro S.A. junto con la investigación de campo. Estos métodos ayudaron a definir las inconsistencias que existen dentro de los procesos de creación y despacho de pedidos, así como también identificar la necesidad de implementar nuevas tecnologías las cuales ayudaran a mejorar el rendimiento de sus procesos dando como resultado una mejor atención al cliente.

En las entrevistas aplicadas al encargado de bodega y gerente general de la empresa Ambagro S.A., se formularon preguntas abiertas con la finalidad de poder responderlas en base a su criterio y experiencia, exponiendo sus necesidades con respecto a la gestión de pedidos y al seguimiento de los vendedores respectivamente, igualmente con la innovación de su empresa, aplicando nuevas tecnologías para adaptarse a las necesidades actuales.

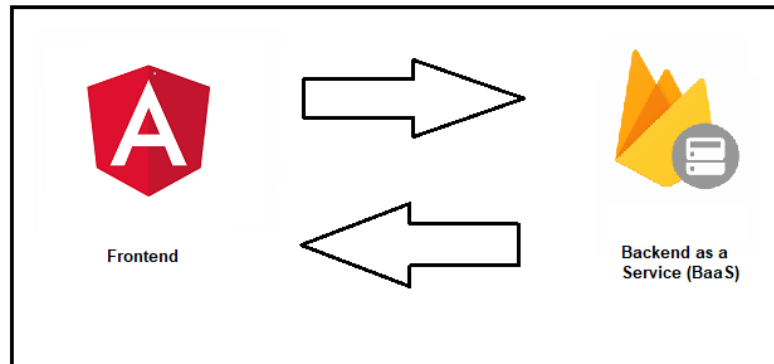
En la encuesta aplicada a los vendedores de la empresa Ambagro S.A., se formularon preguntas cerradas enfocadas a encontrar problemas e inconsistencias que se encuentren en el proceso de generación de pedidos y dar algunas ideas las cuales ayudarían a mejorar dichos problemas.

3.2.1.2 Arquitectura de la aplicación

En la figura 11 se muestra la arquitectura de la aplicación web progresiva. El Backend usado es Firebase el cual es un BaaS (Backend as a Service). BaaS es un servicio que proporciona a los desarrolladores un modelo el cual permite utilizar una serie de herramientas sin la necesidad de tener que desarrollar un Backend.

Angular es el Frontend de la aplicación, éste se encargará de conectarse con Firebase para utilizar sus funcionalidades y poder realizar acciones como añadir, editar, recuperar y eliminar información.

Figura 11. Arquitectura de la aplicación



Elaborado por: Diego Santamaria

La aplicación web progresiva funcionará de forma online, ya que se utilizará la API de geolocalización de los navegadores web la cual entrega la ubicación exacta del dispositivo en el cual se instale el sistema web. Esta API trabaja simultáneamente con el GPS del dispositivo ya que busca utilizar la mejor funcionalidad disponible para obtener esta información. Para acceder a la posición del dispositivo móvil se empleará el método “Geolocation.getCurrentPosition()” propio de JavaScript, el cual funciona con una conexión a internet.

3.2.1.3 Definición de roles

A continuación, se detallan los roles de usuario que tiene la aplicación web progresiva.

Tabla 13. Definición de roles

Rol de usuario	Descripción
Administrador	<ul style="list-style-type: none">• Establece el rol de los nuevos usuarios, así como también habilita la vista para el registro de estos.• Habilita e inhabilita cuentas de usuarios del sistema.• Genera reportes en base a todos los usuario, usuarios habilitados y usuarios inhabilitados.
Bodeguero	<ul style="list-style-type: none">• Gestiona las categorías, nombres productos, empaques y productos que se manejen en la empresa.• Observa y despacha los pedidos creados por los vendedores.
Vendedor	<ul style="list-style-type: none">• Ingresa kilometraje de su vehículo• Crea nuevos clientes• Crea, actualiza o elimina pedidos• Crea visitas• Crea cobranzas
Gerente	<ul style="list-style-type: none">• Genera reportes de los kilometrajes, pedidos, visitas y cobranzas realizados por los vendedores.

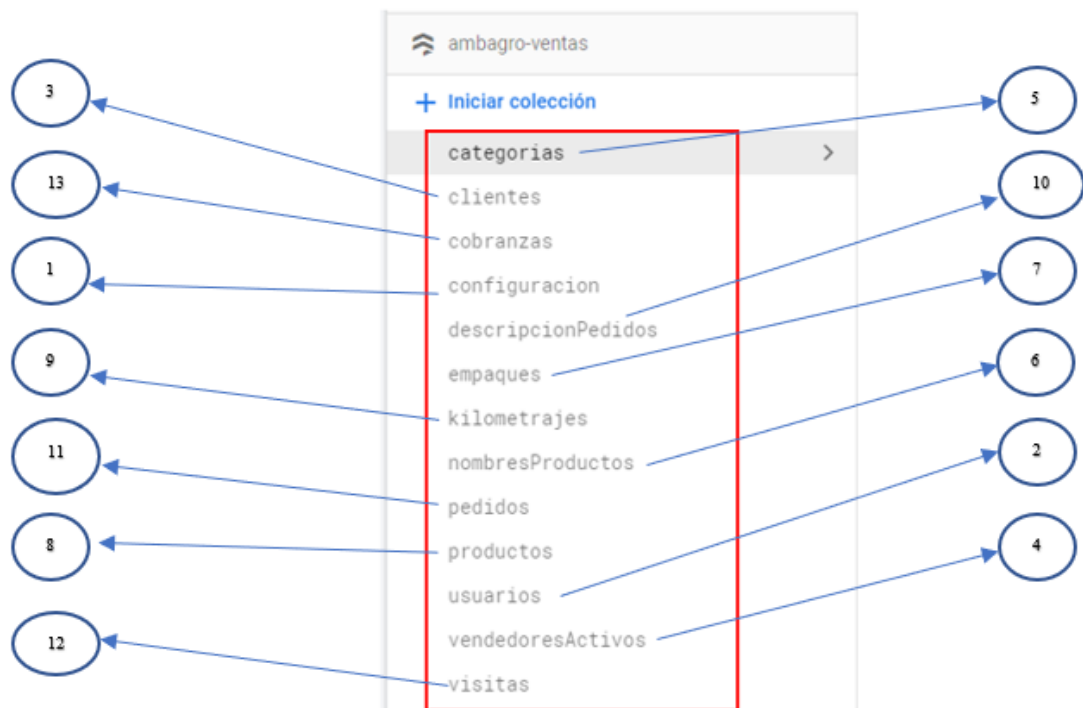
Elaborado por: Diego Santamaria

3.2.1.4 Diseño de la base de datos

Para el desarrollo del presente proyecto se utilizará la base de datos NoSQL Firestore de Firebase por lo que, para el modelado de esta, primero se diseñaron las interfaces ya que, para generar una base de datos NoSQL se deben realizar las vistas del sistema para poder crear las colecciones las cuales almacenarán los datos que se utilizarán en la aplicación web.

A continuación se detalla las colecciones con sus respectivos campos los cuales alojaran la información del sistema web.

Figura 12. Colecciones de Base de datos NoSQL



Elaborado por: Diego Santamaria

1. Colección de configuración

Esta colección aloja las configuraciones esenciales del sistema web, cada una de estas configuraciones realizan una función en específico para que el sistema trabaje correctamente.

El modelo "ConfiguracionPerfil" maneja el perfil del usuario del nuevo registro de usuario del sistema web.

Figura 13. Modelo configuración perfil

```
export interface ConfiguracionPerfil{  
  perfilUsuario?: string;  
}
```

Elaborado por: Diego Santamaria

El modelo “ConfiguracionPermitirRegistro” maneja el estado de registro de usuarios del sistema.

Figura 14. Modelo Configuración permitir registro

```
export interface ConfiguracionPermitirRegistro{  
  permitirRegistro?: boolean;  
}
```

Elaborado por: Diego Santamaria

El modelo “ConfiguracionTamanoClientes” maneja el tamaño de la colección denominada clientes y el token que cada usuario vendedor tiene alojado en su local storage.

Figura 15. Modelo configuración tamaño cliente

```
export interface ConfiguracionTamanoClientes{  
  tamanoClientes?: number;  
  tokenUpdateCliente?: string;  
}
```

Elaborado por: Diego Santamaria

El modelo “ConfiguracionNumeroPedido” maneja el tamaño de la colección denominada pedidos.

Figura 16. Modelo configuración número pedido

```
export interface ConfiguracionNumeroPedido{  
  numeroPedido?: number;  
}
```

Elaborado por: Diego Santamaria

El modelo “ConfiguracionNumeroVisita” maneja el tamaño de la colección denominada visitas.

Figura 17. Modelo configuración número visita

```
export interface ConfiguracionNumeroVisita{  
  numeroVisita?: number;  
}
```

Elaborado por: Diego Santamaria

El modelo “configuracionNumeroCobranza” maneja el tamaño de la colección denominada cobranzas.

Figura 18. Modelo configuración número cobranza

```
export interface ConfiguracionNumeroCobranza{  
  numeroCobranza?: number;  
}
```

Elaborado por: Diego Santamaria

El modelo “configuracionNumeroKilometraje” maneja el tamaño de la colección denominada kilometrajes.

Figura 19. Modelo configuración número kilometraje

```
export interface ConfiguracionNumeroKilometraje{  
  numeroKilometraje?: number;  
}
```

Elaborado por: Diego Santamaria

2. Colección de usuarios

Esta colección contiene a todos los usuarios del sistema web que han sido registrados exitosamente.

Figura 20. Modelo usuario

```
export interface Usuario{
  nombreYapellido?: string;
  email?:string;
  telefono?:number;
  foto?: string;
  perfil?: string;
  activo?: boolean;
  uid?: string;
  tokenEmail?: string;
}
```

Elaborado por: Diego Santamaria

3. Colección de clientes

La colección “clientes” contiene todos los clientes ingresados en el sistema para utilizarlos en diferentes módulos.

Figura 21. Modelo cliente

```
export interface Cliente{
  id?: string;
  ruc?: string
  nombre?: string;
  nombreComercial?: string;
  telefono?: string;
  direccion?: string;
}
```

Elaborado por: Diego Santamaria

4. Colección de vendedores activos

Esta colección aloja los vendedores que se encuentran activos en el sistema, depende de las colecciones vendedores y kilometrajes

Figura 22. Modelo vendedores activos

```
export interface vendedoresActivos{
  id?: string;
  vendedorID?: string;
  fechaGenerada?: string;
  horaGenerada?: string;
  kilometrajeID?: string;
  vendedor?: string;
  telefono?: string;
  email?: string;
  foto?: string;
  kilometrajeActual?: string;
}
```

Elaborado por: Diego Santamaria

5. Colección de categorías

La colección “categorías” contiene todas las categorías ingresadas en el sistema.

Figura 23. Modelo categoría

```
export interface Categoria{
  id?: string;
  nombre?: string;
}
```

Elaborado por: Diego Santamaria

6. Colección de nombres productos

La colección “nombreProducto” aloja los nombres de los productos en dependencia a la colección categoría.

Figura 24. Modelo nombre producto

```
export interface nombreProducto{
  id?: string;
  categoria?: string;
  nombre?: string;
}
```

Elaborado por: Diego Santamaria

7. Colección de empaques

La colección empaques guarda los empaques necesarios para crear productos.

Figura 25. Modelo empaque

```
export interface Empaque{
  id?: string;
  nombre?: string;
}
```

Elaborado por: Diego Santamaria

8. Colección de productos

La colección “productos” almacena los productos creados en dependencia a las colecciones de “categorias”, “nombresProductos” y “empaques”

Figura 26. Modelo producto

```
export interface Producto{
  id?: string;
  categoria?: string;
  nombre?: string;
  empaque?: string;
  precio?: number;
  estado?: boolean;
}
```

Elaborado por: Diego Santamaria

9. Colección de kilometrajes

Esta colección aloja el kilometraje ingresado por los vendedores en el sistema.

Figura 27. Modelo kilometraje

```
export interface Kilometraje{
  id?: string;
  numeroKilometraje?: number;
  idUser?: string;
  fecha?: string;
  horaInicio?: string;
  horaFin?: string;
  kilometrajeInicial?: number;
  kilometrajeFinal?: number;
  latInicial?: number;
  lonInicial?: number;
  latFinal?: number;
  lonFinal?: number;
  detalle?: string;
}
```

Elaborado por: Diego Santamaria

10. Colección de descripción pedido

La colección “DescripcionPedido” guarda la descripción de cada pedido generado por un vendedor.

Figura 28. Modelo descripción pedido

```
export interface DescripcionPedido{
  id?: string;
  categoria?: string;
  producto?: string;
  empaque?: string;
  precio?: number;
  cantidad?: number;
  descuento?: number;
  subtotal?: number;
}
```

Elaborado por: Diego Santamaria

11. Colección de pedidos

La colección “pedidos” almacena los pedidos generados por los vendedores en dependencia de las colecciones “clientes”, “usuarios” y “descripcionPedidos”.

Figura 29. Modelo pedido

```
export interface Pedido{
  id?: string;
  numeroPedido?: string;
  rucCliente?: string;
  nombreCliente?: string;
  nombreComercial?: string;
  direccion?: string;
  fecha?: string;
  hora?: string;
  diasCredito?: number;
  despachado?: boolean;
  idVendedor?: string;
  vendedor?: string;
  idDescripcionPedido?: string;
  total?: number;
  telefono?: string;
  lat?: string;
  lon?: string;
}
```

Elaborado por: Diego Santamaria

12. Colección de visitas

Esta colección aloja las visitas creadas por los vendedores del sistema.

Figura 30. Modelo visita

```
export interface Visita{
  id?: string;
  numeroVisita?: string;
  rucCliente?: string;
  nombreCliente?: string;
  nombreComercial?: string;
  telefono?: string;
  direccion?: string;
  fechaCreacion?: string;
  horaCreacion?: string;
  descripcion?: string;
  lat?: string;
  lon?: string;
  idVendedor?: string;
  vendedor?: string;
}
```

Elaborado por: Diego Santamaria

13. Colección de cobranzas

La colección “cobranzas” guarda las cobranzas generadas por los vendedores del sistema web.

Figura 31. Modelo cobranza

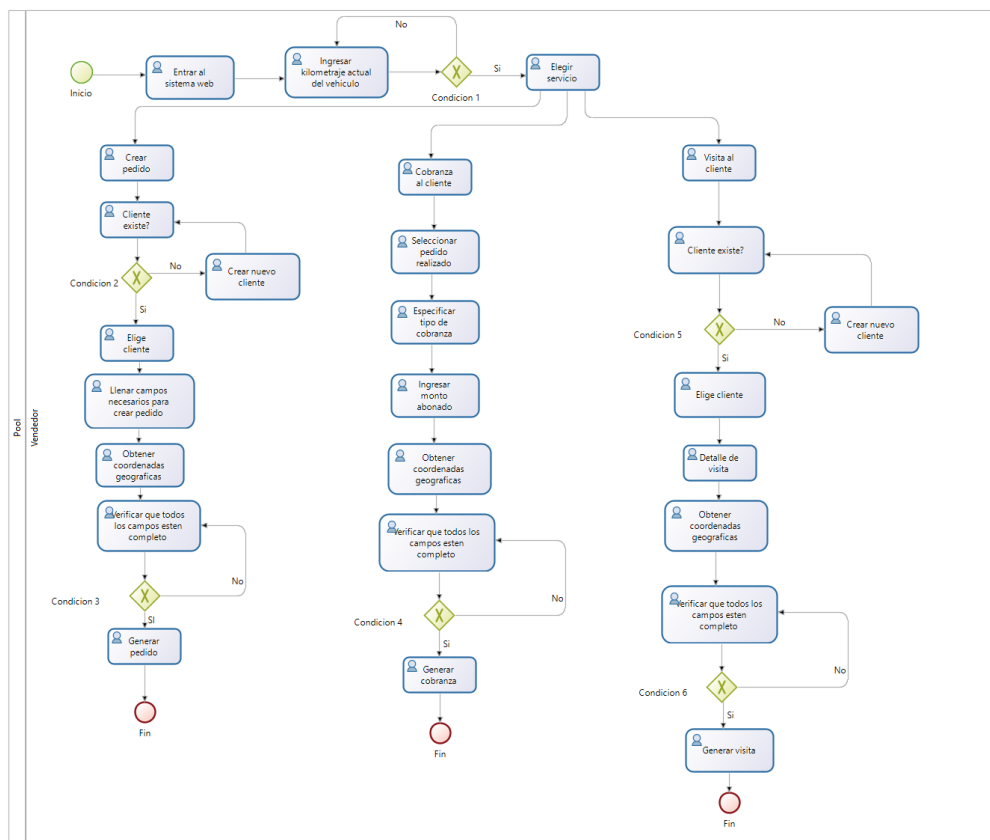
```
export interface Cobranza{
  id?: string;
  numeroCobranza?: string;
  rucCliente?: string;
  nombreCliente?: string;
  nombreComercial?: string;
  telefono?: string;
  direccion?: string;
  fechaCreacion?: string;
  horaCreacion?: string;
  tipoCobranza?: string;
  concepto?: string;
  monto?: number;
  numeroFactura?: string;
  numeroComprobante?: string;
  lat?: string;
  lon?: string;
  idVendedor?: string;
  vendedor?: string;
}
```

Elaborado por: Diego Santamaria

3.2.1.5 Diagrama de proceso para el control de los vendedores de la empresa Ambagro S.A.

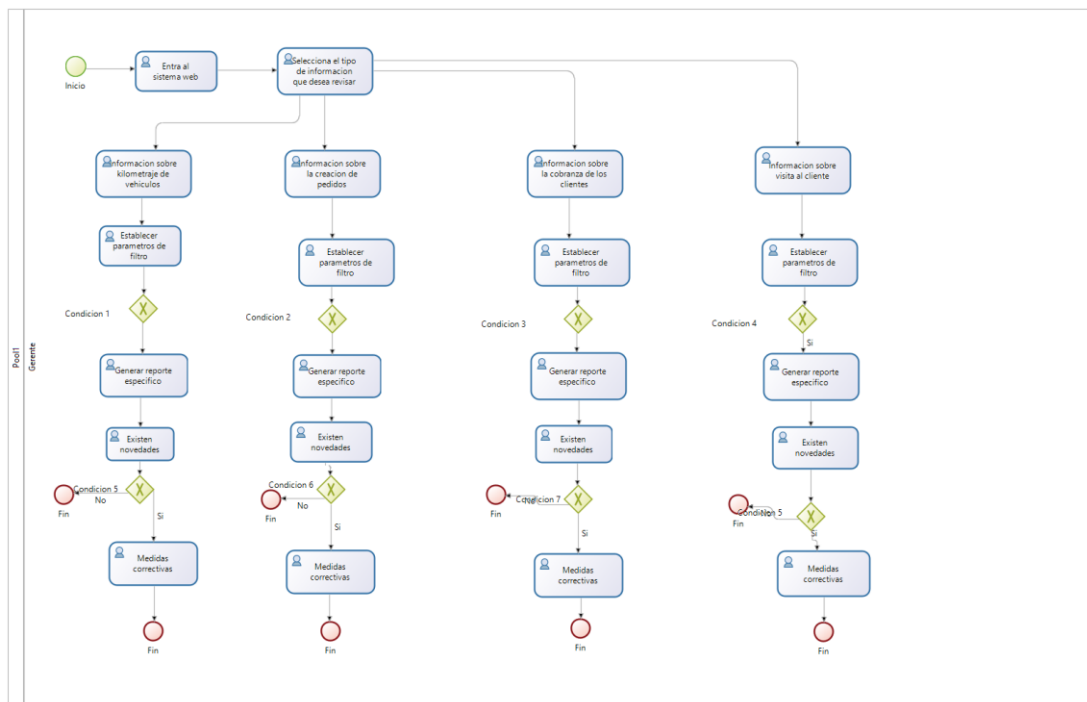
En base a los requerimientos, encuestas e investigación de campo realizada en la empresa Ambagro S.A. se pudo llevar a cabo el siguiente diagrama de procesos el cual ayudará a mejorar el control de vendedores de la empresa mediante la aplicación web progresiva.

Figura 32. Diagrama de procesos para el registro de actividades del usuario vendedor



Elaborado por: Diego Santamaria

Figura 33. Diagrama de procesos para el control y reportería de actividades del usuario vendedor



Elaborado por: Diego Santamaria

El proceso propuesto para el control de los vendedores de la empresa Ambagro S.A. consiste en que los usuarios con el perfil vendedor ingresen al sistema web progresivo con su correo electrónico y contraseña. Una vez dentro del sistema, deberán ingresar el kilometraje actual de su vehículo para poder utilizar los diferentes módulos del sistema; caso contrario no podrán utilizar dichos módulos.

Una vez ingresado el kilometraje de su vehículo, podrán seleccionar qué actividad van a realizar en el local que se encuentren, estas actividades pueden ser: crear pedido, visita al cliente y cobranza. Cada una de las actividades mencionadas anteriormente cumplen con una tarea específica. Lo que cada una de las tareas tienen en común es que se tomaran las coordenadas geográficas desde el punto exacto en donde se está generando dichas actividades para posteriormente verificar y generar reportes, todo esto se lo puede apreciar en la Figura 32.

El usuario gerente puede revisar las diferentes actividades que realizaron cada uno de los vendedores de la empresa detalladamente gracias a las coordenadas geográficas y kilometraje ingresados por cada usuario vendedor. Esto se lo puede realizar

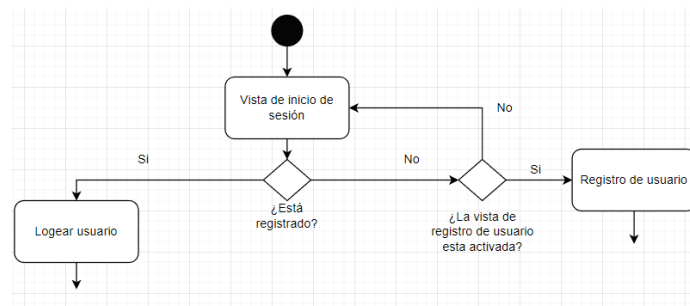
seleccionando que actividad desea revisar, si quiere información de un usuario o de alguna fecha en específico deberá establecer dichos parámetros para obtener la información esperada. Si el usuario gerente identifica sucesos extraños en algún reporte generado, éste se contactará directamente con el vendedor para realizar un llamado de atención, todo esto se lo puede visualizar en la Figura 33.

3.2.1.6 Diagrama de actividades de la aplicación

Una vez planteado el proceso para el control de los vendedores, se procede a detallar las actividades que realizara el sistema web progresivo.

Al iniciar el sistema web, se desplegará la vista de inicio de sesión en la que el usuario deberá ingresar el correo electrónico y contraseña con la que se haya registrado. Si el usuario aún no se ha registrado en el sistema, podrá hacerlo siempre y cuando la vista de registro de usuario este habilitada, caso contrario, no lo podrá hacer.

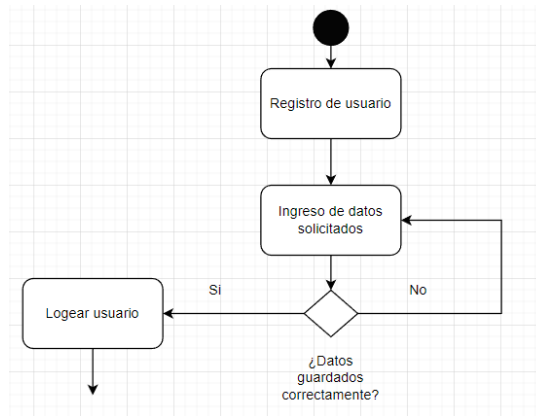
Figura 34. Inicio de sesión del sistema



Elaborado por: Diego Santamaria

Si la vista de registro de usuario está habilitada, el usuario deberá ingresar los datos solicitados para poder ser registrado en el sistema. Si los datos se guardan correctamente el usuario será redirigido al sistema como tal, caso contrario, se le desplegará un mensaje el cual mostrará el posible error.

Figura 35. Registro de usuarios del sistema

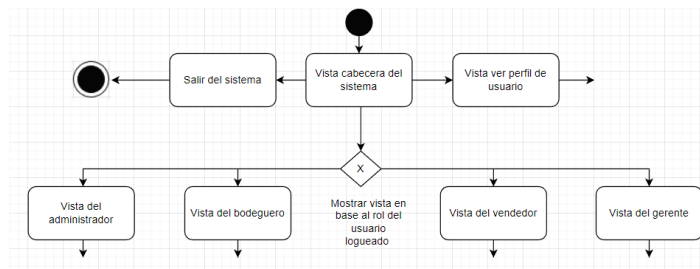


Elaborado por: Diego Santamaria

Una vez dentro del sistema, se desplegará la vista cabecera en la parte superior. Esta vista contiene las opciones para salir del sistema y ver el perfil del usuario logueado.

Para mostrar la vista que contiene los diferentes módulos de cada usuario, se verifica el rol del usuario conectado y se activará su vista.

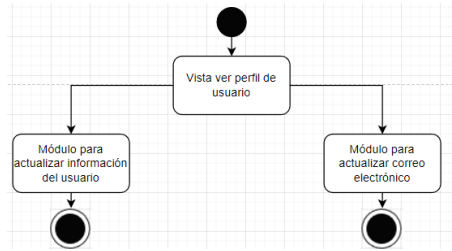
Figura 36. Vistas de los diferentes usuarios del sistema



Elaborado por: Diego Santamaria

La vista del perfil del usuario contiene la información proporcionada al momento de crear su cuenta. Aquí, el usuario podrá actualizar sus datos y también su correo electrónico.

