



**UNIVERSIDAD TECNICA DE AMBATO**

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

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

**Tema:**

---

APLICACIÓN WEB PROGRESIVA (PWA) PARA LA GESTIÓN DE PROCESOS DE  
COBRO DE LA JUNTA DE AGUA POTABLE PARA LA PARROQUIA HUAMBALÓ  
CASERÍO LA MERCED.

---

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

**ÁREA:** Administrativas informáticas

**LÍNEA DE INVESTIGACIÓN:** Aplicaciones informáticas

**AUTOR:** Alex Jair Villegas Alvarado

**TUTOR:** Ing. Leonardo David Torres Valverde

Ambato - Ecuador

septiembre - 2022

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: APLICACIÓN WEB PROGRESIVA (PWA) PARA LA GESTIÓN DE PROCESOS DE COBRO DE LA JUNTA DE AGUA POTABLE PARA LA PARROQUIA HUAMBALÓ CASERÍO LA MERCED, desarrollado bajo la modalidad Proyecto de Investigación por el señor Alex Jair Villegas Alvarado, 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 David Torres Valverde, Mg.

TUTOR

## AUTORÍA

El presente Proyecto de Investigación titulado: APLICACIÓN WEB PROGRESIVA (PWA) PARA LA GESTIÓN DE PROCESOS DE COBRO DE LA JUNTA DE AGUA POTABLE PARA LA PARROQUIA HUAMBALÓ CASERÍO LA MERCED, es absolutamente original, autentico 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

A handwritten signature in blue ink that reads "Alex Villegas". The signature is written over a horizontal line.

Alex Jair Villegas Alvarado

CC: 2300687510

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Alex Jair Villegas Alvarado, estudiante de la Carrera de e 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) PARA LA GESTIÓN DE PROCESOS DE COBRO DE LA JUNTA DE AGUA POTABLE PARA LA PARROQUIA HUAMBALÓ CASERÍO LA MERCED, 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.

PRESIDENTE DEL TRIBUNAL

-----  
Ing. Edison Álvarez

PROFESOR CALIFICADOR

-----  
Ing. Dennis Chicaiza, 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

A handwritten signature in blue ink, reading "Alex Villegas", is written over a horizontal line. The signature is stylized and includes a flourish that extends below the line.

Alex Jair Villegas Alvarado

CC: 2300687510

AUTOR

## **DEDICATORIA**

Este trabajo de investigación está dedicado a mis padres por criarme para ser la persona que soy hoy y quienes con su ejemplo y apoyo incondicional me han permitido cumplir mis metas.

## **AGRADECIMIENTO**

Agradezco a mis padres por los valores que me han inculcado, por siempre apoyarme en los objetivos que me he propuesto, por brindarme la oportunidad de seguir aprendiendo y poder servir a la sociedad.

De la misma forma, deseo expresar mi más sincero agradecimiento a mis tíos quienes han estado presentes apoyándome en cada momento de mi vida. Agradezco además a todos mis profesores por transmitirme sus conocimientos y ser el eje fundamental de mi formación profesional, en especial al Mg. Leonardo Torres por ayudarme incondicionalmente durante el desarrollo de este trabajo de investigación.

# ÍNDICE

APROBACIÓN DEL TUTOR.....	ii
AUTORÍA.....	iii
APROBACIÓN TRIBUNAL DE GRADO .....	iv
DERECHOS DE AUTOR .....	v
DEDICATORIA .....	vi
AGRADECIMIENTO.....	vii
1 MARCO TEORICO .....	1
1.1 Tema de Investigación.....	1
1.2 Antecedentes Investigativos.....	1
1.2.1 Contextualización del problema .....	1
1.2.2 Justificación .....	2
1.2.3 Delimitación.....	2
1.2.3.1 De contenidos.....	2
1.2.3.2 Espacial.....	3
1.2.3.3 Temporal.....	3
1.2.4 Fundamentación teórica .....	3
1.2.4.1 Aplicaciones Web.....	3
1.2.4.2 Comparativa aplicaciones web .....	6
1.2.4.3 Frameworks de desarrollo PWA.....	6
1.2.4.4 Metodologías ágiles de desarrollo .....	9



1.2.4.5	Node.js.....	12
1.2.4.6	Base de datos MySQL .....	13
1.2.4.7	Visual Studio Code.....	14
1.2.4.8	GitHub .....	15
1.3	Objetivos .....	15
1.3.1	Objetivo General .....	15
1.3.2	Objetivos Específicos .....	15
2	METODOLOGÍA.....	16
2.1	Materiales .....	16
2.1.1	Institucionales .....	16
2.1.2	Humano .....	16
2.1.3	Otros.....	16
2.1.4	Económicos.....	17
2.2	Métodos.....	17
2.2.1	Modalidad de la Investigación.....	17
2.2.2	Población y Muestra .....	18
2.2.3	Recolección de información.....	18
2.2.4	Procesamiento y análisis de datos.....	19
2.2.5	Resultado de la entrevista .....	20
2.2.6	Desarrollo del proyecto.....	20
3	RESULTADOS Y DISCUSIÓN .....	22