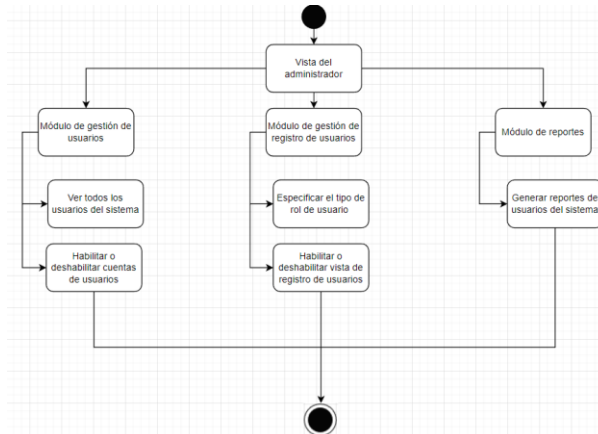
Figura 37. Vista y módulos del perfil de usuario



Elaborado por: Diego Santamaria

En la vista del administrador se podrá acceder a los módulos de gestión de usuarios, registro de usuarios y reportes. Cada módulo permitirá realizar diferentes actividades las cuales ayudan a gestionar y ordenar la información de los usuarios que se registran en el sistema.

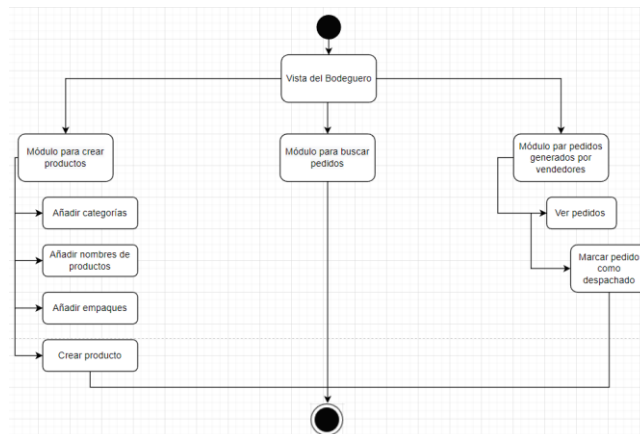
Figura 38. Vista y módulos del usuario administrador del sistema



Elaborado por: Diego Santamaria

La vista del bodeguero abarca los módulos de crear productos, ver los pedidos generados por los vendedores y buscar pedidos. El primer módulo contiene las categorías, nombres de los productos y empaques los cuales ayudarán a crear un producto. El siguiente módulo lista todos los pedidos que se encuentren pendientes por despachar. Una vez despachado el pedido, el usuario bodeguero deberá marcar el pedido como despachado para que desaparezca de la lista. El ultimo módulo lista los pedidos despachados por el usuario bodeguero.

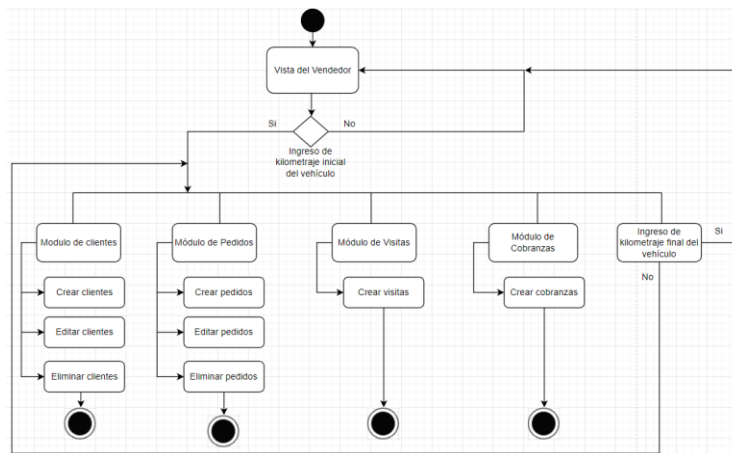
Figura 39. Vista y módulos del usuario bodeguero del sistema



Elaborado por: Diego Santamaria

La vista del vendedor contiene los módulos de clientes, pedidos, visitas, cobranzas y kilometraje. Antes de poder utilizar dichos módulos, el usuario vendedor deberá ingresar el kilometraje actual de su vehículo, caso contrario no podrá utilizarlos. Al finalizar su jornada laboral tendrá que dirigirse al módulo de kilometraje final para registrar el kilometraje actual de su vehículo y poder cerrar el sistema.

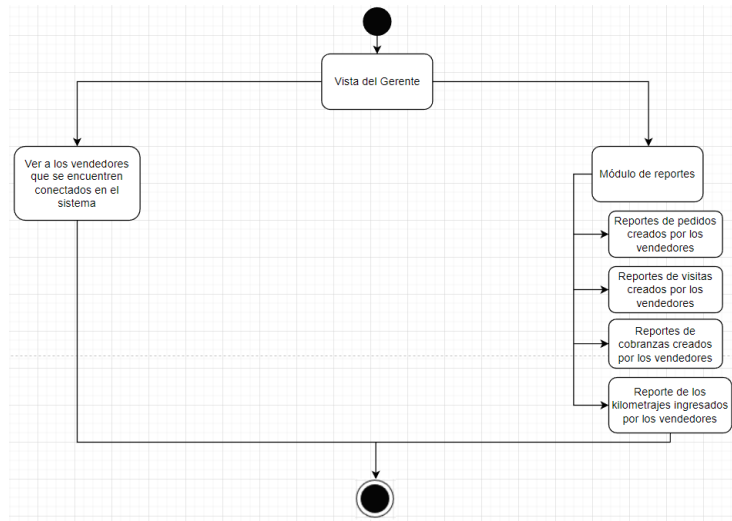
Figura 40. Vista y módulos del usuario vendedor del sistema



Elaborado por: Diego Santamaria

En el módulo del gerente se podrá observar los vendedores activos en el sistema. También contará con un módulo de reportería en el cual, especificando ciertos parámetros, se podrá generar reportes de los pedidos, visitas, cobranzas y kilometrajes de los vendedores.

Figura 41. Vista y módulos del usuario gerente del sistema



Elaborado por: Diego Santamaria

3.2.1.7 Historias de Usuario

Las historias de usuario ayudan a gestionar de una manera ágil los requerimientos de los usuarios, evitando estructurar y diseñar grandes cantidades de documentos. Para realizar las historias de usuario se estableció un modelo que se encuentra en la Tabla 14.

Tabla 14. Plantilla de Historia de Usuario

Historia de Usuario			
Número:		Usuario:	
Nombre de la historia:			
Prioridad en el negocio:		Riesgo en el desarrollo:	
Puntos estimados:		Iteración asignada:	
Programador responsable:			
Descripción:			
Observación:			

Elaborado por: Diego Santamaria

Elementos que conforman la plantilla de historia de usuario.

- **Número:** Identificador único de cada historia de usuario
- **Usuario:** persona o grupo de personas a la que va dirigida la historia de usuario.

- **Nombre de la historia:** Nombre asignado a la historia de usuario.
- **Prioridad en el negocio:** Valoración de la historia de usuario (Alto, Medio y Bajo).
- **Riesgo en el desarrollo:** Valoración de riesgo de la historia de usuario (Alto, Medio, Bajo).
- **Programador responsable:** Nombre de la persona que debe desarrollar dicha historia de usuario.
- **Puntos estimados:** Días asignados para realizar la historia de usuario.
- **Iteración asignada:** Identificación de la iteración a la cual se le asigna la historia de usuario
- **Descripción:** Información agregada para tener un mayor entendimiento en la historia de usuario.
- **Observación:** Se detalla procesos que estén vinculados con esa historia de usuario. Es opcional.

A continuación, se detallan las historias de usuario necesarias para el desarrollo del sistema web.

Tabla 15. Historia de usuario - Establecer la estructura del sistema.

Historia de Usuario			
Número:	001	Usuario:	Desarrollador
Nombre de la historia:	Establecer la estructura del sistema		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	1
Programador responsable:	Diego Santamaría.		
Descripción:	Estructurar el manejo de datos y el desarrollo web del sistema para tener una mayor organización en el proyecto y poder determinar y aplicar tecnologías y herramientas las cuales servirán para dar mayor funcionalidad y beneficios al sistema.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 16. Historia de usuario - Ingreso al sistema web

Historia de Usuario			
Número:	002	Usuario:	Todos
Nombre de la historia:	Ingreso al sistema web		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	4	Iteración asignada:	1
Programador responsable:	Diego Santamaría.		
Descripción:	En este módulo se pedirá el correo electrónico y la contraseña del usuario previamente registrado para que pueda acceder al sistema web y utilizar las diferentes funciones dependiendo del usuario asignado.		
Observación:	El sistema solo contará con un usuario administrador el cual podrá habilitar o inhabilitar las cuentas existentes en el sistema.		

Elaborado por: Diego Santamaria

Tabla 17. Historia de usuario - Restablecer contraseña

Historia de Usuario			
Número:	003	Usuario:	Todos
Nombre de la historia:	Restablecer contraseña		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	1	Iteración asignada:	1
Programador responsable:	Diego Santamaría.		
Descripción:	Los usuarios del sistema podrán restablecer su contraseña ingresando el correo electrónico que utiliza para ingresar al sistema web.		
Observación:	La ventana de restablecer contraseña se podrá habilitar al momento de seleccionar esta opción que se encuentra en el ingreso al sistema web.		

Elaborado por: Diego Santamaria

Tabla 18. Historia de usuario - Visualización de la pantalla de inicio.

Historia de Usuario			
Número:	004	Usuario:	Todos
Nombre de la historia:	Visualización de la pantalla de inicio		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	4	Iteración asignada:	1
Programador responsable:	Diego Santamaría.		
Descripción:	Pantalla en la cual se visualizarán los módulos y utilidades que tendrá cada usuario del sistema, estos módulos se presentaran de acuerdo al tipo de perfil del usuario que ingrese al sistema web.		
Observación:	En la parte superior de esta pantalla se encontrarán las opciones de ver el perfil del usuario y salir del sistema web.		

Elaborado por: Diego Santamaria

Tabla 19. Historia de usuario - Visualización del perfil de usuario

Historia de Usuario			
Número:	005	Usuario:	Todos
Nombre de la historia:	Visualización del perfil de usuario		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	2	Iteración asignada:	1
Programador responsable:	Diego Santamaría.		
Descripción:	Se visualizará la información proporcionada por el usuario al momento de crear su cuenta.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 20. Historia de usuario - Edición del perfil de usuario.

Historia de Usuario			
Número:	006	Usuario:	Todos
Nombre de la historia:	Edición del perfil de usuario		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	2	Iteración asignada:	2
Programador responsable:	Diego Santamaría.		
Descripción:	Módulo que permitirá editar la información básica del usuario.		
Observación:			

Elaborado por: Diego Santamaria

Tabla 21. Historia de usuario - Visualización de usuarios del sistema

Historia de Usuario			
Número:	007	Usuario:	Administrador
Nombre de la historia:	Visualización de usuarios del sistema		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	2
Programador responsable:	Diego Santamaría.		
Descripción:	Se visualizarán en la pantalla de inicio, todos los usuarios del sistema.		
Observación:	En la lista de usuarios se mostrará la información del usuario como su correo electrónico, teléfono, perfil, etc.		

Elaborado por: Diego Santamaria

Tabla 22. Historia de usuario – Funciones de bloqueo de usuario

Historia de Usuario			
Número:	008	Usuario:	Administrador
Nombre de la historia:	Funciones de bloqueo de usuario		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	2
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá bloquear o desbloquear usuarios del sistema que se encuentren de vacaciones o despedidos.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 23. Historia de usuario - Configuración de registro de usuarios.

Historia de Usuario			
Número:	009	Usuario:	Administrador
Nombre de la historia:	Configuración de registro de usuarios.		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	2
Programador responsable:	Diego Santamaría		
Descripción:	En este módulo se definirá el perfil que los usuarios tendrán al momento de registrarse en el sistema web. También se podrá restringir el registro de usuarios, dependiendo de las necesidades de la empresa.		
Observación:	No se podrá agregar más perfiles ya que el sistema web solo trabajará con los perfiles predeterminados.		

Elaborado por: Diego Santamaria

Tabla 24. Historia de usuario – Reportes

Historia de Usuario			
Número:	010	Usuario:	Administrador
Nombre de la historia:	Reportes		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	3	Iteración asignada:	2
Programador responsable:	Diego Santamaría		
Descripción:	En este módulo se podrá generar los reportes de todos los usuarios del sistema, usuarios activos y usuarios inactivos.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 25. Historia de usuario - Listado de categorías

Historia de Usuario			
Número:	011	Usuario:	Bodeguero
Nombre de la historia:	Listado de categorías de nombres productos		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se mostrarán todas las categorías disponibles en el sistema		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 26. Historia de usuario - Añadir categoría

Historia de Usuario			
Número:	012	Usuario:	Bodeguero
Nombre de la historia:	Añadir categoría		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá añadir las diferentes categorías que ofrece la empresa.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 27. Historia de usuario - Editar categoría

Historia de Usuario			
Número:	013	Usuario:	Bodeguero
Nombre de la historia:	Editar categoría		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá editar el nombre de las categorías que se han ingresado previamente.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 28. Historia de usuario - Eliminar categoría

Historia de Usuario			
Número:	014	Usuario:	Bodeguero
Nombre de la historia:	Eliminar categoría		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá eliminar la categoría elegida por el usuario siempre y cuando no esté siendo empleada en la colección productos.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 29. Historia de usuario - Listado de nombres productos

Historia de Usuario			
Número:	015	Usuario:	Bodeguero
Nombre de la historia:	Listado de nombres productos		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se mostrarán los nombres de los productos existentes en el sistema.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 30. Historia de usuario - Añadir nombre producto

Historia de Usuario			
Número:	016	Usuario:	Bodeguero
Nombre de la historia:	Añadir nombre producto		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá añadir nombres de los productos que se encuentren en venta, para ello se tendrá que seleccionar una categoría existente.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 31. Historia de usuario - Editar nombre producto

Historia de Usuario			
Número:	017	Usuario:	Bodeguero
Nombre de la historia:	Editar nombre producto		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá editar los nombres de los productos que se encuentren en venta.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 32. Historia de usuario - Eliminar nombre producto

Historia de Usuario			
Número:	018	Usuario:	Bodeguero
Nombre de la historia:	Eliminar nombre producto		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	3
Programador responsable:	Diego Santamaría		
Descripción:	Se eliminará el nombres del producto seleccionado por el usuario siempre y cuando el nombre producto no esté siendo usado en la colección productos.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 33. Historia de usuario - Listado de empaques

Historia de Usuario			
Número:	019	Usuario:	Bodeguero
Nombre de la historia:	Listado de empaques		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se mostrará todos los empaques ingresados en el sistema.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 34. Historia de usuario - Añadir empaque

Historia de Usuario			
Número:	020	Usuario:	Bodeguero
Nombre de la historia:	Añadir empaque		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá añadir los empaques utilizados para los productos que se encuentran disponibles.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 35. Historia de usuario - Editar empaque

Historia de Usuario			
Número:	021	Usuario:	Bodeguero
Nombre de la historia:	Editar empaque		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá editar los empaques existentes.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 36. Historia de usuario - Eliminar empaque

Historia de Usuario			
Número:	022	Usuario:	Bodeguero
Nombre de la historia:	Eliminar empaque		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá eliminar los empaques seleccionados que se encuentren en el sistema		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 37. Historia de usuario - Listar productos

Historia de Usuario			
Número:	023	Usuario:	Bodeguero
Nombre de la historia:	Listar productos		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se mostrarán los nombres de los productos creados en el sistema.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 38. Historia de usuario - Agregar producto

Historia de Usuario			
Número:	024	Usuario:	Bodeguero
Nombre de la historia:	Agregar producto		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá agregar productos seleccionando su categoría, nombre y empaque.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 39. Historia de usuario - Editar producto

Historia de Usuario			
Número:	025	Usuario:	Bodeguero
Nombre de la historia:	Editar producto		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá editar productos que previamente hayan sido creados.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 40. Historia de usuario - Eliminar producto

Historia de Usuario			
Número:	026	Usuario:	Bodeguero
Nombre de la historia:	Eliminar producto		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá eliminar productos seleccionados del sistema.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 41. Historia de usuario - Ver pedidos realizados

Historia de Usuario			
Número:	027	Usuario:	Bodeguero
Nombre de la historia:	Ver pedidos realizados		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	Se observarán las características que tiene el pedido generado por el usuario para poder despacharlo.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 42. Historia de usuario - Despachar pedido realizado

Historia de Usuario			
Número:	028	Usuario:	Bodeguero
Nombre de la historia:	Despachar pedido realizado		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	4
Programador responsable:	Diego Santamaría		
Descripción:	En la tabla que se muestran los pedidos ubicada en la pantalla de inicio se encontrara un botón el cual permitirá al usuario marcar ese pedido como despachado para que desaparezca del listado de pedidos pendientes.		
Observación:	Se mostrará un mensaje de advertencia para confirmar el despacho del pedido.		

Elaborado por: Diego Santamaria

Tabla 43. Historia de usuario - Ver pedido marcados como despachados

Historia de Usuario			
Número:	029	Usuario:	Bodeguero
Nombre de la historia:	Ver pedidos marcados como despachados		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá ver todos los pedidos que han sido realizados en el sistema por los vendedores.		
Observación:	Los pedidos se visualizarán al momento de seleccionar un vendedor o por el nombre del cliente.		

Elaborado por: Diego Santamaria

Tabla 44. Historia de usuario - Ingresar kilometraje

Historia de Usuario			
Número:	030	Usuario:	Vendedor
Nombre de la historia:	Ingresar kilometraje		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Al ingresar al sistema el vendedor deberá ingresar el kilometraje con el cual inicia sus actividades		
Observación:	Al finalizar su jornada de trabajo, el vendedor deberá ingresar el kilometraje final.		

Elaborado por: Diego Santamaria

Tabla 45. Historia de usuario - Vista de crear pedidos

Historia de Usuario			
Número:	031	Usuario:	Vendedor
Nombre de la historia:	Vista de crear pedidos		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Se podrán observar los pedidos generados por el vendedor que se encuentren pendientes, así como las diferentes opciones que puede utilizar el vendedor.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 46. Historia de usuario - Vista de visita al cliente

Historia de Usuario			
Número:	032	Usuario:	Vendedor
Nombre de la historia:	Vista de visita al cliente		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Se podrán observar las visitas realizadas a un cliente, registradas por el vendedor.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 47. Historia de usuario - Vista de cobranza

Historia de Usuario			
Número:	033	Usuario:	Vendedor
Nombre de la historia:	Vista de cobranza		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Se podrán observar los cobros realizados por el vendedor.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 48. Historia de usuario - Crear pedido

Historia de Usuario			
Número:	034	Usuario:	Vendedor
Nombre de la historia:	Crear pedido		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	El vendedor podrá crear pedidos llenando los campos solicitados.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 49. Historia de usuario - Editar pedido

Historia de Usuario			
Número:	035	Usuario:	Vendedor
Nombre de la historia:	Editar pedido		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá editar los pedidos que aún no han sido despachados por el bodeguero.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 50. Historia de usuario - Eliminar pedido

Historia de Usuario			
Número:	036	Usuario:	Vendedor
Nombre de la historia:	Eliminar pedido		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	1	Iteración asignada:	5
Programador responsable:	Diego Santamaría		
Descripción:	Se podrá eliminar un pedido.		
Observación:	El bodeguero no debe marcar el pedido como despachado, sino el pedido no podrá ser eliminado.		

Elaborado por: Diego Santamaria

Tabla 51. Historia de usuario - Crear Visita

Historia de Usuario			
Número:	037	Usuario:	Vendedor
Nombre de la historia:	Crear Visita		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	6
Programador responsable:	Diego Santamaría		
Descripción:	El vendedor podrá crear una visita, especificando los capos solicitados.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 52. Historia de usuario - Crear cobranza

Historia de Usuario			
Número:	038	Usuario:	Vendedor
Nombre de la historia:	Crear cobranza		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Alto
Puntos estimados:	2	Iteración asignada:	6
Programador responsable:	Diego Santamaría		
Descripción:	El vendedor podrá crear una cobranza, especificando los campos solicitados, no se puede editar o eliminar la cobranza.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

Tabla 53. Historia de usuario – Reportes

Historia de Usuario			
Número:	039	Usuario:	Gerente
Nombre de la historia:	Reportes		
Prioridad en el negocio:	Alto	Riesgo en el desarrollo:	Medio
Puntos estimados:	4	Iteración asignada:	6
Programador responsable:	Diego Santamaría		
Descripción:	En la pantalla de inicio se podrán observar los diferentes reportes que el usuario podrá generar en el sistema con la información proporcionada por el usuario vendedor.		
Observación:	Ninguna.		

Elaborado por: Diego Santamaria

3.2.1.8 Tareas

Las tareas son la unión de actividades las cuales tienen un inicio y un fin. A estas tareas se les asigna un responsable el cual es el encargado de realizar dicha tarea en el tiempo establecido para que se ejecute correctamente.

A continuación, se generan las tareas que se desarrollarán en el presente proyecto, tomando como referencia las historias de usuario previamente creadas.

Tabla 54. Tarea - Establecer la estructura del sistema web.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T1	Código de historia: 001
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Establecer la estructura del sistema web.	
Fecha de inicio:	Fecha de fin:
Descripción: Se analizarán los requerimientos del sistema para poder definir una estructura adecuada.	

Elaborado por: Diego Santamaria

Tabla 55. Tarea - Diseño de interfaz para el ingreso al sistema web.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T2	Código de historia: 002
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Diseño de interfaz para el ingreso al sistema web.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz para que el usuario pueda iniciar sesión y acceder al sistema web	

Elaborado por: Diego Santamaria

Tabla 56. Tarea - Implementación de ingreso al sistema web.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T3	Código de historia: 002
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Implementación de ingreso al sistema web.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará la acción de iniciar sesión si las credenciales del usuario son correctas.	

Elaborado por: Diego Santamaria

Tabla 57. Tarea - Interfaz de restablecimiento de contraseña.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T4	Código de historia: 003
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de restablecimiento de contraseña.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz la cual permitirá al usuario restablecer su contraseña si fuera necesario.	

Elaborado por: Diego Santamaria

Tabla 58. Tarea - Implementación de restablecimiento de contraseña.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T5	Código de historia: 003
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de restablecimiento de contraseña.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará la acción que permita enviar un correo electrónico para al usuario que solicite restablecer su contraseña.	

Elaborado por: Diego Santamaria

Tabla 59. Tarea - Interfaz de inicio del sistema

Tarea	Responsable: Diego Santamaría
Nº de tarea: T6	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de inicio del sistema.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz principal del sistema cuando el usuario administrador ingrese al sistema web.	

Elaborado por: Diego Santamaria

Tabla 60. Tarea - Implementación de la interfaz de inicio del usuario administrador.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T7	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de la interfaz de inicio del usuario administrador	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará un menú para mostrar las diferentes funcionalidades del usuario administrador.	

Elaborado por: Diego Santamaria

Tabla 61. Tarea - Interfaz del listado y funciones de usuarios del sistema.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T8	Código de historia: 007
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz del listado y funciones de usuarios del sistema.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz en la cual se mostrarán todos los usuarios registrados en el sistema, así como también las vistas para ejecutar diferentes acciones sobre estos datos.	

Elaborado por: Diego Santamaria

Tabla 62. Tarea - Implementación del listado y funciones de usuarios del sistema.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T9	Código de historia: 007
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación del listado y funciones de usuarios del sistema.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará los servicios necesarios para listar, añadir, actualizar y eliminar datos relacionados con las cuentas de los usuarios.	

Elaborado por: Diego Santamaria

Tabla 63. Tarea - Diseño de funciones de bloqueo de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T10	Código de historia: 008
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Diseño de funciones de bloqueo de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará las interfaces para poder administrar las cuentas de usuario del sistema web.	

Elaborado por: Diego Santamaria

Tabla 64. Tarea - Implementación de funciones de bloqueo de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T11	Código de historia: 008
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Implementación de funciones de bloqueo de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará las funcionalidades para poder bloquear o desbloquear a un usuario del sistema web.	

Elaborado por: Diego Santamaria

Tabla 65. Tarea - Interfaz de perfil de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T12	Código de historia: 005
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Interfaz de perfil de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de perfil de usuario la cual se encontrará información del usuario.	

Elaborado por: Diego Santamaria

Tabla 66. Tarea - Implementación del perfil de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T13	Código de historia: 005
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Implementación del perfil de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará los servicios que permitan visualizar la información del usuario que se encuentre en el sistema.	

Elaborado por: Diego Santamaria

Tabla 67. Tarea - Interfaz de edición de perfil de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T14	Código de historia: 006
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Interfaz de edición de perfil de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz que permita editar la información del usuario del sistema.	

Elaborado por: Diego Santamaria

Tabla 68. Tarea - Implementación de edición de perfil de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T15	Código de historia: 006
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Implementación de edición de perfil de usuario	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará los servicios para actualizar la nueva información del usuario.	

Elaborado por: Diego Santamaria

Tabla 69. Tarea - Interfaz de configuración de registro de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T16	Código de historia: 009
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Interfaz de configuración de registro de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz para que el usuario pueda configurar el registro de usuarios al sistema.	

Elaborado por: Diego Santamaria

Tabla 70. Tarea - Implementación de configuración de registro de usuario.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T17	Código de historia: 009
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Implementación de configuración de registro de usuario.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán los servicios que permitan modificar ciertos parámetros para el registro de un usuario al sistema.	

Elaborado por: Diego Santamaria

Tabla 71. Tarea - Interfaz de reportes de usuarios del sistema.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T18	Código de historia: 010
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Interfaz de reportes de usuarios del sistema.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz que permita al usuario generar reportes de todos los usuarios del sistema, así como también los usuarios activos e inactivos.	

Elaborado por: Diego Santamaria

Tabla 72. Tarea - Implementación de reportes de usuarios del sistema.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T19	Código de historia: 010
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Implementación de reportes de usuarios del sistema.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará las funcionalidades y servicios para generar los diferentes reportes de los usuarios existentes en el sistema.	

Elaborado por: Diego Santamaria

Tabla 73. Tarea - Interfaz de inicio del usuario bodeguero.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T20	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de inicio del usuario bodeguero.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz principal del sistema cuando el usuario bodeguero ingrese al sistema web.	

Elaborado por: Diego Santamaria

Tabla 74. Tarea - Implementación de pantalla de inicio del usuario bodeguero.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T21	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de pantalla de inicio del usuario bodeguero.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará un menú el cual mostrara todas las funcionalidades que tiene el usuario, así como el listado de los pedidos generados por los usuarios con la funcionalidad de ver el pedido y poder marcarlo como despachado.	

Elaborado por: Diego Santamaria

Tabla 75. Tarea - Interfaz de categorías de productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T22	Código de historia: 011, 012, 013, 014
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Interfaz de categorías de productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de categorías la cual permitirá realizar las diferentes acciones a las categorías existentes en el sistema.	

Elaborado por: Diego Santamaria

Tabla 76. Tarea - Implementación de categorías de productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T23	Código de historia: 011, 012, 013, 014
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Implementación de categorías de productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará las funciones de listar, añadir, editar y eliminar las categorías de los productos del sistema.	

Elaborado por: Diego Santamaria

Tabla 77. Tarea - Interfaz de nombres productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T24	Código de historia: 015, 016, 017, 018
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Interfaz de nombres productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de nombres de productos la cual permitirá realizar las diferentes acciones a los nombres de los productos existentes en el sistema.	

Elaborado por: Diego Santamaria

Tabla 78. Tarea - Implementación de nombres productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T25	Código de historia: 015, 016, 017, 018
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Implementación de nombres productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funciones de listar, añadir, editar y eliminar los nombres de los productos del sistema.	

Elaborado por: Diego Santamaria

Tabla 79. Tarea - Interfaz de empaques de productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T26	Código de historia: 019, 020, 021, 022
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Interfaz de empaques de productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de empaques la cual permitirá realizar las diferentes acciones a los empaques existentes en el sistema.	

Elaborado por: Diego Santamaria

Tabla 80. Tarea - Implementación de empaques de productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T27	Código de historia: 019, 020, 021, 022
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Implementación de empaques de productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funciones de listar, añadir, editar y eliminar los empaques del sistema.	

Elaborado por: Diego Santamaria

Tabla 81. Tarea - Interfaz de productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T28	Código de historia: 023, 024, 025, 026
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Interfaz de productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de productos la cual permitirá realizar las diferentes acciones a los productos existentes en el sistema.	

Elaborado por: Diego Santamaria

Tabla 82. Tarea - Implementación de productos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T29	Código de historia: 023, 024, 025, 026
Tipo de tarea: Desarrollo	Puntos estimados: 4
Nombre de tarea: Implementación de productos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funciones de listar, añadir, editar y eliminar los productos del sistema.	

Elaborado por: Diego Santamaria

Tabla 83. Tarea - Interfaz de los pedidos realizados.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T30	Código de historia: 027, 028
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de los pedidos realizados.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de pedidos realizados por los vendedores.	

Elaborado por: Diego Santamaria

Tabla 84. Tarea - Implementación de pedidos realizados.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T31	Código de historia: 027, 028
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de pedidos realizados.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funcionalidades para poder observar los detalles de cada pedido realizado, así como marcar como despacharlo, una vez que se haya finalizado la venta.	

Elaborado por: Diego Santamaria

Tabla 85. Tarea - Interfaz de pedidos despachados.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T32	Código de historia: 029
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Interfaz de pedidos despachados.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz de pedidos despachados la cual permitirá utilizar las distintas funcionalidades que tiene.	

Elaborado por: Diego Santamaria

Tabla 86. Tarea - Implementación de pedidos despachados.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T33	Código de historia: 029
Tipo de tarea: Desarrollo	Puntos estimados: 2
Nombre de tarea: Implementación de pedidos despachados.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará las funciones para poder buscar un pedido despachado por el bodeguero.	

Elaborado por: Diego Santamaria

Tabla 87. Tarea - Interfaz de inicio del usuario vendedor.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T34	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de inicio del usuario vendedor.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz principal del sistema cuando el usuario vendedor ingrese al sistema web.	

Elaborado por: Diego Santamaria

Tabla 88. Tarea - Implementación de pantalla de inicio del usuario vendedor.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T35	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de pantalla de inicio del usuario vendedor.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará un menú el cual muestre las diferentes tareas que puede realizar el usuario.	

Elaborado por: Diego Santamaria

Tabla 89. Tarea - Interfaz para crear pedidos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T36	Código de historia: 031, 034, 035, 036
Tipo de tarea: Desarrollo	Puntos estimados: 7
Nombre de tarea: Interfaz para crear pedidos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz para poder crear un pedido en el sistema.	

Elaborado por: Diego Santamaria

Tabla 90. Tarea - Implementación para crear pedidos.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T37	Código de historia: 031, 034, 035, 036
Tipo de tarea: Desarrollo	Puntos estimados: 6
Nombre de tarea: Implementación para crear pedidos.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funciones de listar, añadir, editar y eliminar los pedidos del sistema.	

Elaborado por: Diego Santamaria

Tabla 91. Tarea - Interfaz de visita al cliente.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T38	Código de historia: 032, 037, 038, 039
Tipo de tarea: Desarrollo	Puntos estimados: 6
Nombre de tarea: Interfaz de visita al cliente.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz para poder crear una visita al cliente cada vez que un vendedor realice esta acción.	

Elaborado por: Diego Santamaria

Tabla 92. Tarea - Implementación de visita al cliente.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T39	Código de historia: 032, 037, 038, 039
Tipo de tarea: Desarrollo	Puntos estimados: 6
Nombre de tarea: Implementación de visita al cliente.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funciones de listar, añadir, editar y eliminar las visitas realizadas a los clientes.	

Elaborado por: Diego Santamaria

Tabla 93. Tarea - Interfaz de cobranza.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T40	Código de historia: 033, 040, 041, 042
Tipo de tarea: Desarrollo	Puntos estimados: 6
Nombre de tarea: Interfaz de cobranza.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz para poder crear una cobranza hecha al cliente cada vez que un vendedor realice esta acción.	

Elaborado por: Diego Santamaria

Tabla 94. Tarea - Implementación de cobranza.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T41	Código de historia: 033, 040, 041, 042
Tipo de tarea: Desarrollo	Puntos estimados: 6
Nombre de tarea: Implementación de cobranza.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funciones de listar, añadir, editar y eliminar las cobranzas realizadas a los clientes.	

Elaborado por: Diego Santamaria

Tabla 95. Tarea - Interfaz de ingreso de kilometraje.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T42	Código de historia: 030
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de ingreso de kilometraje.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz para ingresar el kilometraje con el cual inicia un vendedor su jornada de trabajo.	

Elaborado por: Diego Santamaria

Tabla 96. Tarea - Implementación de ingreso de kilometraje.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T43	Código de historia: 030
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de ingreso de kilometraje.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementarán las funcionalidades necesarias para poder ingresar el kilometraje actual del vehículo del vendedor.	

Elaborado por: Diego Santamaria

Tabla 97. Tarea - Interfaz de inicio del usuario gerente.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T44	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Interfaz de inicio del usuario gerente.	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñará la interfaz principal del sistema cuando el usuario gerente ingrese al sistema web.	

Elaborado por: Diego Santamaria

Tabla 98. Tarea - Implementación de la pantalla de inicio.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T45	Código de historia: 004
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de la pantalla de inicio.	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará un menú en el cual se podrán generar diferentes reportes por el usuario.	

Elaborado por: Diego Santamaria

Tabla 99. Tarea - Diseño de reportes de vendedores.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T46	Código de historia: 043
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Diseño de reportes de vendedores	
Fecha de inicio:	Fecha de fin:
Descripción: Se diseñarán las interfaces para poder generar los diferentes reportes que se requieren en el sistema.	

Elaborado por: Diego Santamaria

Tabla 100. Tarea - Implementación de reportes de vendedores.

Tarea	Responsable: Diego Santamaría
Nº de tarea: T47	Código de historia: 043
Tipo de tarea: Desarrollo	Puntos estimados: 1
Nombre de tarea: Implementación de reportes de vendedores	
Fecha de inicio:	Fecha de fin:
Descripción: Se implementará las funcionalidades necesarias para poder generar los reportes de los vendedores solicitados por el usuario gerente.	

Elaborado por: Diego Santamaria

3.2.1.9 Valoración de historias de usuario

Una vez analizados y evaluados los requerimientos y tareas de las historia de usuario, se llegó a la conclusión de que se emplearán 6 horas en cada día laboral, obteniendo así, un total de 30 horas semanales

3.2.1.10 Estimación de historias de usuario

Iteración 1

Tabla 101. Estimación iteración 1.

		Tiempo estimado	
Número	Historia de usuario	Días	Horas
01	Establecer la estructura del sistema	1	2
02	Ingreso al sistema web	4	24
03	Restablecer contraseña	1	4
04	Visualización de la pantalla de inicio	1	5
05	Visualización del perfil de usuario	2	12
Tiempo total		9	47

Elaborado por: Diego Santamaria

Iteración 2

Tabla 102. Estimación iteración 2.

		Tiempo estimado	
Número	Historia de usuario	Días	Horas
06	Edición del perfil de usuario	2	12
07	Visualización de usuarios del sistema	1	6
08	Funciones de bloqueo de usuario	2	12
09	Configuración de registro de usuarios	2	12
10	Reportes	3	18
Tiempo total		10	60

Elaborado por: Diego Santamaria

Iteración 3

Tabla 103. Estimación iteración 3.

		Tiempo estimado	
Número	Historia de usuario	Días	Horas
11	Listado de categorías	1	3
12	Añadir categorías	1	6
13	Editar categorías	1	6
14	Eliminar categorías	1	2

15	Listado de productos	1	3
16	Añadir nombre producto	1	6
17	Editar nombre producto	1	6
18	Eliminar nombre producto	1	2
Tiempo total		8	34

Elaborado por: Diego Santamaria

Iteración 4

Tabla 104. Estimación iteración 4.

Número	Historia de usuario	Tiempo estimado	
		Días	Horas
19	Listado de empaques	1	3
20	Añadir empaque	1	6
21	Editar empaque	1	6
22	Eliminar empaque	1	2
23	Listar productos	1	3
24	Agregar producto	1	6
25	Editar producto	1	6
26	Eliminar producto	1	2
27	Ver pedidos realizados	1	4
28	Despachar pedidos realizados	1	4
Tiempo total		10	42

Elaborado por: Diego Santamaria

Iteración 5

Tabla 105. Estimación iteración 5.

Número	Historia de usuario	Tiempo estimado	
		Días	Horas
29	Ver pedidos marcados como despachados	2	12
30	Ingresar kilometraje	1	6
31	Vista de crear pedidos	1	4
32	Vista de visita al cliente	1	4
33	Vista de cobranza	1	4
34	Crear pedido	2	9
35	Editar pedido	2	9
36	Eliminar pedido	1	6
Tiempo total		11	54

Elaborado por: Diego Santamaria

Iteración 6

Tabla 106. Estimación iteración 6.

		Tiempo estimado	
Número	Historia de usuario	Días	Horas
37	Reportes	4	24
Tiempo total		4	7

Elaborado por: Diego Santamaria

3.2.1.11 Plan de entrega

Tabla 107. Plan de entrega.

N°	Historia de usuario	Tiempo estimado		Iteración asignada						Entrega asignada					
		Días	Horas	1	2	3	4	5	6	1	2	3	4	5	6
01	Establecer la estructura del sistema	1	2	X							X				
02	Ingreso al sistema web	4	24	X							X				
03	Restablecer contraseña	1	4	X							X				
04	Visualización de la pantalla de inicio	1	5	X							X				
05	Visualización del perfil de usuario	2	12	X							X				
06	Edición del perfil de usuario	2	12		X							X			
07	Visualización de usuarios del sistema	1	6		X							X			
08	Funciones de bloqueo de usuario	2	12		X							X			
09	Configuración de registro de usuarios	2	12		X							X			
10	Reportes	3	18		X							X			
11	Listado de categorías	1	3			X							X		
12	Añadir categorías	1	6			X							X		
13	Editar categorías	1	6			X							X		
14	Eliminar categorías	1	2			X							X		
15	Listado de productos	1	3			X							X		
16	Añadir nombre producto	1	6			X							X		
17	Editar nombre producto	1	6			X							X		
18	Eliminar nombre producto	1	2			X							X		
19	Listado de empaques	1	3				X							X	
20	Añadir empaque	1	6				X							X	

Tabla 108. Plan de iteraciones.

Iteración	N°	Historia	Prioridad	Riesgo	Estado de desarrollo	Pruebas
1	01	Establecer la estructura del sistema	1	Alto	Completo	Aprobado
	02	Ingreso al sistema web	1	Alto	Completo	Aprobado
	03	Restablecer contraseña	4	Medio	Completo	Aprobado
	04	Visualización de la pantalla de inicio	1	Medio	Completo	Aprobado
	05	Visualización del perfil de usuario	1	Medio	Completo	Aprobado
2	06	Edición del perfil de usuario	1	Medio	Completo	Aprobado
	07	Visualización de usuarios del sistema	1	Alto	Completo	Aprobado
	08	Funciones de bloqueo de usuario	1	Alto	Completo	Aprobado
	09	Configuración de registro de usuarios	1	Alto	Completo	Aprobado
	10	Reportes	1	Medio	Completo	Aprobado
3	11	Listado de categorías	1	Medio	Completo	Aprobado

	12	Añadir categorías	1	Alto	Completo	Aprobado
	13	Editar categorías	1	Alto	Completo	Aprobado
	14	Eliminar categorías	1	Alto	Completo	Aprobado
	15	Listado de productos	1	Medio	Completo	Aprobado
	16	Añadir nombre producto	1	Alto	Completo	Aprobado
	17	Editar nombre producto	1	Alto	Completo	Aprobado
	18	Eliminar nombre producto	1	Alto	Completo	Aprobado
4	19	Listado de empaques	1	Medio	Completo	Aprobado
	20	Añadir empaque	1	Alto	Completo	Aprobado
	21	Editar empaque	1	Alto	Completo	Aprobado
	22	Eliminar empaque	1	Alto	Completo	Aprobado
	23	Listar productos	1	Medio	Completo	Aprobado
	24	Agregar producto	1	Alto	Completo	Aprobado
	25	Editar producto	1	Alto	Completo	Aprobado
	26	Eliminar producto	1	Alto	Completo	Aprobado

	27	Ver pedidos realizados	1	Alto	Completo	Aprobado
	28	Despachar pedidos realizados	1	Alto	Completo	Aprobado
5	29	Ver pedidos marcados como despachados	1	Alto	Completo	Aprobado
	30	Ingresar kilometraje	1	Alto	Completo	Aprobado
	31	Vista de crear pedidos	1	Alto	Completo	Aprobado
	32	Vista de visita al cliente	1	Alto	Completo	Aprobado
	33	Vista de cobranza	1	Alto	Completo	Aprobado
	34	Crear pedido	1	Alto	Completo	Aprobado
	35	Editar pedido	1	Alto	Completo	Aprobado
	36	Eliminar pedido	1	Alto	Completo	Aprobado
6	37	Reportes	1	Medio	Completo	Aprobado

Elaborado por: Diego Santamaria

3.2.2 Fase II: Diseño

3.2.2.1 Metáfora

La aplicación web progresiva desarrollada para la empresa Ambagro S.A. proporcionará diferentes beneficios para los empleados que intervienen en el proceso de gestión de pedidos, así como también el control de los vendedores de la empresa.

El sistema web contará con una base de datos NoSQL la cual permitirá guardar o presentar información necesaria para cada usuario de la aplicación web. A continuación, se detallan los módulos que conforman la aplicación web progresiva:

1. **Módulo de seguridad:** Los usuarios deben estar registrados en el sistema web, así como también tener habilitada su cuenta para poder ingresar a la aplicación.
2. **Módulo de información de usuario:** Visualización de la información personal que colocaron los usuarios al momento de crear su cuenta en la aplicación web.
3. **Módulo de gestión de cuentas de usuario:** Se administrarán todas las cuentas de los usuarios del sistema, inhabilitar cuentas, habilitar cuentas y especificar parámetros para el registro de nuevos usuarios.
4. **Módulo de creación de pedidos:** El usuario con el perfil de “vendedor” podrá crear nuevos pedidos llenando un formulario parametrizado.
5. **Módulo de visita al cliente:** El usuario con el perfil de “vendedor” podrá generar una visita al cliente cuando esté realizando esta acción, especificando ciertos parámetros.
6. **Módulo de cobranza al cliente:** El usuario con el perfil de “vendedor” podrá generar una cobranza que se haya hecho a un cliente especificando los parámetros establecidos.
7. **Módulo de añadir productos:** Se podrá añadir productos ofrecidos por la empresa especificando su categoría, nombre, empaque y precio.
8. **Módulo de gestión de pedidos generados:** El usuario “bodeguero” podrá visualizar y despachar los pedidos de generados por los vendedores de la empresa.
9. **Módulo de Búsqueda de pedidos despachados:** El usuario “bodeguero” podrá buscar un pedido despachado anteriormente especificando ciertos parámetros previamente solicitados.

10. Módulo de control de vendedores: El usuario “gerente” podrá visualizar todos los pedidos, visitas y cobranzas generados por los vendedores de la empresa, verificando su ubicación geográfica en un mapa. En este módulo se podrá comunicar directamente vía mail, o WhatsApp el usuario gerente con el vendedor que registre alguna anomalía en su trabajo.

11. Módulo de reportaría: Visualización de la información necesaria para cada usuario que requiera un reporte.

3.2.2.2 Iteraciones

A continuación, se procede a realizar las iteraciones establecidas referentes a las historias de usuario

Iteración 1

Tabla 109. Planificación de iteración 1

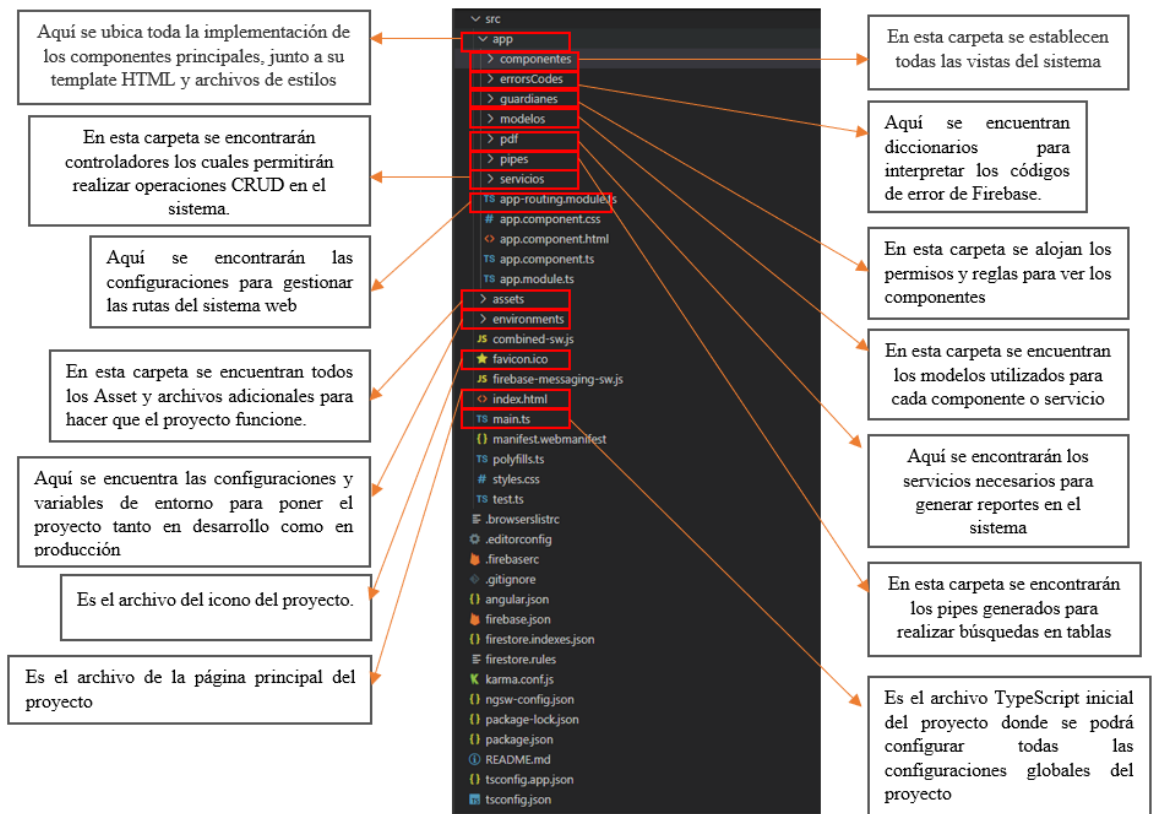
Número	Historia de usuario	Prioridad	Riesgo
01	Establecer la estructura del sistema	1	Alto
02	Ingreso al sistema web	1	Alto
03	Restablecer contraseña	4	Medio
04	Visualización de la pantalla de inicio	1	Medio
05	Visualización del perfil de usuario	1	Medio

Elaborado por: Diego Santamaria

N.º 01: Establecer la estructura del sistema

El presente proyecto se lo desarrollará con el Framework Angular el cual posee una estructura para sus proyectos muy simple y sencilla la cual separa los diferentes recursos de una forma estructurada para un desarrollo más amigable.

Figura 42. Estructura del sistema web con Angular

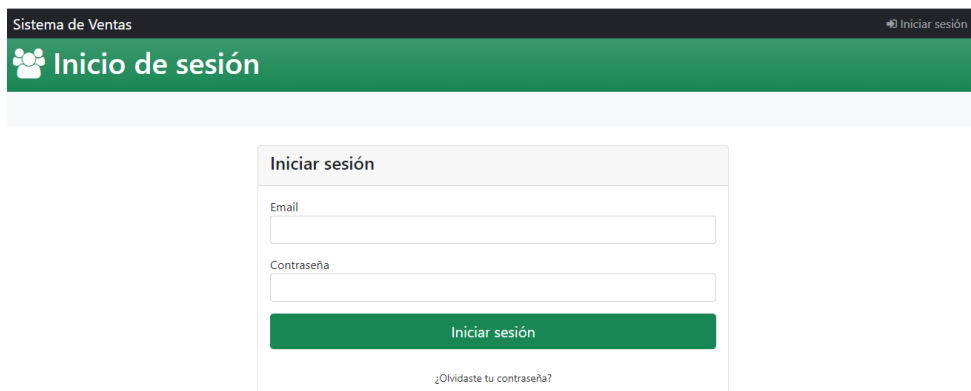


Elaborado por: Diego Santamaria

N.º 02: Ingreso al sistema web

Para ingresar al sistema web se debe establecer el correo electrónico y contraseña ingresado por el usuario al momento de crear su cuenta (Figura 14).

Figura 43. Interfaz de inicio de sesión

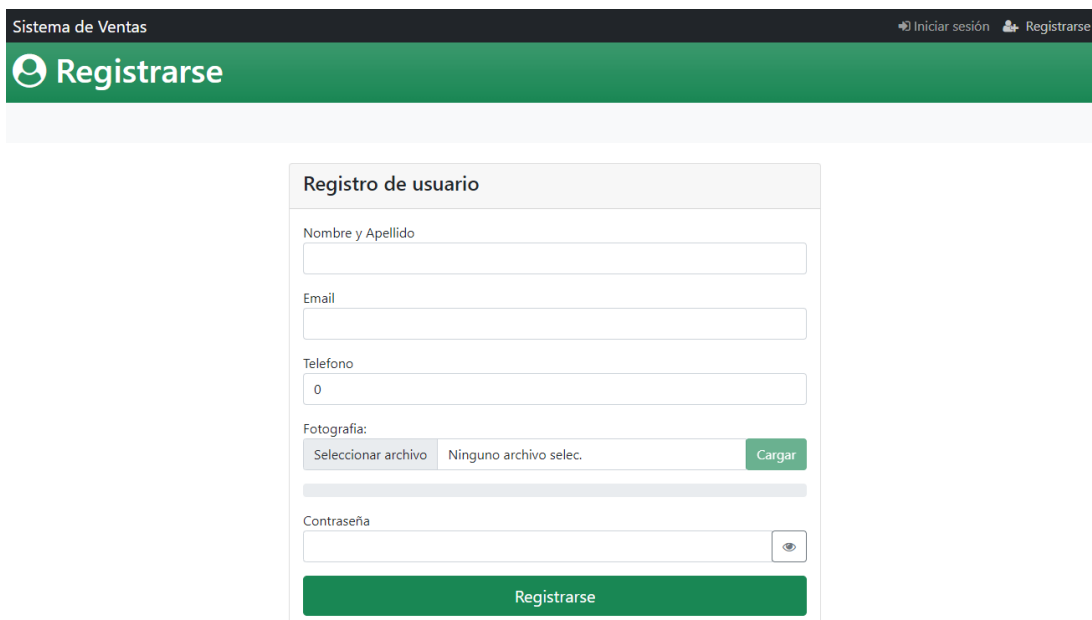


The screenshot shows the login page of the 'Sistema de Ventas' web application. The header includes the text 'Sistema de Ventas' on the left and 'Iniciar sesión' on the right. Below the header is a green banner with a white icon of three people and the text 'Inicio de sesión'. The main content area features a white box titled 'Iniciar sesión' containing two input fields: 'Email' and 'Contraseña'. Below these fields is a green button labeled 'Iniciar sesión' and a link that says '¿Olvidaste tu contraseña?'.