3.1	Análisis y discusión de los resultados.....	22
3.1.1	Comparativa de metodologías ágiles .....	22
3.1.2	Metodología de desarrollo seleccionada .....	22
3.1.3	Comparativa de frameworks de desarrollo PWA.....	23
3.1.4	Framework de desarrollo seleccionado .....	24
3.1.5	Determinación de la tecnología Back-End .....	24
3.1.6	Arquitectura de la aplicación.....	24
3.2	Desarrollo de la propuesta .....	25
3.2.1	Fase I: Planificación .....	25
3.2.1.1	Levantamiento de información .....	25
3.2.1.2	Análisis de resultados .....	27
3.2.1.3	Definición de roles.....	27
3.2.1.4	Modelado de procesos .....	27
3.2.1.5	Historias de usuario .....	30
3.2.1.6	Valoración de las historias de usuario.....	43
3.2.1.7	Estimación de historias de usuario.....	44
3.2.1.8	Tareas .....	45
3.2.1.9	Plan de entrega .....	60
3.2.2	Fase II: Diseño del proyecto.....	61
3.2.2.1	Metáfora de la aplicación .....	61
3.2.2.2	Iteración I.....	62

3.2.2.3	Iteración II.....	67
3.2.2.4	Iteración III .....	70
3.2.2.5	Iteración IV .....	72
3.2.2.6	Iteración V .....	75
3.2.2.7	Iteración VI.....	78
3.2.2.8	Iteración VII.....	82
3.2.2.9	Iteración VIII.....	85
3.2.3	Fase III: Codificación .....	91
3.2.3.1	Métodos del backend (Node.js).....	91
3.2.3.2	Métodos del Frontend (Angular - Servicios).....	106
3.2.3.3	Métodos del Frontend (Angular - Interfaces) .....	120
3.2.4	Fase IV: Pruebas.....	144
4	Conclusiones y recomendaciones .....	158
4.1	Conclusiones.....	158
4.2	Recomendaciones.....	158

## ÍNDICE DE ILUSTRACIONES

Ilustración 1. Logo de Vue.js .....	7
Ilustración 2. Logo de React .....	8
Ilustración 3. Logo de Angular .....	8
Ilustración 4. Proceso de las metodologías ágiles.....	10
Ilustración 5. Metodología SCRUM.....	11
Ilustración 6. Programación extrema.....	12
Ilustración 7. Logo de Node.js .....	12
Ilustración 8. Logo de MySQL .....	13
Ilustración 9. Logo de Visual Studio Code .....	14
Ilustración 10. Logo de GitHub.....	15
Ilustración 11. Arquitectura Cliente-Servidor .....	25
Ilustración 12. Módulos definidos .....	26
Ilustración 13. Ingreso de usuarios al sistema .....	28
Ilustración 14. Ingreso de socios al sistema .....	28
Ilustración 15. Modificación de usuarios y socios.....	28
Ilustración 16. Ingreso de mediciones al sistema.....	29
Ilustración 17. Cobro de facturas .....	29
Ilustración 18. Cuadre de caja.....	29
Ilustración 19. Compras.....	30
Ilustración 20. Inicialización del proyecto .....	64

Ilustración 21. Levantar el servidor de Angular .....	64
Ilustración 22. Organización de directorios del proyecto .....	65
Ilustración 23. Diseño de la base de datos .....	66
Ilustración 24. Interfaz módulo usuarios.....	67
Ilustración 25. Interfaz módulo socios.....	68
Ilustración 26. Interfaz módulo lecturas .....	69
Ilustración 27. Interfaz módulo de facturas.....	69
Ilustración 28. Interfaz módulo recaudación .....	70
Ilustración 29. Interfaz módulo mortuoria .....	71
Ilustración 30. Interfaz módulo cuadro de caja.....	71
Ilustración 31. Interfaz módulo compras .....	72
Ilustración 32. Interfaz módulo reportes .....	73
Ilustración 33. Interfaz Login.....	73
Ilustración 34. Listado de usuarios .....	74
Ilustración 35. Creación de usuarios.....	74
Ilustración 36. Edición de usuarios.....	75
Ilustración 37. Habilitar y deshabilitar usuarios .....	76
Ilustración 38. Listar socios .....	76
Ilustración 39. Crear socio .....	77
Ilustración 40. Editar socio .....	77
Ilustración 41. Habilitar y deshabilitar socios .....	78