Elaborado por: Diego Santamaria

Al momento de registrar usuarios en el sistema, el usuario administrador deberá habilitar esta opción. Una vez culminado el registro de usuarios, el usuario administrador debe deshabilitar la opción de registro de usuarios.

Figura 44. Interfaz de registros de usuarios.



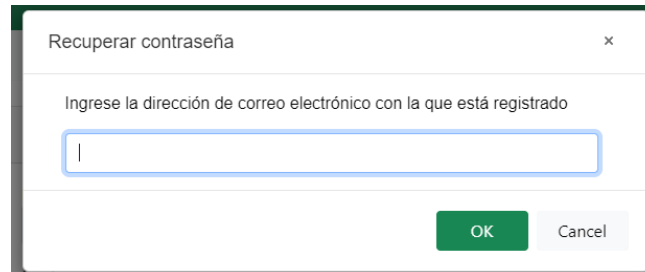
The screenshot shows the registration page of the 'Sistema de Ventas' web application. The header includes the text 'Sistema de Ventas' on the left and 'Iniciar sesión' and 'Registrarse' on the right. Below the header is a green banner with a white icon of a person and the text 'Registrarse'. The main content area features a white box titled 'Registro de usuario' containing several input fields: 'Nombre y Apellido', 'Email', 'Telefono' (with the value '0'), and 'Fotografía:'. The 'Fotografía:' field includes a 'Seleccionar archivo' button, the text 'Ninguno archivo selec.', and a 'Cargar' button. Below these fields is a 'Contraseña' field with a toggle icon. At the bottom of the box is a green button labeled 'Registrarse'.

Elaborado por: Diego Santamaria

N.º 03: Restablecer contraseña

Los usuarios del sistema podrán reestablecer la contraseña de su cuenta ingresando el correo electrónico con el cual se han registrado en el sistema web.

Figura 45. Interfaz de restablecer contraseña de usuario



A modal window titled "Recuperar contraseña" with a close button (x) in the top right corner. The main text inside the modal says "Ingrese la dirección de correo electrónico con la que está registrado". Below this text is a text input field with a blue border and a vertical cursor. At the bottom right of the modal, there are two buttons: a green "OK" button and a grey "Cancel" button.

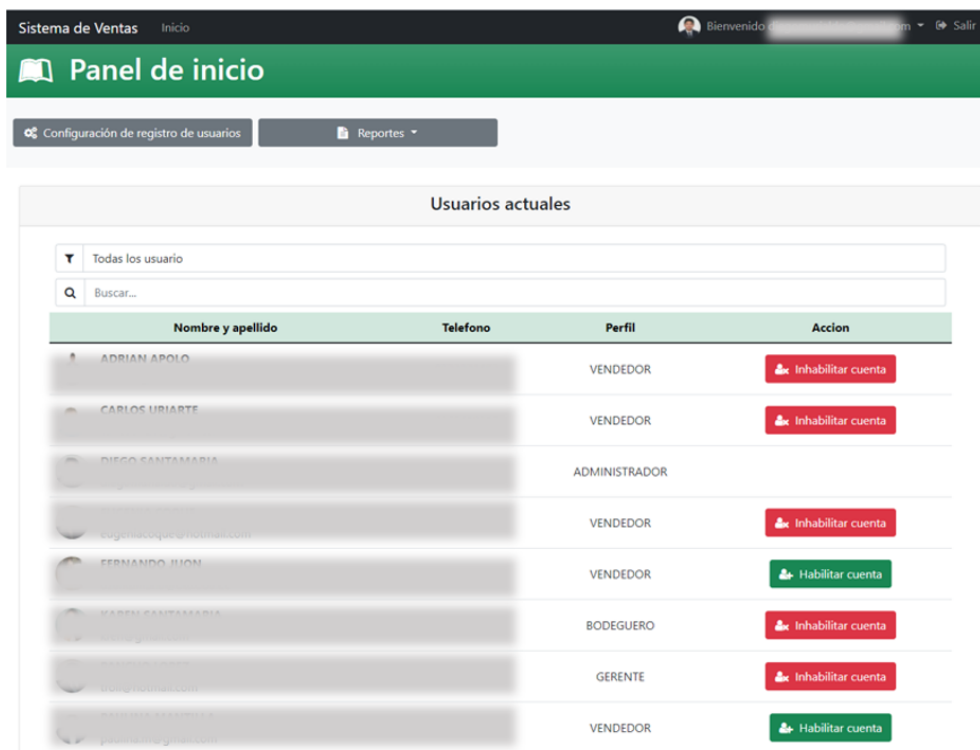
Elaborado por: Diego Santamaria

N.º 04: Visualización de la pantalla de inicio

Al momento de ingresar al sistema web, se desplegarán los diferentes componentes de inicio, en dependencia de los perfiles de cada usuario.

- Interfaz de inicio del perfil administrador

Figura 46. Interfaz de inicio del usuario Administrador



The dashboard for the Administrator profile is titled "Sistema de Ventas Inicio" and "Bienvenido". It features a green header with "Panel de inicio" and a navigation bar with "Configuración de registro de usuarios" and "Reportes". The main content area is titled "Usuarios actuales" and contains a search bar and a table of active users.

Nombre y apellido	Telefono	Perfil	Accion
ADRIAN APOLO		VENDEDOR	Inhabilitar cuenta
CARLOS UBIARTE		VENDEDOR	Inhabilitar cuenta
DIEGO SANTAMARIA		ADMINISTRADOR	
EUGENIO QUINONES		VENDEDOR	Inhabilitar cuenta
FERNANDO ILIOM		VENDEDOR	Habilitar cuenta
HAROLD SANTAMARIA		BODEGUERO	Inhabilitar cuenta
JOSE RODRIGUEZ		GERENTE	Inhabilitar cuenta
JOSE RODRIGUEZ		VENDEDOR	Habilitar cuenta

Elaborado por: Diego Santamaria

- Interfaz de inicio del perfil bodeguero

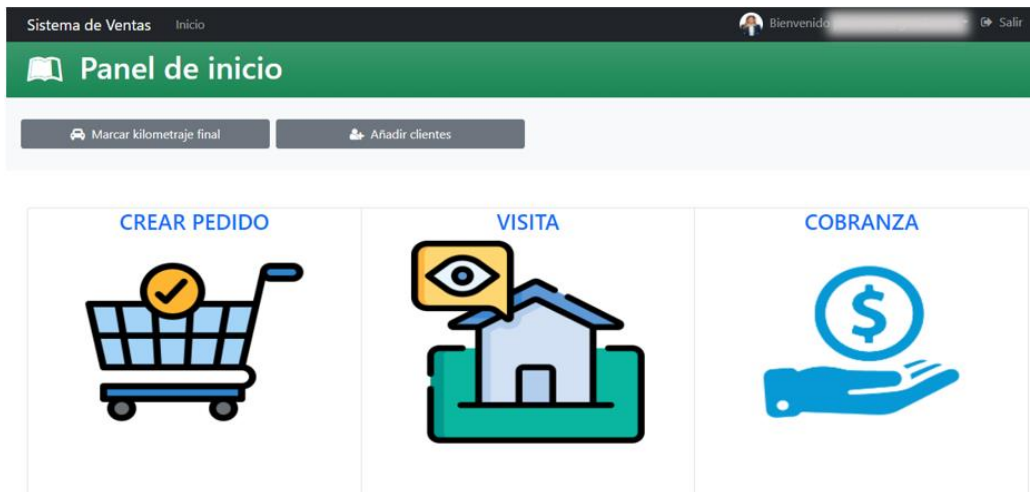
Figura 47. Interfaz de inicio del usuario bodeguero



Elaborado por: Diego Santamaria

- Interfaz de inicio del perfil vendedor

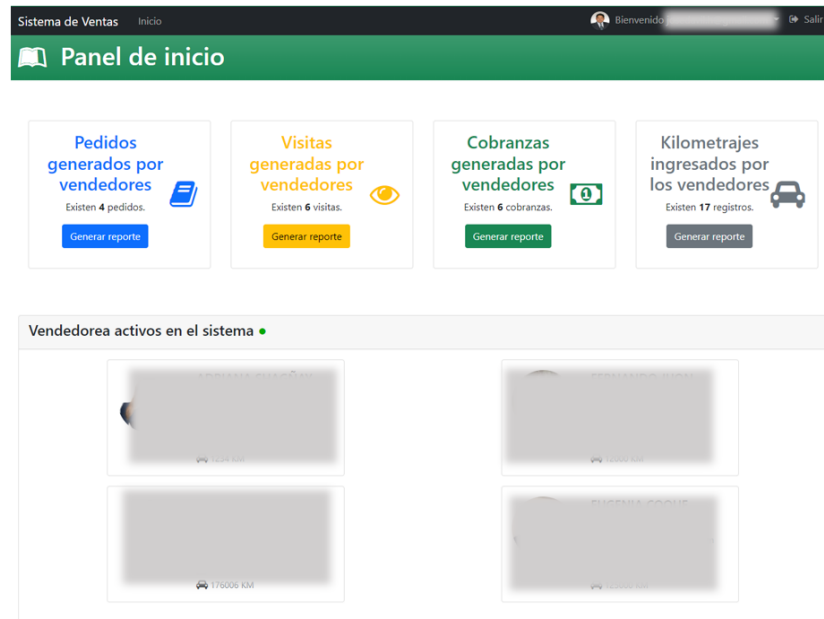
Figura 48. Interfaz de inicio del usuario Vendedor



Elaborado por: Diego Santamaria

- Interfaz de inicio del perfil gerente

Figura 49. Interfaz de inicio del usuario gerente

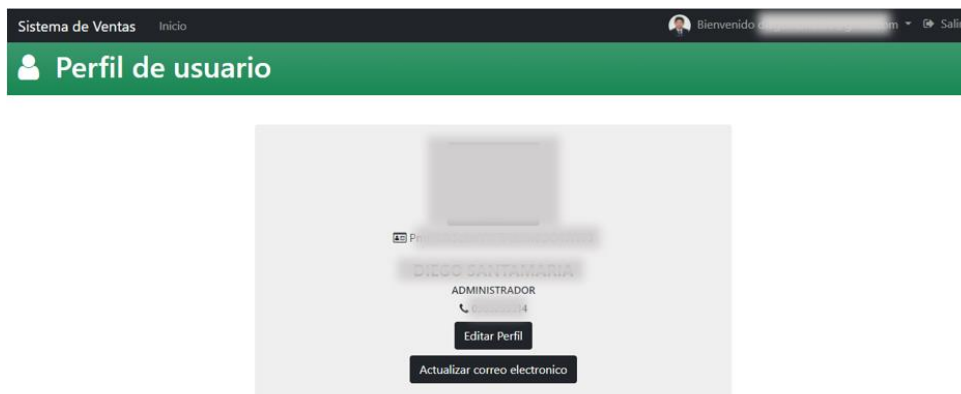


Elaborado por: Diego Santamaria

N.º 05: Visualización del perfil de usuario

Cada usuario podrá ver su información ingresada en el sistema, así como también actualizarla.

Figura 50. Interfaz del perfil del usuario



Elaborado por: Diego Santamaria

Iteración 2

Tabla 110. Planificación de iteración 2

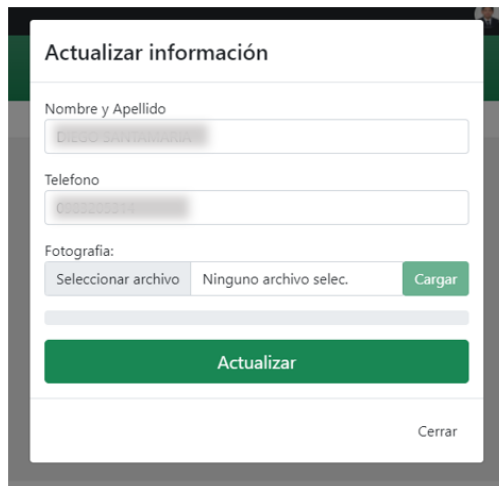
Número	Historia de usuario	Prioridad	Riesgo
06	Edición del perfil de usuario	1	Medio
07	Visualización de usuarios del sistema	1	Alto
08	Funciones de bloqueo de usuario	1	Alto
09	Configuración de registro de usuarios	1	Alto
10	Reportes	1	Medio

Elaborado por: Diego Santamaria

N.º 06: Edición del perfil de usuario

El usuario podrá actualizar casi todo su información si esta cambiase o fuese errónea.

Figura 51. Interfaz de edición de información del usuario



The screenshot shows a mobile application interface for updating user information. The title is 'Actualizar información'. It features three input fields: 'Nombre y Apellido' with the placeholder 'Diego Santamaria', 'Telefono' with the placeholder '0333333333', and 'Fotografía:'. Below the photo field are two buttons: 'Seleccionar archivo' and 'Cargar'. At the bottom, there is a large green 'Actualizar' button and a smaller 'Cerrar' button in the bottom right corner.

Elaborado por: Diego Santamaria

Figura 52. Interfaz de actualización del correo electrónico del usuario

Elaborado por: Diego Santamaria

N.º 07: Visualización de usuarios del sistema

El usuario administrador podrá observar todos los usuarios del sistema para posteriormente realizar más acciones.

Figura 53. Visualización de usuarios existentes en el sistema

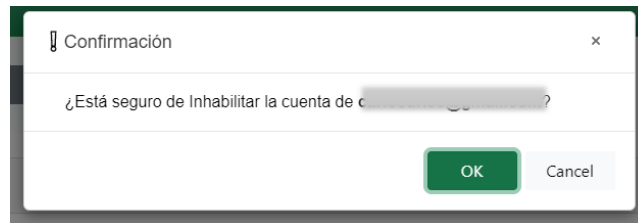
Nombre y apellido	Telefono	Perfil	Accion
ADRIAN APOLO		VENDEDOR	Inhabilitar cuenta
CARLOS URIARTE		VENDEDOR	Inhabilitar cuenta
Diego Santamaria		ADMINISTRADOR	
...		VENDEDOR	Inhabilitar cuenta
FERNANDO IJON		VENDEDOR	Habilitar cuenta
MARCELO SANTAMARIA		BODEGUERO	Inhabilitar cuenta
...		GERENTE	Inhabilitar cuenta
...		VENDEDOR	Habilitar cuenta

Elaborado por: Diego Santamaria

N.º 08: Funciones de bloqueo de usuario

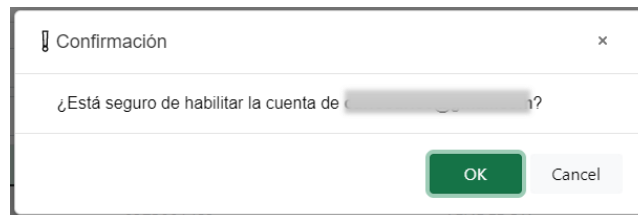
El usuario administrador podrá bloquear o habilitar las cuentas de todos los usuarios del sistema, excepto la de él.

Figura 54. Función de bloqueo de usuario



Elaborado por: Diego Santamaria

Figura 55. Función de desbloqueo de usuario

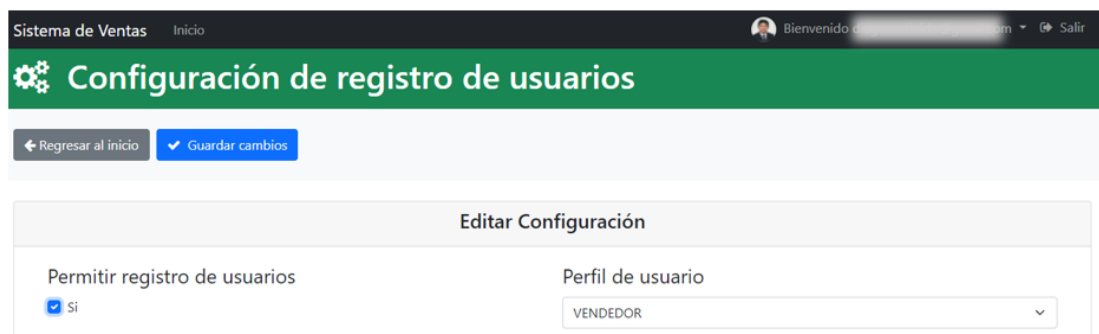


Elaborado por: Diego Santamaria

N.º 09: Configuración de registro de usuarios

En la parte de configuración de registro de usuarios se podrá especificar si los empleados de la empresa pueden o no registrarse en el sistema, aquí también se puede especificar el tipo de perfil con el que se va a registrar el nuevo usuario.

Figura 56. Interfaz de configuración de registro de usuarios.



Elaborado por: Diego Santamaria

N.º 10: Reportes

El usuario administrador podrá generar reportes de los usuarios del sistema en dependencia al estado de estos.

Figura 57. Interfaz de reportaría de usuarios existentes del sistema

Sistema de Ventas Inicio Bienvenido a [Nombre] Salir

Reporte de todos los usuarios

Regresar al inicio Imprimir Guardar como PDF

Vista previa

USUARIOS EXISTENTE EN EL SISTEMA DE CONTROL DE VENTAS

Fecha: 3/5/2022 E-mail: [Email]
Usuario: [Usuario] Telefono: [Telefono]

#	ID	Nombre y apellido	Email	Telefono	Perfil	Estado
1	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
2	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
3	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
4	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
5	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	Desactivado
6	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
7	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
8	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]

Elaborado por: Diego Santamaria

Figura 58. Interfaz de reportaría de usuarios habilitados del sistema

Sistema de Ventas Inicio Bienvenido a [Nombre] Salir

Reporte de usuarios Habilitados

Regresar al inicio Imprimir Guardar como PDF

Vista previa

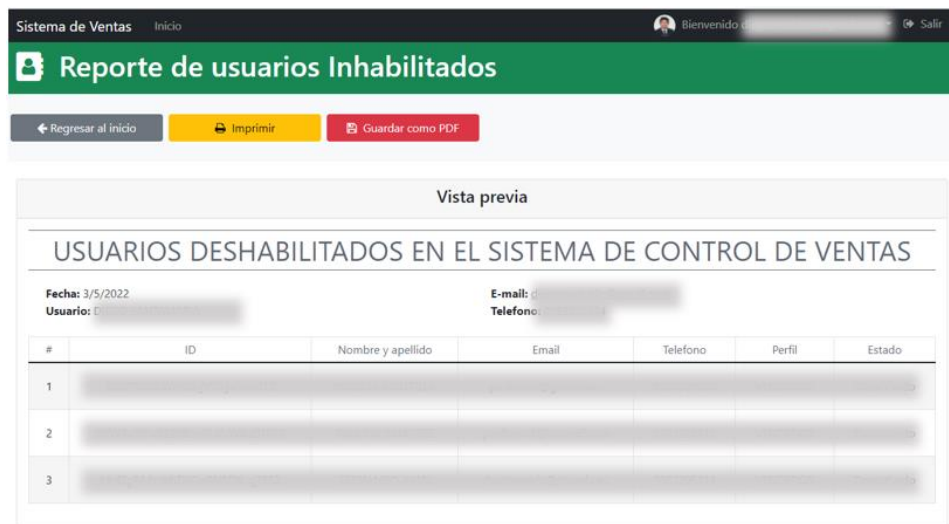
USUARIOS HABILITADOS EN EL SISTEMA DE CONTROL DE VENTAS

Fecha: 3/5/2022 E-mail: [Email]
Usuario: [Usuario] Telefono: [Telefono]

#	ID	Nombre y apellido	Email	Telefono	Perfil	Estado
1	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
2	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
3	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
4	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
5	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]
6	[ID]	[Nombre y apellido]	[Email]	[Telefono]	[Perfil]	[Estado]

Elaborado por: Diego Santamaria

Figura 59. Interfaz de reportaría de usuarios inhabilitados del sistema



Elaborado por: Diego Santamaria

Iteración 3

Tabla 111. Planificación de iteración 3

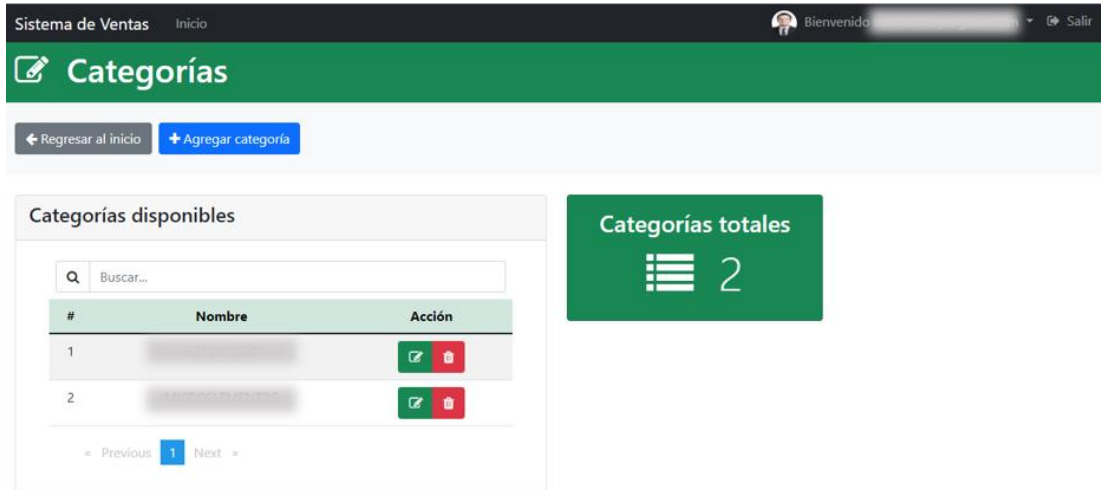
Número	Historia de usuario	Prioridad	Riesgo
11	Listado de categorías	1	Medio
12	Añadir categorías	1	Alto
13	Editar categorías	1	Alto
14	Eliminar categorías	1	Alto
15	Listado de nombre productos	1	Medio
16	Añadir nombre producto	1	Alto
17	Editar nombre producto	1	Alto
18	Eliminar nombre producto	1	Alto

Elaborado por: Diego Santamaria

N.º 11: Listado de categorías

El usuario bodeguero podrá visualizar todas las categorías que se encuentran en el sistema web.

Figura 60. Interfaz de listado de categorías

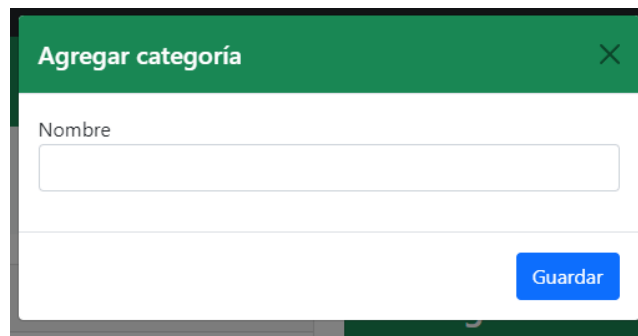


Elaborado por: Diego Santamaria

N.º 12: Añadir categorías

El usuario bodeguero podrá añadir categorías al sistema web.

Figura 61. Añadir categoría



Elaborado por: Diego Santamaria

N.º 13: Editar categorías

El usuario bodeguero podrá editar las categorías que se encuentran ingresadas en el sistema web.

Figura 62. Editar categoría



Actualizar categoría

Nombre

BIOESTIMULANTES

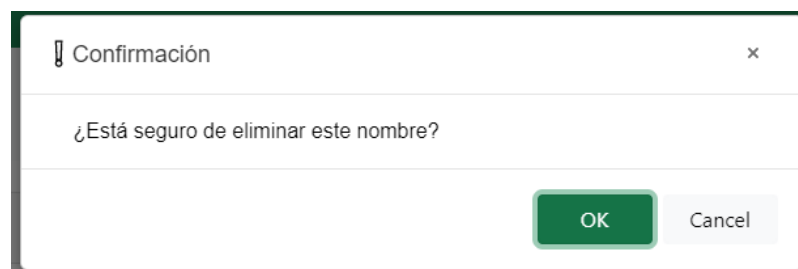
Guardar

Elaborado por: Diego Santamaria

N.º 14: Eliminar categorías

El usuario bodeguero puede eliminar las categorías del sistema siempre y cuando esta no esté asociada a otro registro.

Figura 63. Eliminar categoría



Confirmación

¿Está seguro de eliminar este nombre?

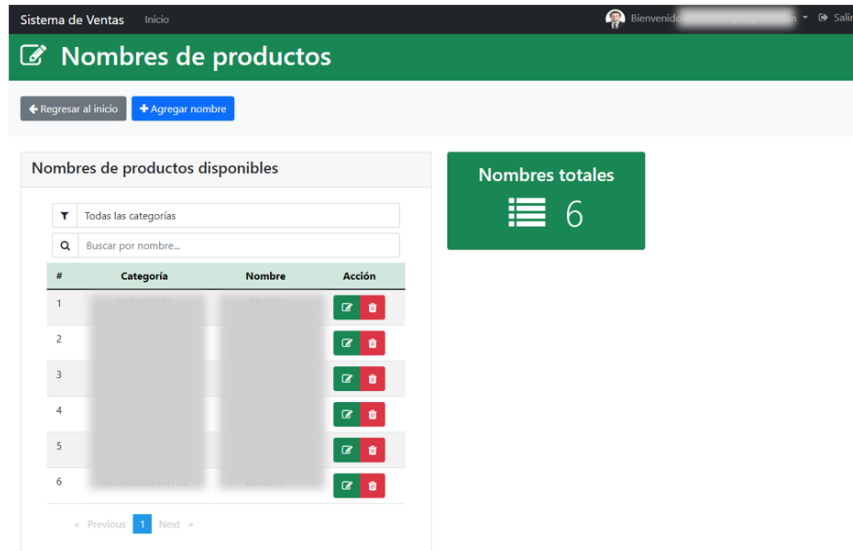
OK Cancel

Elaborado por: Diego Santamaria

N.º 15: Listado de nombre productos

El usuario bodeguero podrá visualizar todos los nombres productos que se encuentran en el sistema web.