Ilustración 42. Lista de lecturas.....	79
Ilustración 43. Agregar lecturas .....	79
Ilustración 44. Editar lecturas .....	80
Ilustración 45. Eliminar lecturas .....	80
Ilustración 46. Listar facturas.....	81
Ilustración 47. Cobro de facturas .....	81
Ilustración 48. Lisa mortuoria .....	82
Ilustración 49. Agregar mortuoria .....	83
Ilustración 50. Eliminar mortuoria .....	83
Ilustración 51. Listar cuadro de caja.....	84
Ilustración 52. Agregar cuadro de caja .....	84
Ilustración 53. Listar compras.....	85
Ilustración 54. Agregar compras .....	86
Ilustración 55. Editar compra.....	86
Ilustración 56. Eliminar compra.....	87
Ilustración 57. Reporte de socios .....	87
Ilustración 58. Reporte de usuarios .....	88
Ilustración 59. Reporte de lecturas .....	88
Ilustración 60. Reporte de facturas .....	89
Ilustración 61. Reporte de mortuorias.....	89
Ilustración 62. Reporte de cuadros .....	90

Ilustración 63. Reporte de compras .....	90
Ilustración 64. Autenticación .....	91
Ilustración 65. Verificación de JSON Web Token .....	92
Ilustración 66. Verificación del rol del usuario .....	92
Ilustración 67. Cambio de contraseña.....	93
Ilustración 68. Listado de usuarios .....	93
Ilustración 69. Agregar usuario.....	94
Ilustración 70. Editar usuario .....	94
Ilustración 71. Buscar usuario.....	95
Ilustración 72. Listar socios .....	95
Ilustración 73. Agregar socio .....	96
Ilustración 74. Editar socio .....	96
Ilustración 75. Buscar socio.....	97
Ilustración 76. Listar lecturas.....	97
Ilustración 77. Agregar lectura.....	98
Ilustración 78. Editar lectura.....	98
Ilustración 79. Eliminar lectura.....	99
Ilustración 80. Listar facturas.....	99
Ilustración 81. Agregar lectura.....	99
Ilustración 82. Buscar facturas por fechas .....	100
Ilustración 83. Buscar lecturas por factura.....	100

Ilustración 84. Buscar lecturas por socio .....	101
Ilustración 85. Buscar lecturas para cobro .....	101
Ilustración 86. Buscar lecturas por socio .....	102
Ilustración 87. Listar mortuoria.....	102
Ilustración 88. Agregar mortuoria .....	103
Ilustración 89. Eliminar mortuoria .....	103
Ilustración 90. Buscar mortuoria .....	104
Ilustración 91. Listar cuadros de caja.....	104
Ilustración 92. Agregar cuadro de caja .....	104
Ilustración 93. Listado de compras.....	105
Ilustración 94. Agregar compra.....	105
Ilustración 95. Editar compra.....	106
Ilustración 96. Eliminar compra.....	106
Ilustración 97. Login .....	107
Ilustración 98. Verificación del token.....	107
Ilustración 99. Guard rutas aplicación .....	108
Ilustración 100. Generar cabecera para las consultas.....	108
Ilustración 101. Listar usuarios .....	109
Ilustración 102. Agregar usuario .....	109
Ilustración 103. Editar usuario .....	109
Ilustración 104. Cambiar contraseña .....	110



Ilustración 105. Buscar usuario .....	110
Ilustración 106. Listar socios .....	111
Ilustración 107. Agregar socio .....	111
Ilustración 108. Editar socio .....	111
Ilustración 109. Buscar socios.....	112
Ilustración 110. Listar lecturas .....	112
Ilustración 111. Agregar lectura.....	113
Ilustración 112. Editar lectura .....	113
Ilustración 113. Eliminar lectura .....	113
Ilustración 114. Buscar lectura para cobro.....	114
Ilustración 115. Buscar lectura por fechas .....	114
Ilustración 116. Listar facturas.....	115
Ilustración 117. Agregar factura.....	115
Ilustración 118. Buscar facturas por fechas.....	115
Ilustración 119. Buscar lecturas por facturas .....	116
Ilustración 120. Buscar lecturas por socio .....	116
Ilustración 121. Listar mortuoria.....	117
Ilustración 122. Agregar mortuoria .....	117
Ilustración 123. Agregar mortuoria .....	117
Ilustración 124. Listar cuadro de caja .....	118
Ilustración 125. Agregar cuadro de caja.....	118

Ilustración 126. Listar compras.....	119
Ilustración 127. Agregar compra.....	119
Ilustración 128. Editar compra.....	119
Ilustración 129. Eliminar compra.....	120
Ilustración 130. Abrir modales.....	120
Ilustración 131. Formularios reactivos usuarios.....	121
Ilustración 132. Cambiar estado del usuario.....	122
Ilustración 133. Guardar y modificar usuario.....	123
Ilustración 134. Validar cedula.....	124
Ilustración 135. Buscar usuario.....	125
Ilustración 136. Formulario reactivo socios.....	125
Ilustración 137. Guardar y editar socios.....	126
Ilustración 138. Cambiar estado del socio.....	127
Ilustración 139. Buscar socio.....	127
Ilustración 140. Formulario reactivo lecturas.....	128
Ilustración 141. Guardar y editar lectura.....	129
Ilustración 142. Siguiete lectura.....	130
Ilustración 143. Lectura anterior.....	131
Ilustración 144. Buscar socio.....	132
Ilustración 145. Calcular totales.....	133
Ilustración 146. Eliminar lectura.....	133