Figura 64. Listar nombre producto



Elaborado por: Diego Santamaria

N.º 16: Añadir nombre producto

El usuario bodeguero podrá añadir nombres de productos al sistema web.

Figura 65. Añadir nombre producto

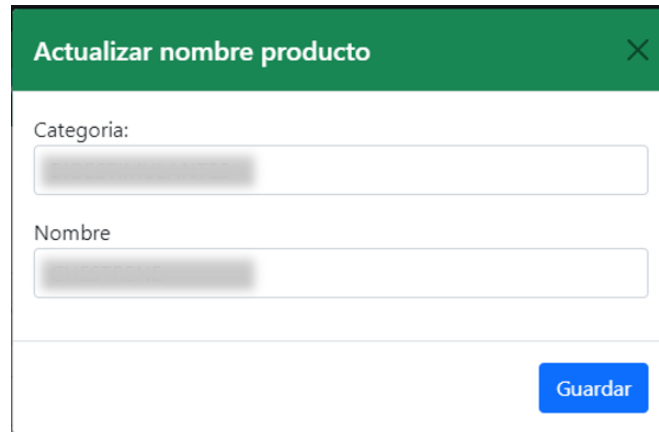
The screenshot shows a modal window titled 'Agregar nombre de producto'. It contains two input fields: a dropdown menu for 'Categoría' with the placeholder text 'Seleccione una categoría', and a text input field for 'Nombre'. At the bottom right of the modal is a blue button labeled 'Guardar'.

Elaborado por: Diego Santamaria

N.º 17: Editar nombre producto

El usuario bodeguero podrá editar los nombre de productos que se encuentran ingresadas en el sistema web.

Figura 66. Editar nombre producto



Actualizar nombre producto

Categoría:

Nombre

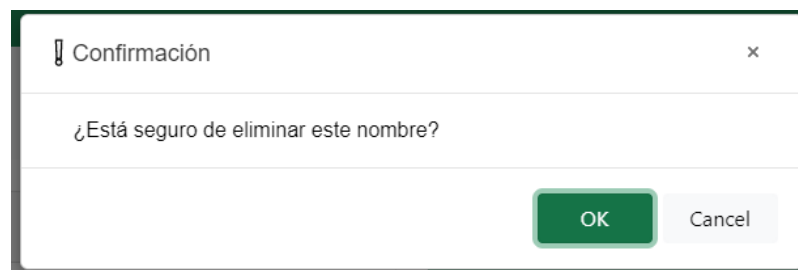
Guardar

Elaborado por: Diego Santamaria

N.º 18: Eliminar nombre producto

El usuario bodeguero puede eliminar los nombres de productos del sistema siempre y cuando esta no esté asociada a otro registro.

Figura 67. Eliminar nombre producto



Confirmación

¿Está seguro de eliminar este nombre?

OK Cancel

Elaborado por: Diego Santamaria

Iteración 4

Tabla 112. Planificación de iteración 4

Número	Historia de usuario	Prioridad	Riesgo
19	Listado de empaques	1	Medio
20	Añadir empaque	1	Alto
21	Editar empaque	1	Alto
22	Eliminar empaque	1	Alto
23	Listar productos	1	Medio
24	Agregar producto	1	Alto
25	Editar producto	1	Alto
26	Eliminar producto	1	Alto
27	Ver pedidos realizados	1	Alto
28	Despachar pedidos realizados	1	Alto

Elaborado por: Diego Santamaria

N.º 19: Listado de empaques

El usuario bodeguero podrá visualizar todos los empaques que se encuentran en el sistema web.

Figura 68. Lista de empaques

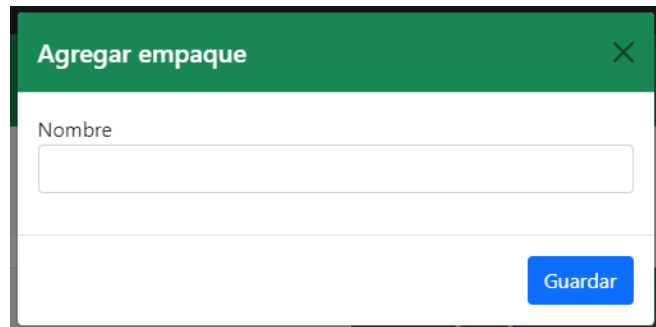
The screenshot displays the 'Empaques' (Packages) section of the 'Sistema de Ventas' (Sales System) web application. At the top, there is a navigation bar with 'Inicio' (Home) and a user profile. Below the header, there are two buttons: 'Regresar al inicio' (Return to home) and 'Agregar empaque' (Add package). The main content area is divided into two sections. On the left, 'Empaques disponibles' (Available packages) features a search bar and a table with 5 rows. Each row contains a package number, a blurred package name, and two action icons: a green edit icon and a red delete icon. On the right, a green summary card shows 'Empaques totales' (Total packages) with a list icon and the number '5'. At the bottom of the table, there are navigation controls: '< Previous', a highlighted '1', and 'Next >'.

Elaborado por: Diego Santamaria

N.º 20: Añadir empaque

El usuario bodeguero podrá añadir empaques al sistema web.

Figura 69. Añadir empaque



El formulario 'Añadir empaque' tiene un encabezado verde con el título 'Añadir empaque' y un ícono de cerrar (X). El cuerpo del formulario es blanco y contiene un campo de texto etiquetado 'Nombre' que está vacío. En la parte inferior derecha del formulario hay un botón azul con el texto 'Guardar'.

Elaborado por: Diego Santamaria

N.º 21: Editar empaque

El usuario bodeguero podrá editar los empaques que se encuentran ingresadas en el sistema web.

Figura 70. Editar empaque



El formulario 'Actualizar empaque' tiene un encabezado verde con el título 'Actualizar empaque' y un ícono de cerrar (X). El cuerpo del formulario es blanco y contiene un campo de texto etiquetado 'Nombre' que contiene un texto desenfocado. En la parte inferior derecha del formulario hay un botón azul con el texto 'Guardar'.

Elaborado por: Diego Santamaria

N.º 22: Eliminar empaque

El usuario bodeguero puede eliminar los empaques del sistema siempre y cuando esta no esté asociada a otro registro.

Figura 71. Eliminar empaque

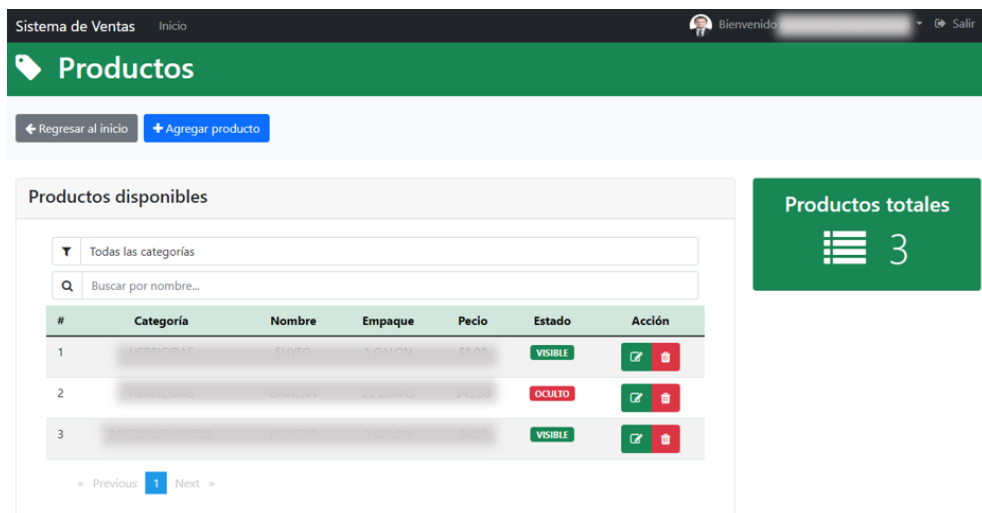


Elaborado por: Diego Santamaria

N.º 23: Listar productos

El usuario bodeguero podrá visualizar todos los productos que se encuentran en el sistema web.

Figura 72. Listar productos

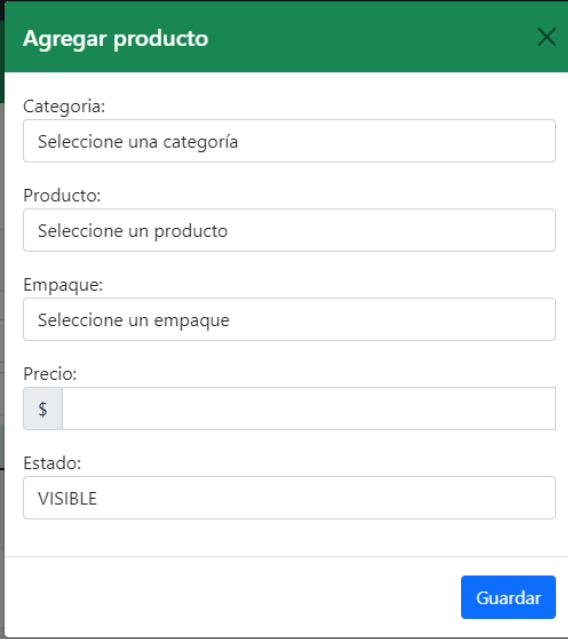


Elaborado por: Diego Santamaria

N.º 24: Añadir producto

El usuario bodeguero podrá añadir productos al sistema web.

Figura 73. Agregar producto



El formulario 'Agregar producto' tiene un encabezado verde con el título 'Agregar producto' y un ícono de cerrar (X). Los campos de entrada son:

- Categoría:** Un campo de texto con el placeholder 'Seleccione una categoría'.
- Producto:** Un campo de texto con el placeholder 'Seleccione un producto'.
- Empaque:** Un campo de texto con el placeholder 'Seleccione un empaque'.
- Precio:** Un campo de texto con un ícono de dólar (\$) a la izquierda.
- Estado:** Un campo de texto con el valor 'VISIBLE'.

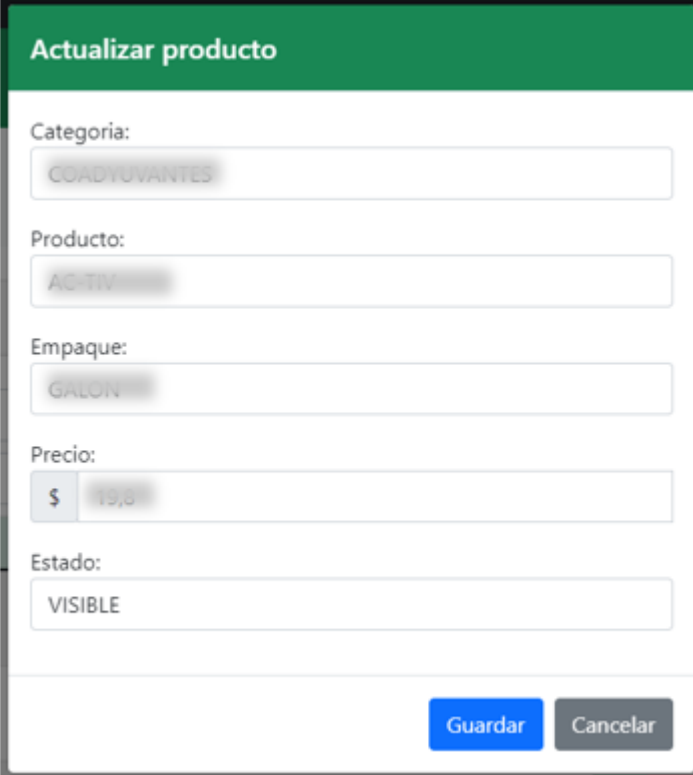
En la parte inferior derecha del formulario hay un botón azul con el texto 'Guardar'.

Elaborado por: Diego Santamaria

N.º 25: Editar producto

El usuario bodeguero podrá editar los productos que se encuentran ingresadas en el sistema web.

Figura 74. Editar producto



Actualizar producto

Categoría:
COADYUVANTES

Producto:
ACTIV

Empaque:
GALON

Precio:
\$ 19,00

Estado:
VISIBLE


Guardar Cancelar

Elaborado por: Diego Santamaria

N.º 26: Eliminar producto

El usuario bodeguero puede eliminar los productos del sistema siempre y cuando esta no esté asociada a otro registro.

Figura 75. Eliminar producto



Confirmación

¿Está seguro de eliminar este producto?

OK Cancel

Elaborado por: Diego Santamaria

N.º 27: Ver pedidos realizados

El usuario bodeguero podrá visualizar todos los pedidos creados por los vendedores, si aún no están despachados.

Figura 76. Ver pedidos realizados

#	RUC CLIENTE	CLIENTE	VENDEDOR	DIRECCIÓN	FECHA DE CREACIÓN	HORA DE CREACIÓN	TOTAL	ACCIÓN
1	1804237830001	LA POETA CARRERA FRANCISCO	EUGENIA		26-06-2022	16:15:40	275.41	
2	0101586740001	CARRERA FRANCISCO		CIENICA	26-06-2022	16:17:40	5858.33	

Elaborado por: Diego Santamaria

N.º 28: Despachar pedidos realizados

El usuario bodeguero puede ver el pedido generado por un vendedor, imprimir el pedido o guardarlo como PDF. Una vez despachado el producto se puede marcar el pedido como despachado para que desaparezca de la lista de pedidos pendientes.

Figura 77. Despachar pedidos realizados

The screenshot shows a web application interface for 'Sistema de Ventas'. At the top, there is a navigation bar with 'Inicio' and a user profile 'Bienvenido...'. Below this is a green header with a 'Pedido' icon and label. A secondary navigation bar contains buttons for 'Regresar al inicio', 'Imprimir', 'Guardar como PDF', and 'Despachado'. The main content area is titled 'Vista previa' and displays a preview of a 'NOTA DE PEDIDO N°1'. This preview includes fields for 'Ruc:', 'Cliente:', 'Nombre comercial:', 'Dirección:', 'Teléfono:', 'Vendedor:', 'Id:', 'Fecha:', 'Hora:', 'Días credito:', and 'Total: \$...'. Below the form fields is a table with the following columns: 'Cant.', 'Presentación', 'Producto', 'Cajas', 'Precio', 'Desc.', and 'Subtotal'. The table contains several rows of data, which are partially obscured by blurring in the image.

Elaborado por: Diego Santamaria

Iteración 5

Tabla 113. Planificación de iteración 5

Número	Historia de usuario	Prioridad	Riesgo
29	Ver pedidos marcados como despachados	1	Alto
30	Ingresar kilometraje	1	Alto
31	Vista de crear pedidos	1	Alto
32	Vista de visita al cliente	1	Alto
33	Vista de cobranza	1	Alto
34	Crear pedido	1	Alto
35	Editar pedido	1	Alto
36	Eliminar pedido	1	Alto

Elaborado por: Diego Santamaria

N.º 29: Ver pedidos marcados como despachados

El usuario bodeguero podrá buscar un pedido que se encuentre registrado en el sistema web.

Figura 78. Interfaz de ver pedidos marcados como despachados

Sistema de Ventas Inicio Bienvenido [Nombre] Salir

Buscar pedidos

← Regresar al inicio

Busqueda

Buscar por: Cliente

Cliente: Nombre: Nombre comercial:

Desde: Hasta: Estado:

#	NUM. PEDIDO	RUC CLIENTE	CLIENTE	VENDEDOR	DIRECCIÓN	FECHA DE CREACIÓN	TOTAL	ACCIÓN
No existen búsquedas.								

Next >

Registros de búsqueda totales: 0

Elaborado por: Diego Santamaria

N.º 30: Ingresar kilometraje

El usuario vendedor tendrá que ingresar el kilometraje de su vehículo con el cual se inicie la jornada laboral para poder utilizar el sistema web.

Figura 79. Interfaz de ingreso de kilometraje inicial

Sistema de Ventas Inicio Bienvenido [Nombre] Salir

Panel de inicio

KILOMETRAJE DE VEHICULO

Ingrese el kilometraje actual de su vehículo para iniciar el sistema.

Elaborado por: Diego Santamaria

N.º 31: Vista de crear pedidos

El usuario vendedor podrá ver sus pedidos, o crear un nuevo pedido en este módulo.

Figura 80. Interfaz de inicio del módulo de crear pedidos

Sistema de Ventas Inicio Bienvenido Salir

Pedidos

Regresar al inicio + Crear pedido

Pedidos creados y no despachados

Buscar por RUC/CI, nombre del cliente...

#	RUC/CI	CLIENTE	TELÉFONO	FECHA	TOTAL	ACCIÓN
No existen pedidos disponibles.						

< Previous 1 Next >

Pedidos totales

0

Elaborado por: Diego Santamaria

N.º 32: Vista de visita al cliente

El usuario vendedor podrá crear una visita ingresando los campos solicitados.

Figura 81. Interfaz del módulo Visita

Sistema de Ventas Inicio Bienvenido Salir

Visita

Crear Visita

Cliente:

Nombre:

Nombre comercial:

Dirección:

Teléfono:

Fecha:

Descripción:
Escriba el motivo de su visita...

Elaborado por: Diego Santamaria

N.º 33: Vista de cobranza

El usuario vendedor podrá crear una cobranza ingresando los campos solicitados.

Figura 82. Interfaz del módulo Cobranza

Sistema de Ventas Inicio Bienvenido Salir

Cobranza

Crear Cobranza

Cliente: Nombre: Nombre comercial:

Dirección: Teléfono: Fecha:

Número de factura: Número de recibo de cobro:

Tipo de cobranza: Concepto: Monto:

Elaborado por: Diego Santamaria

N.º 34: Crear pedido

El usuario vendedor podrá crear un nuevo pedido llenando los campos solicitados.

Figura 83. Interfaz para crear un pedido

Sistema de Ventas Inicio Bienvenido Salir

Crear pedido

Cliente: Nombre: Nombre comercial:

Dirección: Fecha: Días credito:

Productos agregados

#	PRODUCTO	EMPAQUE	PRECIO	CANTIDAD	DESCUENTO	SUBTOTAL	ACCIÓN
No se han agregado productos.							

Next >

Total \$0.00

Elaborado por: Diego Santamaria

N.º 35: Editar pedido

El usuario vendedor podrá editar un pedido que haya sido creado previamente.

Figura 84. Editar pedidos

The screenshot shows the 'Editar pedido' (Edit order) screen in a sales system. The interface is divided into several sections:

- Header:** 'Sistema de Ventas Inicio' and 'Bienvenido [usuario]'.
- Form Fields:** Fields for 'Cliente', 'Nombre', 'Nombre comercial', 'Dirección', 'Fecha', and 'Días credito'.
- Productos agregados:** A table with columns: #, PRODUCTO, EMPAQUE, PRECIO, CANTIDAD, DESCUENTO, SUBTOTAL, and ACCIÓN. It contains two rows of products.
- Total:** A green box showing the total amount with a dollar sign and a plus sign to add more items.
- Buttons:** 'Guardar' (Save) and 'Cancelar' (Cancel) buttons at the bottom right.

#	PRODUCTO	EMPAQUE	PRECIO	CANTIDAD	DESCUENTO	SUBTOTAL	ACCIÓN
1	CAJAS	LITRO	\$41.00	5	0%	\$165.00	[Edit] [Delete]
2	DOVEY	GALON	\$74.20	12	20%	\$713.28	[Edit] [Delete]

Elaborado por: Diego Santamaria

N.º 36: Eliminar pedido

El usuario vendedor podrá eliminar un pedido que haya sido creado y no haya sido despachado por el bodeguero.

Figura 85. Eliminar pedido

The screenshot shows a confirmation dialog box titled 'Confirmación'. The text inside asks: '¿Está seguro de eliminar este productopedido?'. There are two buttons at the bottom: 'OK' (green) and 'Cancel' (grey).

Elaborado por: Diego Santamaria

Iteración 6

Tabla 114. Planificación de iteración 6

Número	Historia de usuario	Prioridad	Riesgo
37	Reportes	1	Medio

Elaborado por: Diego Santamaria

N.º 37: Reportes

El usuario gerente puede generar reportes de los pedidos, visitas y cobranzas generados por los vendedores del sistema.

Figura 86. Interfaz de generar reporte de pedido

Sistema de Ventas Inicio Bienvenido Salir

Buscar pedidos

← Regresar al inicio

Busqueda

Buscar por:
Cliente

Cliente: Nombre: Nombre comercial:

Desde: Hasta: Estado:

01/06/2022 23/06/2022 Despachado

Buscar

Buscar por Ruc, cliente o vendedor...

#	NUM. PEDIDO	RUC CLIENTE	CLIENTE	VENDEDOR	DIRECCIÓN	FECHA DE CREACIÓN	TOTAL	ACCIÓN
No existen búsquedas.								

Next =>

Registros de búsqueda totales 0

Elaborado por: Diego Santamaria

Reporte de pedidos generado por el usuario gerente

Figura 87. Reporte de pedido

Imprimir Guardar como PDF Contactar vendedor

Vista previa

REPORTE DEL PEDIDO # 3

Ruc: [redacted] Vendedor: [redacted]
Cliente: [redacted] Fecha: [redacted]
Nombre comercial: [redacted] Hora: [redacted]
Dirección: [redacted] Dias credito: [redacted]
Telefono: [redacted] Despachado: [redacted]

Total: [redacted]

Ubicación de la creación del pedido

Productos agregados a la compra

#	Categoría	Producto	Empaque	Cantidad	Precio	Descuento	Subtotal
1							

Elaborado por: Diego Santamaria

Interfaz para generar reportes de visitas

Figura 88. Interfaz de generar reporte de visitas

Sistema de Ventas Inicio Bienvenido Salir

Buscar Visitas

Regresar al inicio

Busqueda

Buscar por: Cliente

Cliente: [input] Nombre: [input] Nombre comercial: [input]

Desde: 01/06/2022 Hasta: 23/06/2022

Buscar

Buscar por Ruc, cliente o vendedor...

#	NUM. VISITA	RUC CLIENTE	CLIENTE	VENDEDOR	DIRECCIÓN	FECHA DE CREACIÓN	ACCIÓN
No existen búsquedas.							

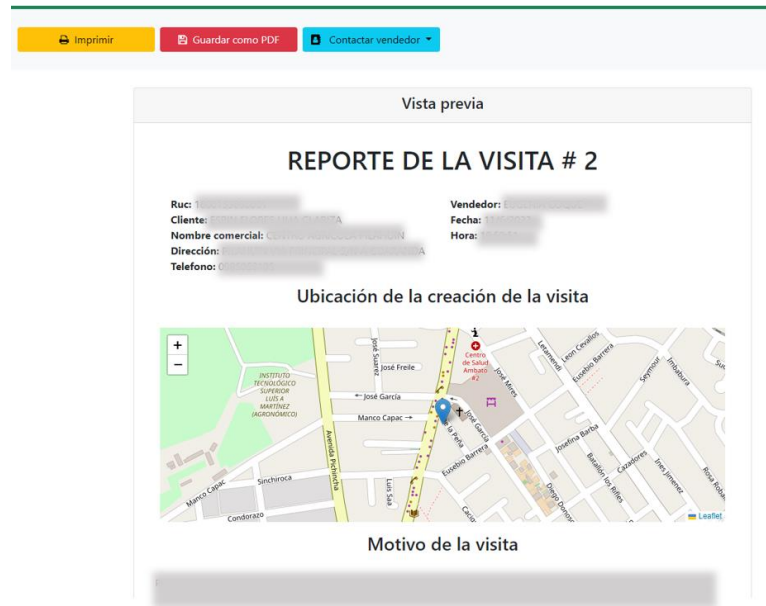
Previous 1 Next

Registros de búsqueda totales: 0

Elaborado por: Diego Santamaria

Reporte de visitas generado por el usuario gerente

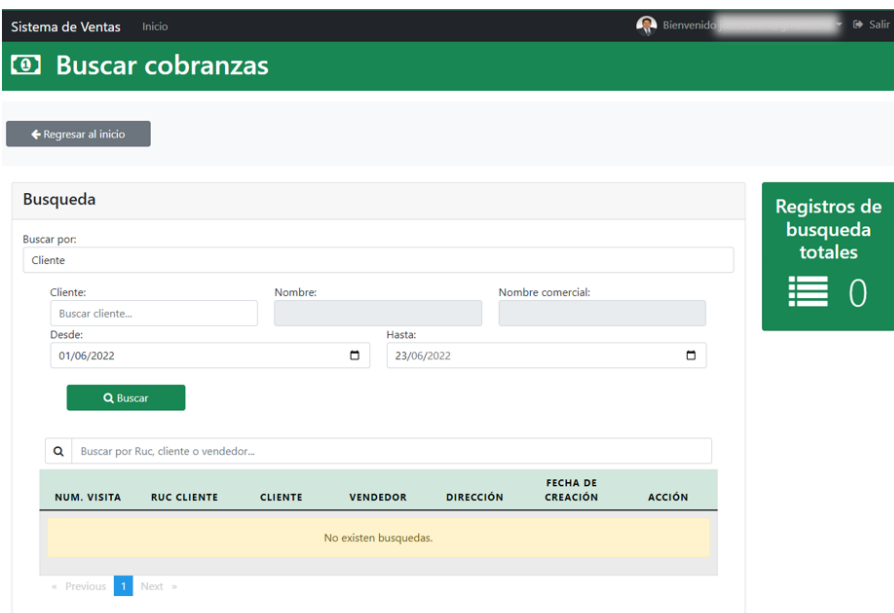
Figura 89. Reporte de Visita



Elaborado por: Diego Santamaria

Interfaz para generar reportes de cobranzas

Figura 90. Interfaz de generar reporte de cobranzas



Elaborado por: Diego Santamaria

Reporte de cobranza generado por el usuario gerente

Figura 91. Reporte de Cobranza

Imprimir Guardar como PDF Contactar vendedor

Vista previa

REPORTE DE LA COBRANZA # 6

Ruc: Vendedor:
Cliente: Fecha:
Nombre comercial: Hora:
Dirección:
Telefono: 0212345678

Ubicación de la creación de la cobranza

Número de factura: Número de comprobante:
Tipo de cobranza: Concepto:
Monto:

Elaborado por: Diego Santamaria

Interfaz para generar reportes de kilometrajes

Figura 92. Interfaz de generar reporte de kilometrajes

Sistema de Ventas Inicio Bienvenido Salir

Buscar kilometrajes

Regresar al inicio

Busqueda

Vendedor: Email: Telefono:
Desde: 01/06/2022 Hasta: 23/06/2022
Buscar

Buscar por Ruc, cliente o vendedor...