Ilustración 147. Buscar lecturas .....	134
Ilustración 148. Generar PDF .....	134
Ilustración 149. Buscar lecturas por fechas .....	135
Ilustración 150. Buscar mortuoria por socios .....	135
Ilustración 151. Buscar mortuoria por fechas.....	136
Ilustración 152. Guardar mortuoria .....	136
Ilustración 153. Buscar socio mortuoria .....	137
Ilustración 154. Eliminar mortuoria .....	137
Ilustración 155. Formulario reactivo cuadro de caja .....	138
Ilustración 156. Verificar cuadro de caja .....	139
Ilustración 157. Verificar cuadro de caja .....	139
Ilustración 158. Guardar cuadro de caja.....	140
Ilustración 159. Buscar cuadro de caja .....	141
Ilustración 160. Formulario reactivo compras.....	141
Ilustración 161. Guardar y editar compras .....	142
Ilustración 162. Eliminar compra .....	143
Ilustración 163. Calcular total compra.....	143
Ilustración 164. Buscar compras .....	144

## ÍNDICE DE TABLAS

Tabla 1. Comparativa aplicaciones web .....	6
Tabla 2. Cuadro de Presupuesto .....	17
Tabla 3. Población y Muestra.....	18
Tabla 4. Plan de recolección de la información.....	19
Tabla 5. Comparativa de metodologías ágiles .....	22
Tabla 6.Comparativa de frameworks de desarrollo PWA.....	23
Tabla 7. Definición de roles.....	27
Tabla 8.Plantilla de historias de usuario .....	30
Tabla 9. Historia de usuario - Estructura del proyecto.....	32
Tabla 10. Historia de usuario - Diseño de la base de datos .....	32
Tabla 11. Historia de usuario - Diseño de interfaz módulo usuarios.....	32
Tabla 12. Historia de usuario - Diseño de interfaz módulo socios.....	33
Tabla 13. Historia de usuario - Diseño de interfaz módulo lecturas .....	33
Tabla 14. Historia de usuario - Diseño de interfaz módulo de facturas.....	33
Tabla 15. Historia de usuario - Diseño de interfaz módulo recaudación.....	34
Tabla 16. Historia de usuario - Diseño interfaz modulo mortuoria.....	34
Tabla 17. Historia de usuario - Diseño de interfaz módulo cuadro de caja.....	34
Tabla 18. Historia de usuario - Diseño de interfaz módulo compras .....	35
Tabla 19. Historia de usuario - Diseño de interfaz módulo reportes .....	35
Tabla 20. Historia de usuario - Inicio de sesión .....	35

Tabla 21. Historia de usuarios - Listado de usuarios .....	36
Tabla 22. Historia de usuario - Creación de usuarios .....	36
Tabla 23. Historia de usuarios - Edición de usuarios .....	36
Tabla 24. Historia de usuario - Habilitar y deshabilitar de usuarios .....	37
Tabla 25. Historia de usuario - Listar socios .....	37
Tabla 26. Historia de usuario - Crear socio.....	37
Tabla 27. Historia de usuario - Editar socios .....	38
Tabla 28. Historia de usuario - Habilitar y deshabilitar socios .....	38
Tabla 29. Historia de usuario - Listar lecturas.....	38
Tabla 30. Historia de usuario - Agregar lecturas .....	39
Tabla 31. Historia de usuario - Editar lecturas .....	39
Tabla 32. Historia de usuario - Eliminar lecturas .....	39
Tabla 33. Historia de usuario - Listar Facturas.....	40
Tabla 34. Historia de usuario - Cobro de factura.....	40
Tabla 35. Historia de usuario - Listar mortuoria .....	40
Tabla 36. Historia de usuario - Agregar mortuoria .....	41
Tabla 37. Historia de usuario - Eliminar mortuoria .....	41
Tabla 38. Historia de usuario – Listar cuadro de caja .....	41
Tabla 39. Historia de usuario - Agregar cuadro de caja .....	42
Tabla 40. Historia de usuario - Listar compras.....	42
Tabla 41. Historia de usuario - Agregar compra .....	42

Tabla 42. Historia de usuario - Editar compra.....	43
Tabla 43. Historia de usuario - Eliminar compra.....	43
Tabla 44. Historia de usuario - Generación de reportes .....	43
Tabla 45. Estimación de historias de usuario .....	44
Tabla 46. Plantilla de tareas de historias de usuario .....	45
Tabla 47. Tarea - Estructurar el proyecto.....	45
Tabla 48. Tarea - Diseño de la base de datos .....	45
Tabla 49. Tarea - Diseño de interfaz módulo usuarios.....	46
Tabla 50. Tarea - Diseño de interfaz módulo socios.....	46
Tabla 51. Tarea - Diseño de interfaz módulo de lecturas .....	46
Tabla 52. Tarea - Diseño de interfaz módulo facturas .....	47
Tabla 53. Tarea - Diseño de interfaz módulo recaudación .....	47
Tabla 54. Tarea - Diseño de interfaz módulo cuadro de caja.....	47
Tabla 55. Tarea - Diseño de interfaz módulo cuadro de caja.....	48
Tabla 56. Tarea - Diseño de interfaz módulo compras .....	48
Tabla 57. Tarea - Diseño de interfaz módulo reportes .....	48
Tabla 58. Tarea - Diseño de interfaz inicio de sesión .....	49
Tabla 59. Tarea - Listado de usuarios.....	49
Tabla 60. Tarea - Creación de usuarios.....	49
Tabla 61. Tarea - Edición de usuarios .....	50
Tabla 62. Tarea - Habilitar y deshabilitar usuarios .....	50

Tabla 63. Tarea - Listar socios .....	50
Tabla 64. Tarea - Crear socio.....	51
Tabla 65. Tarea - Editar socios.....	51
Tabla 66.Tarea - Habilitar y deshabilitar socios .....	51
Tabla 67. Tarea - Listar lecturas.....	52
Tabla 68. Tarea - Agregar lecturas .....	52
Tabla 69. Tarea - Editar lecturas .....	52
Tabla 70. Tarea - Eliminar lecturas .....	53
Tabla 71. Tarea - Listar facturas .....	53
Tabla 72. Tarea - Cobro de facturas .....	53
Tabla 73. Tarea - Listar mortuoria.....	54
Tabla 74. Tarea - Agregar mortuoria.....	54
Tabla 75. Tarea - Eliminar mortuoria .....	54
Tabla 76. Tarea – Listar cuadro de caja.....	55
Tabla 77. Tarea – Agregar cuadro de caja.....	55
Tabla 78. Tarea – Listar compras .....	55
Tabla 79. Tarea – Agregar compra.....	56
Tabla 80. Tarea – Editar compra .....	56
Tabla 81. Tarea – Eliminar compra .....	56
Tabla 82. Tarea – Reportes Socios.....	57
Tabla 83. Tarea - Reportes usuarios .....	57

Tabla 84. Tarea - Reporte lecturas.....	57
Tabla 85. Tarea – Reportes facturas .....	58
Tabla 86. Tarea - Reportes mortuoria .....	58
Tabla 87. Tarea – Reportes cuadro de caja.....	58
Tabla 88. Tarea - Reportes compras .....	59
Tabla 89. Planificación de entrega .....	60
Tabla 90. Estimación de la primera iteración .....	62
Tabla 91. Estimación de la segunda iteración .....	67
Tabla 92. Estimación de la tercera iteración .....	70
Tabla 93. Estimación de la cuarta iteración .....	72
Tabla 94. Estimación de la quinta iteración .....	75
Tabla 95. Estimación de la sexta iteración .....	78
Tabla 96. Estimación de la séptima iteración .....	82
Tabla 97. Estimación de la octava iteración .....	85
Tabla 98. Prueba de acepción 001 .....	145
Tabla 99. Prueba de acepción 002.....	145
Tabla 100. Prueba de acepción 003.....	146
Tabla 101. Prueba de acepción 004.....	146
Tabla 102. Prueba de acepción 005.....	147
Tabla 103. Prueba de acepción 006.....	147
Tabla 104. Prueba de acepción 007.....	148



Tabla 105. Prueba de acepción 008.....	148
Tabla 106. Prueba de acepción 009.....	149
Tabla 107. Prueba de acepción 010.....	149
Tabla 108. Prueba de acepción 011.....	150
Tabla 109. Prueba de acepción 012.....	150
Tabla 110. Prueba de acepción 013.....	151
Tabla 111. Prueba de acepción 014.....	151
Tabla 112. Prueba de acepción 015.....	152
Tabla 113. Prueba de acepción 016.....	152
Tabla 114. Prueba de acepción 017.....	153
Tabla 115. Prueba de acepción 018.....	153
Tabla 116. Prueba de acepción 019.....	154
Tabla 117. Prueba de acepción 020.....	154
Tabla 118. Prueba de acepción 021.....	155
Tabla 119. Prueba de acepción 022.....	155
Tabla 120. Prueba de acepción 023.....	156
Tabla 121. Prueba de acepción 024.....	156
Tabla 122. Prueba de acepción 024.....	157

## RESUMEN EJECUTIVO

El presente trabajo está basado en el mejoramiento de la gestión de procesos de la junta de agua potable Huambaló caserío La Merced, dado que la información que se maneja no se encuentra registrada de una forma óptima y esto la hace susceptible a pérdidas y de igual forma brindar un mayor control la información manejada, así como también el control de acceso a los mismos datos manejando roles para realizar diferentes acciones dentro de la aplicación, mejorando de esta forma la seguridad de los datos y la gestión de procesos.

La aplicación fue desarrollada utilizando el framework Angular en su versión 13.3.1, el cual utiliza como motor el lenguaje TypeScript el cual es un lenguaje de programación libre y es mantenido por la empresa Microsoft, TypeScript es un superconjunto de JavaScript que en esencia añade tipos estáticos y objetos basados en clases. El Back-end se desarrolló en Node.js en su versión 14.15.0 y con el package manager versión 13.3.1, ofrece una amplia variedad de librerías las cuales permiten desarrollar servidores robustos y seguros.