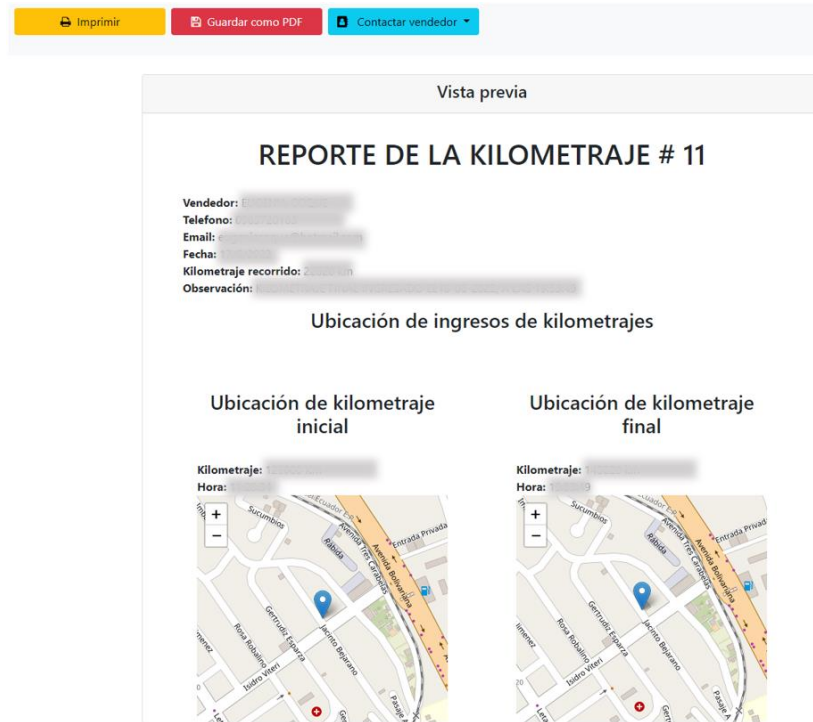
#	FECHA DE CREACION	KILOMETRAJE INICIAL	KILOMETRAJE FINAL	KILOMETRAJE RECORRIDO	ACCIÓN
No existen búsquedas.					

Registros de busqueda totales: 0

Elaborado por: Diego Santamaria

Reporte de kilometraje generado por el usuario gerente

Figura 93. Reporte de Kilometraje



Elaborado por: Diego Santamaria

3.2.3 Fase III: Codificación

Conexión del proyecto Firebase con en framework Angular

Archivo environment y environment production los cuales alojan las credenciales del proyecto creado en Firebase en el cual se encuentran todas las herramientas y características las cuales se usaron para realizar el sistema web.

Figura 94. Conexión Firebase con Angular

```
export const environment = {  
  production: true,  
  firestore: {  
    apiKey: "AIzaSyA...",  
    authDomain: "...",  
    projectId: "...",  
    storageBucket: "...",  
    messagingSenderId: "...",  
    appId: "1:...",  
    measurementId: "G-..."  
  }  
};
```

Elaborado por: Diego Santamaria

Modelos y servicios Back-End

Servicios login

Creación de un nuevo usuario con el servicio de autenticación ofrecido por el Back-End Firebase. Una vez creado el nuevo usuario, se guarda en la base de datos Firestore con el método “saveUser”.

Figura 95. Método crear un nuevo usuarios

```
signUp(usuario: Usuario, password: string) {
  return new Promise((resolve) => {
    this.authService
      .createUserWithEmailAndPassword(usuario.email!, password)
      .then(async (res) => {
        resolve(res);
        let uid = res.user?.uid;
        this.saveUser(usuario, uid!)
          .then()
          .catch((error) => {
            console.log(error);
          });
      });
    .catch((error) => {
      this.flashMessages.show(
        this.errorCode.firebaseErrors[error.code] || error.message,
        { cssClass: 'alert-danger', timeout: 4000 }
      );
    });
  });
}
```

Elaborado por: Diego Santamaria

Figura 96. Método Guardar usuarios

```
saveUser(usuario: Usuario, uid: string) {
  return new Promise((resolve) => {
    this.usuariosColeccion
      .doc(uid)
      .set(usuario)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message;
      });
  });
}
```

Elaborado por: Diego Santamaria

Inicio de sesión de usuario previamente registrado en el sistema

Figura 97. Método de inicio de sesión

```
signIn(email: string, password: string) {
  return new Promise((resolve, reject) => {
    this.authService.signInWithEmailAndPassword(email, password).then(
      (datos) => resolve(datos),
      (error) =>
        reject(this.errorCode.firebaseErrors[error.code] || error.message)
    );
  });
}
```

Elaborado por: Diego Santamaria

Restablecer contraseña de usuario mediante su email

Figura 98. Método de reestablecer contraseña

```
resetPassword(email: string) {
  return new Promise((resolve, reject) => {
    this.authService.sendPasswordResetEmail(email).then(
      (datos) => resolve(datos),
      (error) =>
        reject(this.errorCode.firebaseErrors[error.code] || error.message)
    );
  });
}
```

Elaborado por: Diego Santamaria

Obtener todos los datos del usuario que ha iniciado sesión exitosamente

Figura 99. Método de obtener la autenticación de inicio de sesión de un usuario

```
getAuth() {
  return this.authService.authState.pipe(map((auth) => auth));
}
```

Elaborado por: Diego Santamaria

Servicio usuario

Obtener todos los usuarios registrados en el sistema web

Figura 100. Método de obtener usuarios del sistema

```
getUsers(): Observable<Usuario[]> {
  this.usuarios = this.usuariosColeccion.snapshotChanges().pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as Usuario;
        datos.uid = accion.payload.doc.id;
        return datos;
      });
    });
  );
  return this.usuarios;
}
```

Elaborado por: Diego Santamaria

Obtener un usuario del sistema mediante su ID

Figura 101. Método para obtener un usuario por su ID

```
getUserById(id: string) {
  this.usuarioDoc = this.db.doc<Usuario>(`usuarios/${id}`);
  return (this.usuario = this.usuarioDoc.snapshotChanges().pipe(
    map((accion) => {
      if (accion.payload.exists === false) {
        return null;
      } else {
        const datos = accion.payload.data() as Usuario;
        return datos as any;
      }
    })
  ));
}
```

Elaborado por: Diego Santamaria

Actualizar la información de un usuario del sistema.

Figura 102. Método para actualizar un usuario

```
updateUser(uid: string, usuario: Usuario) {
  return new Promise((resolve) => {
    delete usuario.uid;
    this.usuarioDoc = this.db.doc(`usuarios/${uid}`);
    this.usuarioDoc.update(usuario).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Obtener los usuarios inactivos del sistema.

Figura 103. Método para obtener los usuarios deshabilitados del sistema

```
getDisabledUsers(): Observable<Usuario[]> {
  this.usuarios = this.db
    .collection<Usuario>('usuarios', (ref) =>
      ref.where('activo', '=', false)
    )
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Usuario;
          datos.uid = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.usuarios;
}
```

Elaborado por: Diego Santamaria

Obtener los usuarios activos del sistema.

Figura 104. Método para obtener los usuarios habilitados en el sistema

```
getEnabledUsers(): Observable<Usuario[]> {
  this.usuarios = this.db
    .collection<Usuario>('usuarios', (ref) => ref.where('activo', '==', true))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Usuario;
          datos.uid = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.usuarios;
}
```

Elaborado por: Diego Santamaria

Guardar Token para las notificaciones de cada usuario.

Figura 105. Método para guardar token de usuario

```
saveToken(usuario: Usuario) {
  return new Promise<(resolve) => {
    this.usuarioDoc = this.db.doc(`usuarios/${usuario.uid}`);
    delete usuario.uid;
    this.usuarioDoc.update(usuario).then(resolve).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message
    });
  });
}
```

Elaborado por: Diego Santamaria

Servicio configuración

Obtener configuración del tipo de perfil de usuario

Figura 106. Método para obtener la configuración de tipo de perfil de usuario

```
getSettingProfile(): Observable<ConfiguracionPerfil> {
  this.configuracionPerfilDoc = this.db.doc<ConfiguracionPerfil>(
    `configuracion/${this.idPerfil}`
  );
  this.configuracionPerfil = this.configuracionPerfilDoc.valueChanges() as any;
  return this.configuracionPerfil;
}
```

Elaborado por: Diego Santamaria

Actualizar configuración del tipo de perfil de usuario

Figura 107. Método para cambiar la configuración del tipo de perfil de usuario

```
changeSettingProfile(configuracion: ConfiguracionPerfil) {
  return new Promise((resolve) => {
    this.configuracionPerfilDoc = this.db.doc<ConfiguracionPerfil>(
      `configuracion/${this.idPerfil}`
    );
    this.configuracionPerfilDoc.update(configuracion).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Obtener configuración de permitir registros de usuarios

Figura 108. Método para obtener la configuración de permitir registro de usuarios

```
getSettingAllowRegistration(): Observable<ConfiguracionPermitirRegistro> {
  this.configuracionPermitirRegistroDoc = this.db.doc<
    ConfiguracionPermitirRegistro
  >(`configuracion/${this.idPermitirRegistro}`);
  this.configuracionPermitirRegistro = this.configuracionPermitirRegistroDoc.valueChanges() as any;
  return this.configuracionPermitirRegistro;
}
```

Elaborado por: Diego Santamaria

Actualizar configuración de permitir registros de usuarios

Figura 109. Método para cambiar la configuración del permiso de registro de usuarios del sistema

```
changeSettingAllowRegistration(configuracion: ConfiguracionPermitirRegistro) {
  return new Promise((resolve) => {
    this.configuracionPermitirRegistroDoc = this.db.doc<
      ConfiguracionPermitirRegistro
    >(`configuracion/${this.idPermitirRegistro}`);
    this.configuracionPermitirRegistroDoc
      .update(configuracion)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message;
      });
  });
}
```

Elaborado por: Diego Santamaria

Obtener el número de clientes actuales en el sistema

Figura 110. Método para obtener la configuración de número de clientes del sistema

```
getSettingLengthClientes(): Observable<ConfiguracionTamanoClientes> {
  this.configuracionTamanoClientesDoc = this.db.doc<ConfiguracionTamanoClientes>(
    `configuracion/${this.idTamanoArrayClientes}`
  );
  this.configuracionTamanoClientes = this.configuracionTamanoClientesDoc.valueChanges() as any;
  return this.configuracionTamanoClientes;
}
```

Elaborado por: Diego Santamaria

Actualizar el número de clientes actuales en el sistema

Figura 111. Método para cambiar la configuración del número de clientes del sistema

```
changeSettingLengthClientes(configuracion: ConfiguracionTamanoClientes) {
  return new Promise((resolve) => {
    this.configuracionTamanoClientesDoc = this.db.doc<
      ConfiguracionTamanoClientes
    >(`configuracion/${this.idTamanoArrayClientes}`);
    this.configuracionTamanoClientesDoc
      .update(configuracion).then(resolve)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message;
      });
  });
}
```

Elaborado por: Diego Santamaria

Obtener el token de la configuración del sistema.

Figura 112. Método para obtener la configuración de token de actualización de clientes

```
getTokenUpdateClientes(): Observable<ConfiguracionTamanoClientes> {
  this.configuracionTamanoClientesDoc = this.db.doc<ConfiguracionTamanoClientes>(
    `configuracion/${this.idTamanoArrayClientes}`
  );
  this.configuracionTamanoClientes = this.configuracionTamanoClientesDoc.valueChanges() as any;
  return this.configuracionTamanoClientes;
}
```

Elaborado por: Diego Santamaria

Actualizar el token de la configuración del sistema el cual permitirá verificar si la lista de clientes se ha actualizado.

Figura 113. Método para cambiar la configuración del token de actualización de clientes

```
setTokenUpdateClientes(tokenUpdate: ConfiguracionTamanoClientes){
  return new Promise((resolve) => {
    this.configuracionTamanoClientesDoc= this.db.doc(`configuracion/${this.idTamanoArrayClientes}`);
    this.configuracionTamanoClientesDoc.update(tokenUpdate)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Obtener el número de pedido actual del sistema

Figura 114. Método para obtener la configuración de numero de pedidos del sistema

```
getNumeroPedido(): Observable<ConfiguracionNumeroPedido> {
  this.configuracionNumeroPedidoDoc = this.db.doc<ConfiguracionNumeroPedido>(
    `configuracion/${this.idNumeroPedido}`
  );
  this.configuracionNumeroPedido = this.configuracionNumeroPedidoDoc.valueChanges() as any;
  return this.configuracionNumeroPedido;
}
```

Elaborado por: Diego Santamaria

Actualizar el número de pedido actual del sistema

Figura 115. Método para cambiar la configuración del número de pedidos del sistema

```
updateNumeroPedido(configuracion: ConfiguracionNumeroPedido) {
  return new Promise((resolve) => {
    this.configuracionNumeroPedidoDoc = this.db.doc<ConfiguracionNumeroPedido>(
      `configuracion/${this.idNumeroPedido}`
    );
    this.configuracionNumeroPedidoDoc.update(configuracion).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
    resolve(true);
  });
}
```

Elaborado por: Diego Santamaria

Obtener el número de visita actual del sistema

Figura 116. Método para obtener la configuración de número de visitas del sistema

```
getNumeroVisita(): Observable<ConfiguracionNumeroVisita> {
  this.configuracionNumeroVisitaDoc = this.db.doc<ConfiguracionNumeroVisita>(
    `configuracion/${this.idNumeroVisita}`
  );
  this.configuracionNumeroVisita = this.configuracionNumeroVisitaDoc.valueChanges() as any;
  return this.configuracionNumeroVisita;
}
```

Elaborado por: Diego Santamaria

Actualizar el número de visita actual del sistema

Figura 117. Método para cambiar la configuración del número de visitas del sistema

```
updateNumeroVisita(configuracion: ConfiguracionNumeroVisita) {
  return new Promise((resolve) => {
    this.configuracionNumeroVisitaDoc = this.db.doc<ConfiguracionNumeroVisita>(
      `configuracion/${this.idNumeroVisita}`
    );
    this.configuracionNumeroVisitaDoc.update(configuracion).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
    resolve(true);
  });
}
```

Elaborado por: Diego Santamaria

Obtener el número de cobranza actual del sistema

Figura 118. Método para obtener la configuración de numero de cobranzas del sistema

```
getNumeroCobranza(): Observable<ConfiguracionNumeroCobranza> {
  this.configuracionNumeroCobranzaDoc = this.db.doc<ConfiguracionNumeroCobranza>(
    `configuracion/${this.idNumeroCobranza}`
  );
  this.configuracionNumeroCobranza = this.configuracionNumeroCobranzaDoc.valueChanges() as any;
  return this.configuracionNumeroCobranza;
}
```

Elaborado por: Diego Santamaria

Actualizar el número de cobranza actual del sistema

Figura 119. Método para cambiar la configuración del número de cobranza del sistema

```
updateNumeroCobranza(configuracion: ConfiguracionNumeroCobranza) {
  return new Promise((resolve) => {
    this.configuracionNumeroCobranzaDoc = this.db.doc<ConfiguracionNumeroCobranza>(
      `configuracion/${this.idNumeroCobranza}`
    );
    this.configuracionNumeroCobranzaDoc.update(configuracion).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
    resolve(true);
  });
}
```

Elaborado por: Diego Santamaria

Obtener el número de kilometraje actual del sistema

Figura 120. Método para obtener la configuración de numero de kilometrajes del sistema

```
getNumeroKilometraje(): Observable<ConfiguracionNumeroKilometraje> {
  this.configuracionNumeroKilometrajeDoc = this.db.doc<ConfiguracionNumeroKilometraje>(
    `configuracion/${this.idNumeroKilometraje}`
  );
  this.configuracionNumeroKilometraje = this.configuracionNumeroKilometrajeDoc.valueChanges() as any;
  return this.configuracionNumeroKilometraje;
}
```

Elaborado por: Diego Santamaria

Actualizar el número de kilometraje actual del sistema

Figura 121. Método para cambiar la configuración del número de kilometraje del sistema

```
updateNumeroKilometraje(configuracion: ConfiguracionNumeroKilometraje) {
  return new Promise((resolve) => {
    this.configuracionNumeroKilometrajeDoc = this.db.doc<ConfiguracionNumeroKilometraje>(
      `configuracion/${this.idNumeroKilometraje}`
    );
    this.configuracionNumeroKilometrajeDoc.update(configuracion).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
    resolve(true);
  });
}
```

Elaborado por: Diego Santamaria

Servicio categoría

Obtener categoría

Figura 122. Método obtener categoría

```
getNameCategories(): Observable<Categoria[]> {
  this.categorias = this.categoriaColeccion.snapshotChanges().pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as Categoria;
        datos.id = accion.payload.doc.id;
        return datos;
      });
    });
  return this.categorias;
}
```

Elaborado por: Diego Santamaria

Guardar categoría

Figura 123. Método guardar categoría

```
saveNameCategorie(nombre: Categoria) {
  return new Promise((resolve) => {
    this.categoriaColeccion.add(nombre)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Actualizar categoría

Figura 124. Método actualizar categoría

```
updateCategory(categoria: Categoria) {
  return new Promise((resolve) => {
    this.categoriaDoc = this.db.doc(`categorias/${categoria.id}`);
    delete categoria.id;
    this.categoriaDoc.update(categoria)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Eliminar categoría

Figura 125. Método eliminar categoría

```
deleteCategory(nombreProducto: Categoria) {
  return new Promise((resolve) => {
    this.categoriaDoc = this.db.doc(`categorias/${nombreProducto.id}`);
    this.categoriaDoc.delete()
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Servicio nombre producto

Obtener nombre producto

Figura 126. Método obtener nombre producto

```
getNameProducts(): Observable<nombreProducto[]> {
  this.nombresProductos = this.nombreProductoColeccion.snapshotChanges().pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as nombreProducto;
        datos.id = accion.payload.doc.id;
        return datos;
      });
    });
  );
  return this.nombresProductos;
}
```

Elaborado por: Diego Santamaria

Guardar nombre producto

Figura 127. Método guardar nombre producto

```
saveNameProduct(nombre: nombreProducto) {
  return new Promise((resolve) => {
    this.nombreProductoColeccion.add(nombre).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Actualizar nombre producto

Figura 128. Método actualizar nombre producto

```
updateNameProduct(nombreProducto: nombreProducto) {
  return new Promise((resolve) => {
    this.nombreProductoDoc = this.db.doc(
      `nombresProductos/${nombreProducto.id}`
    );
    delete nombreProducto.id;
    this.nombreProductoDoc.update(nombreProducto).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Eliminar nombre producto

Figura 129. Método eliminar nombre producto

```
deleteNameproduct(nombreProducto: nombreProducto) {
  return new Promise((resolve) => {
    this.nombreProductoDoc = this.db.doc(
      `nombresProductos/${nombreProducto.id}`
    );
    this.nombreProductoDoc.delete().catch((error) => {
      this.errorCode.firebaseioErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Servicio empaque

Obtener empaque

Figura 130. Método obtener empaque

```
getEmpaques(): Observable<Empaque[]> {
  this.empaques = this.empaqueColeccion.snapshotChanges().pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as Empaque;
        datos.id = accion.payload.doc.id;
        return datos;
      });
    })
  );
  return this.empaques;
}
```

Elaborado por: Diego Santamaria

Guardar empaque

Figura 131. Método guardar empaque

```
saveEmpaque(nombre: Empaque) {
  return new Promise((resolve) => {
    this.empaqueColeccion.add(nombre)
      .catch((error) => {
        this.errorCode.firebaseioErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Actualizar empaque

Figura 132. Método actualizar empaque

```
updateEmpaque(empaque: Empaque) {
  return new Promise((resolve) => {
    this.empaqueDoc = this.db.doc(`empaques/${empaque.id}`);
    delete empaque.id;
    this.empaqueDoc.update(empaque)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Eliminar empaque

Figura 133. Método eliminar empaque

```
deleteEmpaque(empaque: Empaque) {
  return new Promise((resolve) => {
    this.empaqueDoc = this.db.doc(`empaques/${empaque.id}`);
    this.empaqueDoc.delete()
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Servicio producto

Obtener producto

Figura 134. Método obtener producto

```
getProductos(): Observable<Producto[]> {
  this.productos = this.productoColeccion.snapshotChanges().pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as Producto;
        datos.id = accion.payload.doc.id;
        return datos;
      });
    });
  );
  return this.productos;
}
```

Elaborado por: Diego Santamaria

Obtener producto por categoría

Figura 135. Método obtener producto por categoría

```
getProductsByCategory(categoria: string): Observable<Producto[]> {
  this.productos = this.db
    .collection<Producto>('nombresProductos', (ref) => ref.where('categoria', '=', categoria))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Producto;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.productos;
}
```

Elaborado por: Diego Santamaria

Guardar producto

Figura 136. Método guardar producto

```
saveProduct(producto: Producto) {
  return new Promise((resolve) => {
    this.productoColeccion.add(producto).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Actualizar producto

Figura 137. Método actualizar producto

```
updateProduct(producto: Producto) {
  return new Promise((resolve) => {
    this.productoDoc = this.db.doc(
      `productos/${producto.id}`
    );
    delete producto.id;
    this.productoDoc.update(producto).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Eliminar producto

Figura 138. Método eliminar producto

```
deleteProduct(producto: Producto) {
  return new Promise((resolve) => {
    this.productoDoc = this.db.doc(
      `productos/${producto.id}`
    );
    this.productoDoc.delete().catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Servicio kilometraje

Obtener kilometraje por el ID del usuario vendedor

Figura 139. Método obtener kilometraje por ID

```
getKilometrajeById(id: string) {
  this.kilometrajeDoc = this.db.doc<Kilometraje>(`kilometrajes/${id}`);
  return (this.kilometraje = this.kilometrajeDoc.snapshotChanges().pipe(
    map((accion) => {
      if (accion.payload.exists) {
        const datos = accion.payload.data() as Kilometraje;
        return datos as any;
      } else {
        return null;
      }
    })
  ));
}
```

Elaborado por: Diego Santamaria

Obtener kilometraje por un rango de fecha y el ID del usuario vendedor

Figura 140. Método obtener kilometraje por fecha y vendedor

```
getKilometrajeByDateAndVendedor([firstDate: string, secondDate: string]): Observable<Kilometraje[]>{
  this.kilometrajes = this.db
    .collection<Kilometraje>('kilometrajes', (ref) => ref.orderBy("fecha").startAt(firstDate).endAt(secondDate))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Kilometraje;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.kilometrajes;
}
```

Elaborado por: Diego Santamaria

Guardar kilometraje

Figura 141. Método guardar kilometraje

```
saveKilometraje(kilometraje: Kilometraje) {
  return new Promise((resolve, reject)=>{
    this.kilometrajeColeccion.add(kilometraje).then((resp)=>{
      resolve(resp.id)
    }).catch((error)=>{
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Actualizar kilometraje

Figura 142. Método actualizar kilometraje

```
updateKilometraje(kilometraje: Kilometraje) {
  return new Promise((resolve, reject)=>{
    this.kilometrajeDoc = this.db.doc(
      `kilometrajes/${kilometraje.id}`
    );
    delete kilometraje.id;
    this.kilometrajeDoc.update(kilometraje).then(resolve).catch((error)=>{
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Servicio vendedores activos

Obtener el usuario activo que haya ingresado su kilometraje.

Figura 143. Método obtener vendedores activos

```
getCurrentIdUserKilometraje(): Observable<currentUserKilometraje[]> {
  this.vendedoresKilometrajes = this.vendedorKilometrajeColeccion.snapshotChanges().pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as currentUserKilometraje;
        datos.id = accion.payload.doc.id;
        return datos;
      });
    });
  );
  return this.vendedoresKilometrajes;
}
```

Elaborado por: Diego Santamaria

Guardar el id del usuario activo

Figura 144. Método guardar ID de vendedores activos

```
savecurrentVendedorID(idUser: currentUserKilomeraje) {  
  return new Promise((resolve, reject)=>{  
    this.vendedorKilometrajeColeccion.add(idUser).then(resolve).catch((error)=>{  
      this.errorCode.firebaseErrors[error.code] || error.message;  
    });  
  });  
}
```

Elaborado por: Diego Santamaria

Eliminar el usuario activo

Figura 145. Método eliminar vendedores activos