La metodología utilizada para el desarrollo de esta aplicación fue la metodología XP (Xtreme Programming), esta es una metodología de desarrollo ágil, lo que significa que se centra en los datos importantes de esta forma avanzando rápidamente con el diseño y la programación, evitando en la medida de lo posible la documentación innecesaria.

Palabras clave: Angular, Node.js, TypeScript, PWA, XP, JavaScript, Visual Studio Code.

## **ABSTRACT**

The present work is based on the improvement of the process management of the drinking water board Huambaló hamlet La Merced, since the information that is handled is not registered in an optimal way and this makes it susceptible to losses and in the same way provide greater control of the information handled, as well as access control to the same data handled by roles to perform different actions within the application, thus improving data security and process management.

The application was developed using the Angular framework in its version 13.3.1, which uses the TypeScript language as its engine, which is a free programming language and is maintained by Microsoft, TypeScript is a superset of JavaScript that essentially adds types static and class-based objects. The Back-end is developed in Node.js version 14.15.0 and with the package manager version 13.3.1, it offers a wide variety of libraries which allow developing robust and secure servers.

The methodology used for the development of this application was the XP methodology (Xtreme Programming), this is an agile development methodology, which means that it focuses on the important data in this way, advancing rapidly with the design and programming, avoiding in as much as possible unnecessary documentation.

Keywords: Angular, Node.js, TypeScript, PWA, XP, JavaScript, Visual Studio Code.

## INTRODUCCIÓN

El actual trabajo de investigación denominado “APLICACIÓN WEB PROGRESIVA (PWA) PARA LA GESTIÓN DE PROCESOS DE COBRO DE LA JUNTA DE AGUA POTABLE PARA LA PARROQUIA HUAMBALÓ CASERÍO LA MERCED.” Se encuentra dividido en los siguientes capítulos.

**CAPITULO I:** “MARCO TEORICO”, se describe la problemática que presenta la junta de agua potable Huambaló caserío La Merced, justificando de este modo las razones por las cuales se realizó el desarrollo de esta investigación, además de la contextualización del problema maso, meso y micro de trabajos que tienen relación con la problemática a resolver.

**CAPITULO II:** “METODOLOGIA”, en este capítulo se definen las etapas de desarrollo del proyecto, los materiales a utilizarse, las etapas que conlleva el desarrollo del proyecto y la modalidad de la investigación.

**CAPITULO III:** “RESULTADOS Y DISCUIONES”, se detalla de forma puntual el desarrollo del proyecto de investigación, se establece la metodología con la cual va a ser desarrollado y las diferentes herramientas que serán utilizadas para el correcto funcionamiento del proyecto.

**CAPITULO IV:** “CONCLUSIONES Y RECOMENDACIONES”, se definen las conclusiones y recomendaciones del proyecto de investigación

# CAPÍTULO I

## 1 MARCO TEORICO

### 1.1 Tema de Investigación

APLICACIÓN WEB PROGRESIVA (PWA) PARA LA GESTIÓN DE PROCESOS DE COBRO DE LA JUNTA DE AGUA POTABLE PARA LA PARROQUIA HUAMBALÓ CASERÍO LA MERCED.

### 1.2 Antecedentes Investigativos

#### 1.2.1 Contextualización del problema

Actualmente en el mundo las empresas se han visto obligadas a operar de manera más competente por diversas razones entre una de ellas la pandemia mundial por el COVID-19, por ende se han visto en la necesidad de optimizar la gestión financiera, para esto, se han desarrollado diversos sistemas informáticos que permiten gestionar, obtener información ordenada y automatizada con la finalidad de agilizar los procesos dentro de la empresa y generar un alto nivel de competitividad [1].

El software contable se han convertido en una herramienta de información económica necesaria dentro de los negocios fundamentalmente en las empresas de servicios públicos de los países en desarrollo; haciendo énfasis en que las empresas que brindan servicios públicos en los países en desarrollo son tomadas como ineficientes en la prestación de sus servicios, y un factor importante es la mala gestión de sus cuentas por cobrar [2]. Las empresas necesitan de un sistema que permita a cualquier miembro que solicite la información económica obtenerla de una manera rápida, con lo cual los negocios están en condiciones de realizar notables progresos en materia de eficiencia, crecimiento y beneficios [3].

En la provincia de Tungurahua, existen diversos inconvenientes en las pequeñas y medianas empresas los cuales están relacionados en el ámbito administrativo y financiero, las empresas buscan la automatización de procesos y de información con herramientas que ahorran tiempo y personal, incrementando y potenciando sus actividades comerciales [4].

En el cantón Ambato se pueden identificar problemas a nivel de la competitividad de las pequeñas y medianas empresas, específicamente en el ámbito de conocimiento y aplicación de nuevas tecnologías [5].