```
deleteCurrentVendedorID(IDcurrentUserKilometraje: string) {  
  return new Promise((resolve, reject)=>{  
    this.vendedorKilometrajeDoc = this.db.doc(  
      `vendedoresActivos/${IDcurrentUserKilometraje}`  
    );  
    this.vendedorKilometrajeDoc.delete().then(resolve).catch((error)=>{  
      this.errorCode.firebaseErrors[error.code] || error.message;  
    });  
  });  
}
```

Elaborado por: Diego Santamaria

Servicio clientes

Obtener clientes

Figura 146. Método obtener cliente

```
getClientes(): Observable<Cliente[]> {  
  this.clientes = this.clienteColeccion.snapshotChanges().pipe(  
    map((cambios) => {  
      return cambios.map((accion) => {  
        const datos = accion.payload.doc.data() as Cliente;  
        datos.id = accion.payload.doc.id;  
        return datos;  
      });  
    });  
  return this.clientes;  
}
```

Elaborado por: Diego Santamaria

Guardar cliente

Figura 147. Método guardar cliente

```
saveCliente(cliente: Cliente) {  
  return new Promise((resolve, reject) => {  
    this.clienteColeccion  
      .add(cliente)  
      .then(resolve)  
      .catch((error) => {  
        this.errorCode.firebaseErrors[error.code] || error.message;  
      });  
  });  
}
```

Elaborado por: Diego Santamaria

Actualizar cliente

Figura 148. Método actualizar cliente

```
updateCliente(cliente: Cliente) {
  return new Promise((resolve) => {
    this.clienteDoc = this.db.doc(`clientes/${cliente.id}`);
    delete cliente.id;
    this.clienteDoc
      .update(cliente)
      .then(resolve)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message;
      });
  });
}
```

Elaborado por: Diego Santamaria

Eliminar cliente

Figura 149. Método eliminar cliente

```
deleteCliente(cliente: Cliente) {
  return new Promise((resolve) => {
    this.clienteDoc = this.db.doc(`clientes/${cliente.id}`);
    this.clienteDoc.delete().then(resolve)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message;
      });
  });
}
```

Elaborado por: Diego Santamaria

Servicio pedidos

Obtener pedidos que aún no han sido despachados.

Figura 150. Método obtener pedido

```
getPedidoPorDespachar(): Observable<Pedido[]> {
  this.pedidos = this.db
    .collection<Pedido>('pedidos', (ref) => ref.where('despachado', '!=', false))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Pedido;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.pedidos;
}
```

Elaborado por: Diego Santamaria

Obtener pedido por su ID

Figura 151. Método obtener pedido por su ID

```
getPedidoById(id: string) {
  this.pedidoDoc = this.db.doc<Pedido>(`pedidos/${id}`);
  return (this.pedido = this.pedidoDoc.snapshotChanges().pipe(
    map((accion) => {
      if (accion.payload.exists) {
        const datos = accion.payload.data() as Pedido;
        datos.id = accion.payload.id;
        return datos as any;
      }
    })
  ));
}
```

Elaborado por: Diego Santamaria

Obtener pedidos por su cliente

Figura 152. Método obtener pedido por su cliente

```
getPedidoByCliente(rucCliente: string): Observable<Pedido[]>{
  this.pedidos = this.db
    .collection<Pedido>('pedidos', (ref) => ref.where('rucCliente', '=', rucCliente).where('despachado', '=', true))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Pedido;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.pedidos;
}
```

Elaborado por: Diego Santamaria

Obtener pedido por el número de pedido

Figura 153. Método obtener pedido por su numero

```
getPedidoBynumeroPedido(numeroPedido: number): Observable<Pedido[]>{
  this.pedidos = this.db
    .collection<Pedido>('pedidos', (ref) => ref.where('numeroPedido', '=', numeroPedido))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Pedido;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.pedidos;
}
```

Elaborado por: Diego Santamaria

Obtener pedido por una fecha específica y por el cliente.

Figura 154. Método obtener pedido por una fecha y cliente

```
getPedidoByDateAndClient(firstDate: string, secondDate: string): Observable<Pedido[]>{
  this.pedidos = this.db
  .collection<Pedido>('pedidos', (ref) => ref.orderBy("fecha").startAt(firstDate).endAt(secondDate))
  .snapshotChanges()
  .pipe(
    map((cambios) => {
      return cambios.map((accion) => {
        const datos = accion.payload.doc.data() as Pedido;
        datos.id = accion.payload.doc.id;
        return datos;
      });
    });
  );
  return this.pedidos;
}
```

Elaborado por: Diego Santamaria

Guardar pedido

Figura 155. Método guardar pedido

```
savePedido(pedido: Pedido) {
  return new Promise((resolve) => {
    this.pedidoColeccion.add(pedido).then(resolve).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Actualizar pedido

Figura 156. Método actualizar pedido

```
updatePedido(pedido: Pedido) {
  return new Promise((resolve) => {
    this.pedidoDoc = this.db.doc(`pedidos/${pedido.id}`);
    delete pedido.id;
    this.pedidoDoc.update(pedido).then(resolve)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message;
      });
  });
}
```

Elaborado por: Diego Santamaria

Eliminar pedido

Figura 157. Método eliminar pedido

```
deletePedido(pedido: Pedido) {
  return new Promise((resolve) => {
    this.pedidoDoc = this.db.doc(
      `pedidos/${pedido.id}`
    );
    this.pedidoDoc.delete().then(resolve).catch((error) => {
      this.errorCode.firebaseErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Cambiar estado de pedido como despachado

Figura 158. Método cambiar estado de pedido

```
markPedidoAsDispatched(pedido: Pedido){
  return new Promise((resolve) => {
    this.pedidoDoc = this.db.doc(`pedidos/${pedido.id}`);
    delete pedido.id;
    this.pedidoDoc.update(pedido).then(resolve)
      .catch((error) => {
        this.errorCode.firebaseErrors[error.code] || error.message
      });
  });
}
```

Elaborado por: Diego Santamaria

Servicio descripción pedidos

Obtener la descripción del pedido creado

Figura 159. Método obtener descripción pedidos

```
getDescripcionPedidoById(id: string) {
  this.descripcionPedidoDoc = this.db.doc<DescripcionPedido>(`descripcionPedidos/${id}`);
  return (this.descripcionPedido = this.descripcionPedidoDoc.snapshotChanges().pipe(
    map((accion) => {
      if (accion.payload.exists) {
        const datos = accion.payload.data() as DescripcionPedido;
        return datos as any;
      }else{
        return null;
      }
    })
  ));
}
```

Elaborado por: Diego Santamaria

Guardar la descripción del pedido creado

Figura 160. Método guardar descripción pedidos

```
saveDescripcionPedido(jsonDescripcionPedido: any) {  
  return new Promise((resolve) => {  
    this.descripcionPedidoColeccion.add(jsonDescripcionPedido).then((resp)=>{  
      resolve(resp.id);  
    }).catch((error) => {  
      this.errorCode.firebaseErrors[error.code] || error.message;  
    });  
  });  
}
```

Elaborado por: Diego Santamaria

Actualizar la descripción de un pedido creado

Figura 161. Método actualizar descripción pedidos

```
updateDescripcionPedido(id: string, jsonDescripcionPedido: any) {  
  return new Promise((resolve) => {  
    this.descripcionPedidoDoc = this.db.doc(`descripcionPedidos/${id}`);  
    this.descripcionPedidoDoc.update(jsonDescripcionPedido).catch((error) => {  
      this.errorCode.firebaseErrors[error.code] || error.message  
    });  
  });  
}
```

Elaborado por: Diego Santamaria

Eliminar la descripción de un pedido eliminado

Figura 162. Método eliminar descripción pedidos

```
deleteDescripcionPedido(id: string) {  
  return new Promise((resolve) => {  
    this.descripcionPedidoDoc = this.db.doc(  
      `descripcionPedidos/${id}`  
    );  
    this.descripcionPedidoDoc.delete().then(resolve).catch((error) => {  
      this.errorCode.firebaseErrors[error.code] || error.message;  
    });  
  });  
}
```

Elaborado por: Diego Santamaria

Servicio visitas

Obtener visita por su ID

Figura 163. Método obtener visita por ID

```
getVisitaById(id: string) {
  this.visitaDoc = this.db.doc<Visita>(`visitas/${id}`);
  return (this.visita = this.visitaDoc.snapshotChanges().pipe(
    map((accion) => {
      if (accion.payload.exists) {
        const datos = accion.payload.data() as Visita;
        datos.id = accion.payload.id;
        return datos as any;
      }
    })
  ));
}
```

Elaborado por: Diego Santamaria

Obtener visita por el número de visita

Figura 164. Método obtener visita por su número

```
getVisitaByNumeroVisita(numeroVisita: number): Observable<Visita[]>{
  this.visitas = this.db
    .collection<Visita>('visitas', (ref) => ref.where('numeroVisita', '==', numeroVisita))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Visita;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      });
    );
  return this.visitas;
}
```

Elaborado por: Diego Santamaria

Obtener visita por una fecha y el cliente

Figura 165. Método obtener visita por fecha y cliente

```
getVisitaByDateAndClient(firstDate: string, secondDate: string): Observable<Visita[]>{
  this.visitas = this.db
    .collection<Visita>('visitas', (ref) => ref.orderBy("fechaCreacion").startAt(firstDate).endAt(secondDate))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Visita;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      });
    );
  return this.visitas;
}
```

Elaborado por: Diego Santamaria

Guardar visita

Figura 166. Método guardar visita

```
saveVisita(visita: Visita) {
  return new Promise((resolve) => {
    this.visitaColeccion.add(visita).then(resolve).catch((error) => {
      this.errorCode.firebaseioErrors[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Servicio cobranzas

Obtener cobranza por su ID

Figura 167. Método obtener cobranza

```
getCobranzaById(id: string) {
  this.cobranzaDoc = this.db.doc<Cobranza>(`cobranzas/${id}`);
  return (this.cobranza = this.cobranzaDoc.snapshotChanges()).pipe(
    map((accion) => {
      if (accion.payload.exists) {
        const datos = accion.payload.data() as Cobranza;
        datos.id = accion.payload.id;
        return datos as any;
      }
    })
  );
}
```

Elaborado por: Diego Santamaria

Obtener cobranza por el número de cobranza

Figura 168. Método obtener cobranza por su número

```
getCobranzasByNumeroCobranza(numerovisita: number): Observable<Cobranza[]>{
  this.cobranzas = this.db
    .collection<Cobranza>('cobranzas', (ref) => ref.where('numeroCobranza', '=', numerovisita))
    .snapshotChanges()
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Cobranza;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      })
    );
  return this.cobranzas;
}
```

Elaborado por: Diego Santamaria

Obtener cobranza por una fecha y el cliente

Figura 169. Método obtener cobranza por fecha y cliente

```
getCobranzaByDateAndClient(firstDate: string, secondDate: string): Observable<Cobranza[]>{
  this.cobranzas = this.db
  .collection<Cobranza>('cobranzas', (ref) => ref.orderBy("fechaCreacion").startAt(firstDate).endAt(secondDate))
  .snapshotChanges()
  .pipe(
    .pipe(
      map((cambios) => {
        return cambios.map((accion) => {
          const datos = accion.payload.doc.data() as Cobranza;
          datos.id = accion.payload.doc.id;
          return datos;
        });
      });
    );
  );
  return this.cobranzas;
}
```

Elaborado por: Diego Santamaria

Guardar cobranza

Figura 170. Método guardar cobranza

```
saveCobranza(cobranza: Cobranza) {
  return new Promise((resolve) => {
    this.cobranzaColeccion.add(cobranza).then(resolve).catch((error) => {
      this.errorCode.firebaseio[error.code] || error.message;
    });
  });
}
```

Elaborado por: Diego Santamaria

Servicio geolocalización

Obtener latitud y longitud actual de un vendedor.

Figura 171. Método obtener coordenadas geográficas

```
getCurrentLatLonVendedor() {
  return new Promise((resolve, reject) => {
    navigator.geolocation.getCurrentPosition(
      function (position: any) {
        resolve(position);
      }, (error) => {
        resolve(error);
      },
      {
        maximumAge: 60000,
        timeout: 25000,
        enableHighAccuracy: true
      }
    );
  });
}
```

Elaborado por: Diego Santamaria

Guardianes de rutas

Los guardianes de rutas permitirán validar el inicio de sesión de cada uno de los usuarios para que puedan utilizar el sistema visualizando los diferentes módulos que conforman el sistema web.

Figura 172. Guardianes de rutas de componentes

```
canActivate(): Observable<boolean> {
  return this.afAuth.authState.pipe(
    map((auth) => {
      if (!auth) {
        this.router.navigate(['/login']);
        return false;
      } else {
        return true;
      }
    })
  );
}
```

Elaborado por: Diego Santamaria

```
const routes: Routes = [
  {path: '', component: TableroComponent, canActivate:[AuthGuard]},
  {path: 'login', component: LoginComponent},
  {path: 'registrarse', component: RegistroComponent, canActivate:[ConfiguracionGuard]},
  {path: 'configuracion', component: ConfiguracionComponent, canActivate:[AuthGuard,AuthAdminGuard]},
  {path: 'perfil', component: PerfilUserComponent, canActivate:[AuthGuard]},
  {path: 'nombres-productos', component: NombresProductosComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: 'productos', component: ProductosComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: 'empaques', component: EmpaquesComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: 'categorias', component: CategoriasComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: 'reportes-admin-todos', component: UsuariosTodosComponent, canActivate:[AuthGuard, AuthAdminGuard]},
  {path: 'reportes-admin-habilitados', component: UsuariosHabilitadosComponent, canActivate:[AuthGuard, AuthAdminGuard]},
  {path: 'reportes-admin-deshabilitados', component: UsuariosDeshabilitadosComponent, canActivate:[AuthGuard, AuthAdminGU]},
  {path: 'clientes', component: ClientesComponent, canActivate:[AuthGuard, AuthVendedorGuard]},
  {path: 'ver-pedido/:id', component: VerPedidoComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: 'buscar-pedido', component: BuscarPedidoComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: 'pedidos', component: PedidoComponent, canActivate:[AuthGuard, AuthVendedorGuard]},
  {path: 'pedidos/crear-pedido', component: CrearPedidoComponent, canActivate:[AuthGuard, AuthVendedorGuard]},
  {path: 'pedidos/editar/:id', component: EditarPedidoComponent, canActivate:[AuthGuard, AuthVendedorGuard]},
  {path: 'visitas', component: VisitaComponent, canActivate:[AuthGuard, AuthVendedorGuard]},
  {path: 'cobranzas', component: CobranzaComponent, canActivate:[AuthGuard, AuthVendedorGuard]},
  {path: 'pedidos-buscar', component: PedidosBuscarComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'visitas-buscar', component: VisitasBuscarComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'cobranzas-buscar', component: CobranzasBuscarComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'kilometrajes-buscar', component: KilometrajesBuscarComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'reporte-pedido/:id', component: ReportePedidoComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'reporte-visita/:id', component: ReporteVisitaComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'reporte-cobranza/:id', component: ReporteCobranzaComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'reporte-kilometraje/:id', component: ReporteKilometrajeComponent, canActivate:[AuthGuard, AuthGerenteGuard]},
  {path: 'notificacion-email', component: IngresoTokenComponent, canActivate:[AuthGuard, AuthBodegueroGuard]},
  {path: '**', component: NoEncontradoComponent}
];
```


Métodos para imprimir y descargar PDF

Figura 174. Método para imprimir documento

```
savePDF(DATA: any) {
  var promise = new Promise<void>((resolve, reject) => {
    if (screen.width < 1024) {
      document
        .querySelector('meta[name=viewport]')!
        .setAttribute('content', 'width=1900');
    }
    const doc = new jsPDF('p', 'pt', 'a4');
    const options = {
      background: 'white',
      scale: 3,
    };
  });
  html2canvas(DATA, options)
    .then((canvas) => {
      const img = canvas.toDataURL('image/PNG');
      const bufferX = 15;
      const bufferY = 15;
      const imgProps = (doc as any).getImageProperties(img);
      const pdfWidth = doc.internal.pageSize.getWidth() - 2 * bufferX;
      const pdfHeight = (imgProps.height * pdfWidth) / imgProps.width;
      doc.addImage(
        img,
        'PNG',
        bufferX,
        bufferY,
        pdfWidth,
        pdfHeight,
        undefined,
        'FAST'
      );
      return doc;
    })
    .then((docResult) => {
      document
        .querySelector('meta[name=viewport]')!
        .setAttribute('content', 'width=device-width, initial-scale=1');
      docResult.save(
        `${new Date().toJSON().slice(0, 10).replace(/-/g, '/')}.pdf`
      );
      resolve();
    });
});
return promise;
}
```

Elaborado por: Diego Santamaria

Figura 173. Método para imprimir documento

```
printPDF(DATA: any) {
  var promise = new Promise<void>((resolve, reject) => {
    if (screen.width < 1024) {
      document
        .querySelector('meta[name=viewport]')!
        .setAttribute('content', 'width=1900');
    }
    const doc = new jsPDF('p', 'pt', 'a4');
    const options = {
      scrollly: -window.scrollY,
      background: 'white',
      scale: 3,
    };
  });
  html2canvas(DATA, options)
    .then((canvas) => {
      const img = canvas.toDataURL('image/PNG');
      const bufferX = 15;
      const bufferY = 15;
      const imgProps = (doc as any).getImageProperties(img);
      const pdfWidth = doc.internal.pageSize.getWidth() - 2 * bufferX;
      const pdfHeight = (imgProps.height * pdfWidth) / imgProps.width;
      doc.addImage(
        img,
        'PNG',
        bufferX,
        bufferY,
        pdfWidth,
        pdfHeight,
        undefined,
        'FAST'
      );
      return doc;
    })
    .then((docResult) => {
      document
        .querySelector('meta[name=viewport]')!
        .setAttribute('content', 'width=device-width, initial-scale=1');
      docResult.output('dataurlnewwindow');
      resolve();
    });
});
return promise;
}
```

Elaborado por: Diego Santamaria

3.2.4 Fase IV: Pruebas

Para establecer y verificar que la Aplicación Web Progresiva funciona adecuadamente, se deben realizar pruebas de aceptación las cuales permitirán localizar errores que alteren el funcionamiento del sistema. Estas pruebas permiten examinar y comprobar que los objetivos de las historias de usuario cumplan con el alcance solicitado por el beneficiario.

3.2.4.1 Pruebas de aceptación

Tabla 115. Prueba de aceptación historia 002

Prueba de aceptación	
Número: 1	Historia de usuario: 002
Nombre: Ingreso al sistema web	
Evaluado por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: En este módulo se pide el correo electrónico y la contraseña del usuario previamente registrado para que pueda acceder al sistema web y utilizar las diferentes funciones dependiendo del rol asignado.	
Condiciones de ejecución: Abrir la aplicación web progresiva.	
Entrada: El usuario debe ingresar su correo electrónico y contraseña.	
Resultado esperado: El usuario ingresa al sistema web.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 116. Prueba de aceptación historia 003

Prueba de aceptación	
Número: 2	Historia de usuario: 003
Nombre: Restablecer contraseña	
Evaluado por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Los usuarios del sistema pueden restablecer su contraseña ingresando el correo electrónico que utiliza para ingresar al sistema web.	
Condiciones de ejecución: Abrir la aplicación web progresiva y seleccionar la opción de reestablecer contraseña.	
Entrada: El usuario debe ingresar su correo electrónico y enviar la solicitud para cambiar su contraseña.	
Resultado esperado: El usuario puede cambiar su contraseña.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 117. Prueba de aceptación historia 004

Prueba de aceptación	
Número: 3	Historia de usuario: 004
Nombre: Visualización de la pantalla de inicio	
Evaluated por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Pantalla en la cual se visualizan los módulos y utilidades que tendrá cada usuario del sistema, estos módulos se presentaran de acuerdo al tipo de perfil del usuario que ingrese al sistema web.	
Condiciones de ejecución: Abrir la aplicación web progresiva e iniciar sesión exitosamente.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario puede visualizar los módulos y utilidades necesarias para su trabajo.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 118. Prueba de aceptación historia 005

Prueba de aceptación	
Número: 4	Historia de usuario: 005
Nombre: Visualización del perfil de usuario	
Evaluated por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Se puede visualizar la información proporcionada por el usuario al momento de crear su cuenta.	
Condiciones de ejecución: Abrir la aplicación web progresiva y seleccionar en el menú, la opción de ver perfil.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario puede visualizar su información personal.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 119. Prueba de aceptación historia 006

Prueba de aceptación	
Número: 5	Historia de usuario: 006
Nombre: Edición del perfil de usuario	
Evaluated por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Modulo que permite editar la información básica del usuario.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar en el menú la opción de ver perfil. Dentro de ver perfil seleccionar la opción de editar perfil.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL. Los campos serán habilitados en dependencia a la información ingresada por el usuario.	
Resultado esperado: El usuario puede actualizar su información personal.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 120. Prueba de aceptación historia 007

Prueba de aceptación	
Número: 6	Historia de usuario: 007
Nombre: Visualización de usuarios del sistema	
Evaluated por: Ing. Adriana Sagñay	
Descripción: Se puede visualizar en la pantalla de inicio, todos los usuarios del sistema.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario administrador podrá visualizar los usuarios existentes en el sistema.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 121. Prueba de aceptación historia 008

Prueba de aceptación	
Número: 7	Historia de usuario: 008
Nombre: Funciones de bloqueo de usuario	
Evaluated por: Ing. Adriana Sagñay	
Descripción: Se puede bloquear o desbloquear usuarios del sistema que se encuentren de vacaciones o despedidos.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la cuenta del usuario que se va a inhabilitar o habilitar.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario administrador puede bloquear o desbloquear a los usuarios del sistema.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 122. Prueba de aceptación historia 009

Prueba de aceptación	
Número: 8	Historia de usuario: 009
Nombre: Configuración de registro de usuarios.	
Evaluated por: Ing. Adriana Sagñay	
Descripción: En este módulo se definirá el perfil que los usuarios tendrán al momento de registrarse en el sistema web. También se podrá restringir el registro de usuarios, dependiendo de las necesidades de la empresa.	
Condiciones de ejecución: Abrir la aplicación web progresiva y dirigirse a la opción de configuración de registro de usuario.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario administrador podrá activar o desactivar la opción de registro de nuevos usuarios, así como también definir el perfil de los nuevos usuarios.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 123. Prueba de aceptación historia 010

Prueba de aceptación	
Número: 9	Historia de usuario: 010
Nombre: Reportes	
Evaluado por: Ing. Adriana Sagñay	
Descripción: En este módulo se puede generar los reportes de todos los usuarios del sistema, usuarios activos y usuarios inactivos.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de reportes y elegir el tipo de reporte que se desea generar.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario administrador podrá generar reportes en base a los usuarios registrados en el sistema web.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 124. Prueba de aceptación historia 011

Prueba de aceptación	
Número: 10	Historia de usuario: 011
Nombre: Listado de categorías	
Evaluado por: Sr. Manuel Maji	
Descripción: Se muestran todas las categorías disponibles en el sistema	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar categorías.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero podrá ver todas las categorías.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 125. Prueba de aceptación historia 012

Prueba de aceptación	
Número: 11	Historia de usuario: 012
Nombre: Añadir categoría	
Evaluado por: Sr. Manuel Maji	
Descripción: Se podrá añadir las diferentes categorías que ofrece la empresa.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar categorías.	
Entrada: Los campos necesarios para añadir una categoría se habilitarán.	
Resultado esperado: El usuario bodeguero podrá añadir una categoría.	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 126. Prueba de aceptación historia 013

Prueba de aceptación	
Número: 12	Historia de usuario: 013
Nombre: Editar categoría	
Evaluado por: Sr. Manuel Maji	
Descripción: Se editará el nombre de las categorías que se han ingresado previamente.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar categorías.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL. Los campos para editar una categoría serán habilitados.	
Resultado esperado: El usuario bodeguero podrá editar una categoría.	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 127. Prueba de aceptación historia 014

Prueba de aceptación	
Número: 13	Historia de usuario: 014
Nombre: Eliminar categoría	
Evaluado por: Sr. Manuel Maji	
Descripción: Se eliminará la categoría elegida por el usuario.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar categorías.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero eliminará una categoría.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 128. Prueba de aceptación historia 015

Prueba de aceptación	
Número: 14	Historia de usuario: 015
Nombre: Listado de nombres productos	
Evaluado por: Sr. Manuel Maji	
Descripción: Se muestran los nombres de los productos existentes en el Sistema.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar nombre producto.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero podrá ver todos los nombres productos.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 129. Prueba de aceptación historia 016

Prueba de aceptación	
Número: 15	Historia de usuario: 016
Nombre: Añadir nombre producto	
Evaluated por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Se añadirán nombres de los productos que se encuentren en venta, para ello se tendrá que seleccionar una categoría existente.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar nombre producto.	
Entrada: Los campos necesarios para añadir un nombre producto se habilitarán.	
Resultado esperado: El usuario bodeguero podrá añadir un nombre producto.	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 130. Prueba de aceptación historia 017

Prueba de aceptación	
Número: 16	Historia de usuario: 017
Nombre: Editar nombre producto	
Evaluated por: Sr. Manuel Maji	
Descripción: Se puede editar los nombres de los productos que se encuentren en venta.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar nombre producto.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL. Los campos para editar un nombre producto serán habilitados.	
Resultado esperado: El usuario bodeguero podrá editar un nombre producto.	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 131. Prueba de aceptación historia 018

Prueba de aceptación	
Número: 17	Historia de usuario: 018
Nombre: Eliminar nombre producto	
Evaluated por: Sr. Manuel Maji	
Descripción: Se elimina el nombres del producto seleccionado por el usuario.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar nombre producto.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero eliminará un nombre producto.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 132. Prueba de aceptación historia 019