Así como en la Junta de Agua Potable de la parroquia Huambaló caserío La Merced existen muchos inconvenientes para un buen desarrollo de la institución. Dichos inconvenientes se presentan porque no existe una automatización de la información que posee y maneja la institución, además de que no existen estados financieros actuales, por otro lado, la información se maneja de forma física por lo que no existen respaldos de documentos contables.

### **1.2.2 Justificación**

La presente investigación tiene la finalidad de ayudar en el correcto manejo de la gestión de cuentas por cobrar de la junta de agua potable de la parroquia Huambaló ya que es una institución pública la cual requiere de estos cobros para la realización de nuevas obras por ende es necesaria una herramienta informática la cual ayude a realizar de forma ordenada estos procesos y de esta forma evitar pérdidas financieras.

Con el presente proyecto se pretende beneficiar a los socios y miembros pertenecientes a la junta de agua potable de la parroquia Huambaló puesto que se agilizar los procesos de cobro y se gestionara de mejor forma información.

El proyecto resulta innovador puesto que no cuentan con una herramienta tecnológica que permita gestionar adecuadamente las cuentas por cobrar, la implementación de una PWA permitirá realizar una mejora en la gestión de la información, agilización en los procesos de cobro y ayudara a realizar una adecuada toma de decisiones, dado que se automatizarán procesos y mejora la interacción con los usuarios.

### **1.2.3 Delimitación**

#### **1.2.3.1 De contenidos**

**Área Académica:** Administrativas informáticas

**Línea de investigación:** Aplicaciones informáticas

**Sublínea de investigación:** Gestión Presupuestaria.

### **1.2.3.2 Espacial**

Junta de agua potable de la Parroquia Huambaló.

### **1.2.3.3 Temporal**

La presente investigación se desarrollará en el periodo septiembre 2021 – julio 2022.

## **1.2.4 Fundamentación teórica**

### **1.2.4.1 Aplicaciones Web**

En la actualidad las aplicaciones web están revolucionando el mercado de las herramientas de software tomando en cuenta las ventajas que estas aportan y el hecho de que se necesita internet para todo tipo de transacciones esto ha hecho que el uso de las aplicaciones de escritorio se vuelva obsoleto [6].

Una de las principales ventajas que se pueden recalcar es que, a diferencia de una aplicación de escritorio una aplicación web se aloja en un servidor y puede ser accedida a través de diferentes dispositivos que cuenten con acceso a internet o dentro de una intranet, no requiere de programas instalables, la información que se encuentra almacenada en la aplicación web puede ser compartida por varias personas a la vez y las copias de seguridad se almacenan en los servidores.

### **Aplicación Web Progresiva**

Las Aplicaciones Web Progresivas (PWA) son una nueva generación de aplicaciones que aprovechan las principales tecnologías web (HTML, CSS y JavaScript) para la creación de interfaces para los servicios de Internet. A diferencia de las aplicaciones móviles o las de escritorio, las aplicaciones web progresivas pueden ejecutarse en cualquier navegador compatible con los estándares, además del rendimiento o el incremento de su funcionalidad según las capacidades del dispositivo y de ahí su denominación Progresiva [7].

Las Aplicaciones Web Progresivas unen las mejores características de aplicaciones web y aplicaciones nativas tales como:

- Mejor rendimiento
- Tiempos de carga muy reducidos
- Interfaz similar a la de una app nativa
- No ocupan espacio en la memoria
- Diseño responsive
- Notificaciones push
- Protocolo seguro TLS
- Menor coste de desarrollo

Estas características están transformando a las PWA en el nuevo modelo para el desarrollo de aplicaciones web y móviles, debido a que operan en cualquier navegador.

La arquitectura de una PWA está compuesta de la siguiente manera:

**Una página:** Aquí se mostrará la información que contenga la aplicación.

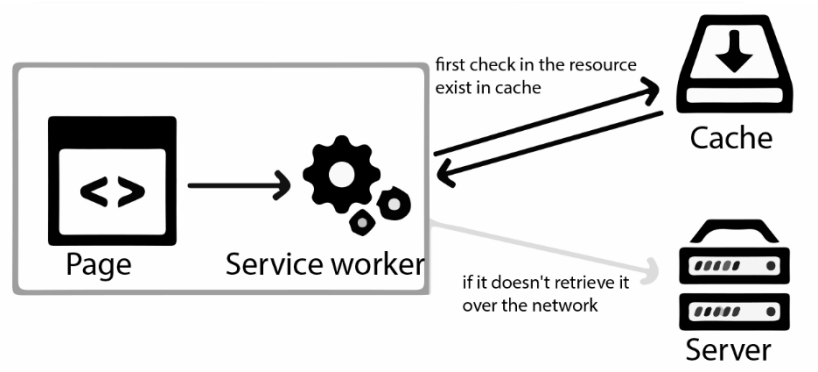
**Service worker:** Es una secuencia de comandos que el navegador ejecuta en segundo plano que permite interceptar y controlar los requests de la red y el almacenamiento del cache del navegador.