Prueba de aceptación	
Número: 18	Historia de usuario: 019
Nombre: Listado de empaques	
Evaluated por: Sr. Manuel Maji	
Descripción: Se muestran todos los empaques ingresados en el sistema.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar empaque.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero podrá ver todos los empaques.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 133. Prueba de aceptación historia 020

Prueba de aceptación	
Número: 19	Historia de usuario: 020
Nombre: Añadir empaque	
Evaluated por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Se añadirán los empaques utilizados para los productos que se encuentran disponibles.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar empaque.	
Entrada: Los campos necesarios para añadir un empaque se habilitarán.	
Resultado esperado: El usuario bodeguero podrá añadir un empaque.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 134. Prueba de aceptación historia 021

Prueba de aceptación	
Número: 20	Historia de usuario: 021
Nombre: Editar empaque	
Evaluated por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Se editarán los empaques existentes.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar empaque.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL. Los campos para editar un empaque serán habilitados.	
Resultado esperado: El usuario bodeguero podrá editar un empaque.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 135. Prueba de aceptación historia 022

Prueba de aceptación	
Número: 21	Historia de usuario: 022
Nombre: Eliminar empaque	
Evaluado por: Sr. José David Holguín, Ing. Adriana Sagñay, Sr. Manuel Maji	
Descripción: Se eliminarán los empaques seleccionados que se encuentren en el sistema	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de añadir y seleccionar empaque.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero eliminará un empaque.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 136. Prueba de aceptación historia 023

Prueba de aceptación	
Número: 22	Historia de usuario: 023
Nombre: Listar productos	
Evaluado por: Sr. Manuel Maji	
Descripción: Se muestran los productos creados en el sistema.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de agregar producto.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero podrá ver todos los productos.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 137. Prueba de aceptación historia 024

Prueba de aceptación	
Número: 23	Historia de usuario: 024
Nombre: Agregar producto	
Evaluated por: Sr. Manuel Maji	
Descripción: Se agregarán productos, seleccionando su categoría, nombre y empaque.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de agregar producto.	
Entrada: Los campos necesarios para añadir un producto se habilitarán.	
Resultado esperado: El usuario bodeguero podrá añadir un producto.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 138. Prueba de aceptación historia 025

Prueba de aceptación	
Número: 24	Historia de usuario: 025
Nombre: Editar producto	
Evaluated por: Sr. Manuel Maji	
Descripción: Se editarán productos que previamente hayan sido creados.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de agregar producto.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL. Los campos para editar un producto serán habilitados.	
Resultado esperado: El usuario bodeguero podrá editar un producto.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 139. Prueba de aceptación historia 026

Prueba de aceptación	
Número: 25	Historia de usuario: 026
Nombre: Eliminar producto	
Evaluated por: Sr. Manuel Maji	
Descripción: Se eliminarán productos seleccionados del sistema.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de agregar producto.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero eliminará un producto.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 140. Prueba de aceptación historia 027

Prueba de aceptación	
Número: 26	Historia de usuario: 027
Nombre: Ver pedidos realizados	
Evaluated por: Sr. Manuel Maji	
Descripción: Se observan las características que tiene el pedido generado por el usuario para poder despacharlo.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero puede ver los pedidos realizados por los vendedores.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 141. Prueba de aceptación historia 028

Prueba de aceptación	
Número: 27	Historia de usuario: 028
Nombre: Despachar pedido realizado	
Evaluated por: Sr. Manuel Maji	
Descripción: En la tabla que se muestran los pedidos ubicada en la pantalla de inicio se encontrara un botón el cual permitirá al usuario marcar ese pedido como despachado para que desaparezca del listado de pedidos pendientes.	
Condiciones de ejecución: Abrir la aplicación web progresiva, ingresar al pedido generado por un vendedor, marcar el pedido como despachado.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero puede despachar los pedidos pendientes.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 142. Prueba de aceptación historia 029

Prueba de aceptación	
Número: 28	Historia de usuario: 029
Nombre: Ver pedido marcados como despachados	
Evaluated por: Sr. Manuel Maji	
Descripción: Se podrá ver todos los pedidos que han sido realizados en el sistema por los vendedores.	
Condiciones de ejecución: Abrir la aplicación web progresiva, iniciar sesión correctamente. Una vez dentro del sistema, colocarse en la opción de buscar pedido.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario bodeguero puede ver los pedidos ya despachados.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 143. Prueba de aceptación historia 030

Prueba de aceptación	
Número: 29	Historia de usuario: 030
Nombre: Ingresar kilometraje	
Evaluated por: Ing. Adriana Sagñay	
Descripción: Al ingresar al sistema el vendedor deberá ingresar el kilometraje con el cual inicia sus actividades	
Condiciones de ejecución: Abrir la aplicación web progresiva, ingresar el kilometraje del vehículo.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario vendedor ingresara al sistema una vez ingresado su kilometraje.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 144. Prueba de aceptación historia 031

Prueba de aceptación	
Número: 30	Historia de usuario: 031
Nombre: Vista de crear pedidos	
Evaluated por: Ing. Adriana Sagñay	
Descripción: Se podrán observar los pedidos generados por el vendedor que se encuentren pendientes, así como las diferentes opciones que puede utilizar el vendedor.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de pedidos.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario vendedor podrá visualizar los pedidos creados por él.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 145. Prueba de aceptación historia 032

Prueba de aceptación	
Número: 31	Historia de usuario: 032
Nombre: Vista de visita al cliente	
Evaluado por: Ing. Adriana Sagñay	
Descripción: Se podrán observar las visitas realizadas a un cliente, registradas por el vendedor.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de visitas.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario vendedor podrá visualizar las visitas creados por él.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 146. Prueba de aceptación historia 033

Prueba de aceptación	
Número: 32	Historia de usuario: 033
Nombre: Vista de cobranza	
Evaluado por: Ing. Adriana Sagñay	
Descripción: Se podrán observar los cobros realizados por el vendedor.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de cobranzas.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario vendedor podrá visualizar las cobranzas creados por él.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 147. Prueba de aceptación historia 034

Prueba de aceptación	
Número: 33	Historia de usuario: 034
Nombre: Crear pedido	
Evaluated por: Ing. Adriana Sagñay	
Descripción: El vendedor podrá crear pedidos llenando los campos solicitados.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de pedidos. Por último, seleccionar la opción de crear pedido.	
Entrada: Los campos necesarios para crear un pedido se habilitarán.	
Resultado esperado: El usuario vendedor podrá crear un pedido.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

Tabla 148. Prueba de aceptación historia 035

Prueba de aceptación	
Número: 34	Historia de usuario: 035
Nombre: Editar pedido	
Evaluated por: Ing. Adriana Sagñay	
Descripción: Se podrá editar los pedidos que aún no han sido despachados por el bodeguero.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de pedidos. Por último, seleccionar el pedido que se quiere editar.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL. Los campos para editar un pedido se habilitarán.	
Resultado esperado: El usuario vendedor podrá editar un pedido siempre y cuando no haya sido despachado por el bodeguero.	
Evaluación de la prueba: Prueba de aceptación admitida	

Tabla 149. Prueba de aceptación historia 036

Prueba de aceptación	
Número: 35	Historia de usuario: 036
Nombre: Eliminar pedido	
Evaluated por: Ing. Adriana Sagñay	
Descripción: Se podrá eliminar un pedido.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de pedidos. Por último, seleccionar el pedido que se quiere eliminar.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario vendedor podrá eliminar un pedido siempre y cuando no haya sido despachado por el bodeguero.	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 150. Prueba de aceptación historia 037

Prueba de aceptación	
Número: 36	Historia de usuario: 037
Nombre: Crear Visita	
Evaluated por: Ing. Adriana Sagñay	
Descripción: El vendedor podrá crear una visita, especificando los campos solicitados.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de visitas.	
Entrada: Los campos necesarios para crear una visita serán habilitados.	
Resultado esperado: El usuario vendedor podrá crear una visita.	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 151. Prueba de aceptación historia 038

Prueba de aceptación	
Número: 37	Historia de usuario: 038
Nombre: Crear cobranza	
Evaluated por: Ing. Adriana Sagñay	
Descripción: El vendedor podrá crear una cobranza, especificando los campos solicitados.	
Condiciones de ejecución: Abrir la aplicación web progresiva, seleccionar la opción de cobranzas.	
Entrada: Los campos necesarios para crear una cobranza serán habilitados.	
Resultado esperado:	
Evaluación de la prueba: Prueba de aceptación admitida	
<i>Elaborado por: Diego Santamaria</i>	

Tabla 152. Prueba de aceptación historia 039

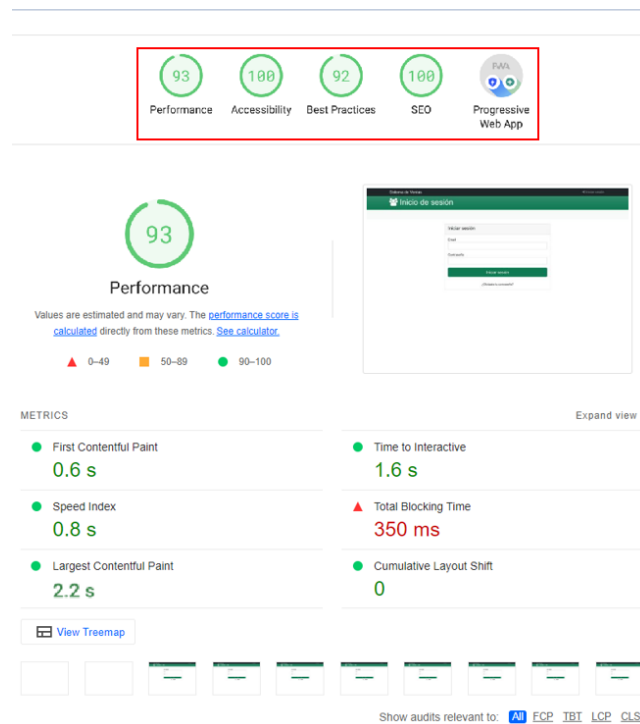
Prueba de aceptación	
Número: 38	Historia de usuario: 039
Nombre: Reportes	
Evaluated por: Sr. José David Holguín	
Descripción: En la pantalla de inicio se podrán observar los diferentes reportes que el usuario podrá generar en el sistema con la información proporcionada por el usuario vendedor.	
Condiciones de ejecución: Abrir la aplicación web progresiva. Seleccionar el tipo de reporte que se desea generar.	
Entrada: La información relacionada con este módulo debe estar almacenada en la base de datos NoSQL.	
Resultado esperado: El usuario gerente puede generar reportes de los pedidos, visitas, cobranzas y kilometrajes especificando ciertos criterios de búsqueda.	
Evaluación de la prueba: Prueba de aceptación admitida	

Elaborado por: Diego Santamaria

3.2.4.2 Pruebas de rendimiento

Para poder realizar las pruebas de rendimiento se utilizó la herramienta Lighthouse la cual permite auditar páginas y sistemas web mediante una serie de pruebas aplicadas al sitio web, posteriormente, esta herramienta crea un informe sobre el desempeño de la página o sistema.

Figura 175. Resultados de prueba de rendimiento.



Elaborado por: Diego Santamaria

Como se puede observar en el informe final de Lighthouse, el rendimiento se encuentra en un 93% debido a que la vida útil de la cache para ciertos módulos es corta lo que provoca un retardo en las vistas repetidas del sistema web.

Figura 176. Sugerencia de performance Lighthouse

DIAGNÓSTICO

▲ Servir activos estáticos con una política de caché eficiente — 6 recursos encontrados

A long cache lifetime can speed up repeat visits to your page. [Learn more.](#)

URL	Cache TTL	Transfer Size
/main.660f349...js (ambagro-ventas.web.app)	1 h	422 KB
/scripts.51e69d9...js (ambagro-ventas.web.app)	1 h	190 KB
/styles.7135e3d...css (ambagro-ventas.web.app)	1 h	22 KB
/polyfills.a15eb40...js (ambagro-ventas.web.app)	1 h	12 KB
/runtime.159c46b...js (ambagro-ventas.web.app)	1 h	2 KB
...js/emailNotification.js (ambagro-ventas.web.app)	1 h	1 KB

Elaborado por: Diego Santamaria

La parte de mejores prácticas se encuentra con un 92% debido a que se debe especificar de una mejor manera las rutas de los recursos utilizados en el sistema web para tener una depuración en producción más adecuada.

Figura 177. Sugerencia de Best practices Lighthouse

▲ Missing source maps for large first-party JavaScript

Source maps translate minified code to the original source code. This helps developers debug in production. In addition, Lighthouse is able to provide further insights. Consider deploying source maps to take advantage of these benefits. [Learn more.](#)

Show 3rd-party resources (3)

URL	Map URL
/scripts.51e69d9...js (ambagro-ventas.web.app)	Large JavaScript file is missing a source map
/main.660f349...js (ambagro-ventas.web.app)	Large JavaScript file is missing a source map

Elaborado por: Diego Santamaria

Como resumen general de las pruebas realizadas en el sistema web progresivo, la herramienta Lighthouse especifica lo siguiente:

Figura 178. Resumen de pruebas de rendimiento en Lighthouse

PASSED AUDITS (11)	Hide
● Uses HTTPS	▼
● Avoids requesting the geolocation permission on page load	▼
● Avoids requesting the notification permission on page load	▼
● Avoids front-end JavaScript libraries with known security vulnerabilities	▼
● Allows users to paste into password fields	▼
● Displays images with correct aspect ratio	▼
● Serves images with appropriate resolution	▼
● Page has the HTML doctype	▼
● Properly defines charset	▼
● Avoids deprecated APIs	▼
● No issues in the Issues panel in Chrome Devtools	▼

Elaborado por: *Diego Santamaria*

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- El desarrollo de la aplicación web progresiva para el control de vendedores y gestión de pedidos ayudó a mejorar la productividad y eficiencia de los vendedores de la empresa, así como también mejorar los procesos que intervienen en la creación y despacho de pedidos.
- Al realizar el análisis de los procesos que se utilizan para crear y despachar pedidos se pudo comprender de mejor manera los requerimientos y funcionalidades que tendrá el sistema web progresivo; también se logró mejorar dichos procesos para tener un mayor rendimiento por parte de los empleados.
- Al proponer un proceso para el control de los vendedores, la eficiencia y rendimiento de estos aumentó considerablemente, esto se pudo evidenciar en los reportes generados en la aplicación web progresiva y en los ingresos producidos. Otro punto que mejoró es la optimización de los recursos brindados por la empresa a los vendedores ya que con los reportes de kilometrajes se puede cotejar los gastos en relación a el uso del vehículo.
- Identificando las diferentes herramientas de software las cuales brindan varias utilidades para desarrollar un sistemas web progresivos, se pudo aprovechar al máximo estas características para obtener un producto final de calidad.
- La implementación de la aplicación web progresiva para el control de vendedores y gestión de pedidos fue satisfactoria, esto puede ser evidenciado mediante las pruebas de aceptación de las historias de usuario.

4.2. Recomendaciones

- Se recomienda investigar todos los servicios ofrecidos por Firebase ya que son muy útiles y sencillos de utilizar.
- Se recomienda tener en cuenta la versión de las librerías que se van a utilizar en el framework Angular ya que las rutas de los imports varían según la versión.
- Se recomienda utilizar la base de datos NoSQL Firestore ya que es muy interactiva y fácil de implementar y usar para proyectos de desarrollo que tengan más énfasis en la parte de información.
- Se recomienda utilizar aplicaciones web progresivas para el desarrollo de sistemas móviles ya que los costes son menores al momento de poner en producción estos sistemas.
- Se recomienda actualizar el plan Spark de Firebase al plan Blaze ya que la base de datos NoSQL Firestore necesitará más almacenamiento para los nuevos registros con el pasar del tiempo.

BIBLIOGRAFÍA

- [1] D. O. Guerra Poalasin and C. Nuñez, “Desarrollo de aplicación web progresiva para la gestión de pedidos aplicada a la empresa de ropa Hoxton,” Ambato, 2021. Accessed: Apr. 03, 2022. [Online]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/32673>
- [2] Padela, “Sistematizar procesos en tu empresa,” 2019. <https://www.lauratg.com/como-sistematizar-tu-negocio/> (accessed Apr. 03, 2022).
- [3] S. D. Yaguargos Castro and E. H. Álvarez Mayorga, “Aplicación web progresiva (PWA) para la automatización de los procesos de gestión e información en Liga Deportiva Parroquial Totoras,” Ambato, 2020. Accessed: Apr. 03, 2022. [Online]. Available: <https://repositorio.uta.edu.ec/jspui/handle/123456789/31308>
- [4] M. del P. Morejón Abril and E. F. Robalino Peña, “La información en bases de datos NoSQL y su incidencia en la generación documental de la secretaría general del honorable consejo universitario,” Ambato, 2018. Accessed: Apr. 03, 2022. [Online]. Available: <http://repositorio.uta.edu.ec/jspui/handle/123456789/27544>
- [5] ByteScout Team, “All about progressive web applications and newSQL.” <https://bytescout.com/blog/progressive-web-applications.html> (accessed Apr. 03, 2022).
- [6] J. Spero, “Por qué una aplicación web progresiva podría ser adecuada para ti,” in *Aplicaciones, tecnologías emergentes, móviles*, 2017. Accessed: Apr. 03, 2022. [Online]. Available: https://www.thinkwithgoogle.com/_qs/documents/2777/722b7_Progressive-Web-Apps-ES.pdf
- [7] Apps Ecuador, “Aplicación Móvil Nativa Vs Progressive Web APP (PWA).” <https://blog.enzymeadvisinggroup.com/aplicaciones-web-progresivas-o-aplicaciones-nativas> (accessed Apr. 03, 2022).
- [8] R. Samuel, “Errores comunes en la gestión de pedidos que podrías evitar,” <https://www.mygestion.com/blog/gestion-de-pedidos-errores>, Apr. 28, 2016.
- [9] J. Cisneros, “Evita 6 errores en la gestión de pedidos,” <https://www.datadec.es/blog/evita-6-errores-en-la-gestion-de-pedidos>, Aug. 14, 2020.
- [10] Universidades.cr, “Desarrollo de Software, ¿de qué trata?,” Oct. 07, 2019. <https://blog.universidades.cr/desarrollo-de-software-de-que-trata> (accessed Apr. 03, 2022).

- [11] I. Linares, “Qué son las aplicaciones web progresivas y cuáles son las mejores,” May 28, 2020. <https://www.xatakandroid.com/aplicaciones-android/que-aplicaciones-web-progresivas-cuales-mejores> (accessed Apr. 03, 2022).
- [12] Acenswhitepapers, “Bases de datos NoSQL. Qué son y tipos que nos podemos encontrar.” Accessed: Apr. 03, 2022. [Online]. Available: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- [13] R. Larrea, “Tesis de grado modelo de gestión administrativo,” 2008. Accessed: Apr. 03, 2022. [Online]. Available: <http://dspace.uniandes.edu.ec/handle/123456789/4976>
- [14] M. Montero, “¿Qué es un plan de ventas?,” 2017. <https://www.emprendepyme.net/que-es-un-plan-de-ventas.html> (accessed Apr. 03, 2022).
- [15] Kyocera, “Claves en la gestión de pedidos.” <https://www.kyoceradocumentsolutions.es/es/smarter-workspaces/insights-hub/articles/claves-en-la-gestion-de-pedidos.html#:~:text=Elementos%20claves%20en%20la%20gesti%C3%B3n%20de%20pedidos&text=Automatizaci%C3%B3n%20de%20la%20gesti%C3%B3n%20permite,elaboraci%C3%B3n%20del%20inventario%20de%20proveedores.> (accessed Apr. 03, 2022).
- [16] M. Maldonado, “Las mejores metodologías ágiles para la creación de software,” Apr. 12, 2018. <https://www.digital55.com/desarrollo-tecnologia/mejores-metodologias-agiles-creacion-software/> (accessed Apr. 03, 2022).
- [17] Ilucion Studio, “La metodología Kanban,” Aug. 07, 2019. <https://www.apd.es/metodologia-kanban/> (accessed Apr. 03, 2022).
- [18] J. Palacios, “Guía de Kanban.” https://resources.kanban.university/wp-content/uploads/2021/08/The-Official-Kanban-Guide_Spanish_A4.pdf (accessed Apr. 03, 2022).
- [19] Redacción APD, “Cómo aplicar la metodología Scrum y qué es el método Scrum,” Jan. 13, 2022. <https://www.apd.es/metodologia-scrum-que-es/#:~:text=La%20metodolog%C3%ADa%20Scrum%20es%20un,resultado%20de%20un%20proyecto%20determinado.> (accessed Apr. 03, 2022).
- [20] D. Calvo, “Metodología XP Programación Extrema (Metodología ágil),” Apr. 07, 2018. <https://www.diegocalvo.es/metodologia-xp-programacion-extrema-metodologia-agil/> (accessed Apr. 03, 2022).
- [21] J. L. Vila Grau, “La Metodología XP: la metodología de desarrollo de software más existosa,” Jul. 08, 2016. <https://proagilist.es/blog/agilidad-y-gestion-agil/agile-scrum/la-metodologia-xp/> (accessed Apr. 03, 2022).
- [22] Instituto Tecnológico de Matehuala, “Arquitectura de las aplicaciones Web,” Feb. 24, 2015. <https://sites.google.com/site/creaciondesitioweb84/arquitectura-de-las-aplicaciones-web> (accessed Apr. 03, 2022).

- [23] “Arquitectura de las aplicaciones Web.” <https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/> (accessed Apr. 03, 2022).
- [24] W. Narro, “Sistema Gestión de Citas Parte 2: Arquitectura N Capas + MVC,” Jun. 03, 2018. <http://www.bitacorasoftware.com/2018/06/sistema-gestion-de-citas-parte-2.html> (accessed Apr. 03, 2022).
- [25] Tecnologías Información, “Bases de datos SQL20,” 2018. <https://ayudaleyprotecciondatos.es/bases-de-datos/sql/> (accessed Apr. 03, 2022).
- [26] A. E. Martín, S. B. Chávez, N. R. Rodríguez, A. Valenzuela, and M. A. Murazzo, “Bases de datos NoSQL en cloud computing,” Jun. 2013, Accessed: Apr. 03, 2022. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/27121#:~:text=Las%20bases%20de%20datos%20NoSQL,en%20tablas%2C%20joins%20y%20transacciones>.
- [27] M. Manzo, “SQL vs NoSQL; Ventajas y Desventajas,” May 16, 2018. <https://medium.com/@marlonmanzo/sql-vs-nosql-ventajas-y-desventajas-849ccc9db3d4> (accessed Apr. 03, 2022).
- [28] M. Pérez Cardona, “Firebase, qué es y para qué sirve la plataforma de Google,” Oct. 14, 2016. <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/#:~:text=Firebase%20se%20trata%20de%20una,usuarios%20y%20ganar%20m%C3%A1s%20dinero>. (accessed Apr. 03, 2022).
- [29] M. Presta, “¿Firebase cuesta dinero?” <https://blog.back4app.com/es/firebase-cuesta-dinero/#:~:text=Este%20es%20el%20plan%20pago,y%20la%20transferencia%20de%20datos>. (accessed Apr. 03, 2022).
- [30] J. L. Coalla, “React | Qué es, para qué sirve y cómo funciona | Descúbrelo todo,” Jun. 23, 2021. <https://tech.tribalyte.eu/blog-que-es-react> (accessed Apr. 03, 2022).
- [31] Quality Devs, “¿Qué es Angular y para qué sirve?,” Sep. 16, 2019. <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/> (accessed Apr. 03, 2022).
- [32] E. I. García Pérez, “¿Qué Es Vue.JS?,” Apr. 01, 2016. <https://lenguajejs.com/vuejs/introduccion/que-es-vue/> (accessed Apr. 03, 2022).
- [33] J. L. Chacón, “TypeScript: qué es, diferencias con JavaScript y por qué aprenderlo,” Oct. 25, 2021. <https://profile.es/blog/que-es-typescript-vs-javascript/> (accessed Apr. 03, 2022).

- [34] Marketinhouse, “Qué es la geolocalización y cómo funciona,” 2018. <https://maplink.global/blog/es/geolocalizacion-movil-en-los-celulares/> (accessed Apr. 03, 2022).
- [35] J. A. Chuquilla Quinatoa and C. F. Aldás Flores, “Sistema web de gestión de documentos generados por los analizadores eléctricos, para el control de índices según la normativa Arconel en la empresa eléctrica Ambato regional centro norte S.A,” Ambato, 2021. Accessed: Apr. 03, 2022. [Online]. Available: <http://redi.uta.edu.ec/jspui/handle/123456789/80437>

ANEXOS

ANEXO A

NOTA DE PEDIDO



Telefax: (03) 2434097 / 2434088
Ambato - Ecuador

Fecha: 11/02/22

PEDIDO Nº 010899

Cliente: Alma Pineda Ross

Ciudad: Riobamba

R.U.C. No: _____

CANT.	EMPAQUE	PRODUCTO	V. UNTL.	TOTAL
20	250cc	AJO 35		
30	100cc	AJO 35		
30	200g	OLIVA		
20	200g	FESTYL		
30	600g	DIMECAL		
30	kg	FOSANYL		
20	kg	SILBY		
20	500cc	ODIN		
30	lt	ODIN		
30	lt	CHESTNUT		
30	lt	BOR CAL		
30	500cc	BOR CAL		
10	250cc	ALICORP		
10	500cc	ALICORP		

Negul Gue

VENDEDOR

[Signature]

COMPRADOR

B / Numerado desde el 10001 al 12000

ANEXO B
MANUAL DE USUARIO

El manual de usuario se puede encontrar en el siguiente enlace:

<https://drive.google.com/file/d/1OzWx3GJhKTkmkwIPF2IsF9Pguu0zZJh8/view?usp=sharing>