**Cache:** Almacena los datos procesados recientemente en una memoria auxiliar.

**Server:** En donde se encuentra alojada la aplicación.

Este esquema prioriza la cache y en caso de que la información requerida no se encuentre disponible en la cache procede a buscar dicha información en el servidor.

Figura 1. Arquitectura de una PWA



Elaborado por: El investigador



**HTML:** Lenguaje de Marcado de Hipertexto (HTML) es un estándar reconocido a nivel mundial y cuyas normas define el organismo World Wide Web Consortium (W3C). HTML5 es la versión más actual de lenguaje de marcado, esta nueva versión no solo incorporar nuevas etiquetas y elimina otras de las versiones anterior, sino que plantea mejoras en áreas que hasta ahora no habían sido abarcadas del lenguaje como la incorporación de multimedia o validación de formularios y para las que era necesario utilizar otras tecnologías [8].

HTML5 establece un avance hacia el diseño de sitios Web. Este estándar aporta funcionalidades que posiblemente aun no estén disponibles para algunos navegadores, pero, se han dado los primeros pasos para incluirlas a la Web [8].

**CSS:** Cascading Style Sheets(CSS) es lo que se denomina lenguaje de hojas de estilo en cascada y se usa para estilizar elementos escritos en un lenguaje de marcado como HTML. CSS permite separa el contenido de la representación visual del sitio.

CSS fue desarrollado por W3C (World Wide Web Consortium) en 1996 por un motivo muy sencillo. HTML no fue diseñado con el fin de tener etiquetas que ayuden a formatear la página. Está hecho solo para escribir el marcado para el sitio [9].

**JavaScript:** JavaScript es el lenguaje de programación encargado brindar una mayor interactividad y dinamismo a las páginas web. Cuando JavaScript se ejecuta en el navegador, no necesita de un compilador. El navegador lee el código, sin necesidad de terceros [10].

## **Aplicaciones Nativas**

Las aplicaciones nativas son aquellas que están desarrolladas para funcionar en una plataforma específica, es decir, se tiene que tomar en consideración el sistema operativo en el que se va a utilizar y su versión [11].

Se realiza la compilación del código fuente para obtener código ejecutable, este proceso es parecido al que se utilizada tradicionalmente en las aplicaciones de escritorio [11]. Cuando las aplicaciones están listas se procede a realizar la publicación en las diferentes App stores (tiendas de aplicaciones) dependiendo del lenguaje en el que fue desarrollado puede ser Play Store(Apps de Android) o App Store(Apps de Apple). Estas pasan por un proceso de revisión para determinar si la aplicación se adecúa a los requerimientos de la

plataforma en la que fue publicada. Una vez aprobada esta revisión las aplicaciones se pone a disposición de los usuarios.

## Web Responsive

Un sitio web responsive es aquel que es capaz de adaptarse a al tamaño del dispositivo en el que se esté visualizando. Hoy en día los usuarios acceden a los sitios web desde diferentes dispositivos tales como pueden ser una laptop, una tablet o un celular. Y en cada uno de estos dispositivos el sitio web debe visualizarse correctamente [12].

El diseño responsivo se encarga de responder al tamaño de los dispositivos desde los que se visualizan los contenidos web adaptando componentes de manera optimizada, adaptando sus dimensiones y mostrándolo [12].

### 1.2.4.2 Comparativa aplicaciones web

Tabla 1. Comparativa aplicaciones web

<b>Características</b>	<b>Aplicación Web Progresiva</b>	<b>Aplicaciones Nativas</b>	<b>Web Responsive</b>
Multi plataforma	Si	No	No
Costo	Bajo	Alto	Bajo
Instalable	Si	Si	No
Usabilidad sin conexión	Si	Si	No
Rendimiento	Medio	Alto	Bajo
Utilización de características de dispositivos	Medio	Alto	Bajo
Implementación	Fácil	Medio/Difícil	Fácil

Elaborado por: El investigador

### 1.2.4.3 Frameworks de desarrollo PWA

La mayor parte de los sitios web actuales usan JavaScript para hacer que las páginas sean más interactivas y para darles funcionalidad a las mismas. Las páginas web tradicionales son aplicaciones de diversas páginas donde se carga un nuevo documento HTML cada vez que el usuario cambia su contenido. Esta opción es relativamente lenta e ineficiente

en comparación con las páginas web modernas del modelo de desarrollo de la aplicación web progresiva (PWA) en estas aplicaciones solo las partes que cambian se actualizan. El uso del patrón PWA reduce la velocidad de carga de la aplicación y mejora la experiencia del usuario.

Las PWA ganan popularidad a medida que avanza el tiempo y de la misma forma los frameworks a los que están integradas. Los frameworks dictan un marco de trabajo, reducen el tiempo empleado para el desarrollo y ayudan a evitar posibles errores. Cada framework es diferente y cuenta con sus respectivas ventajas y desventajas por lo cual es importante seleccionar el correcto para el proyecto de desarrollo de software.

## **Vue**

Ilustración 1. Logo de Vue.js



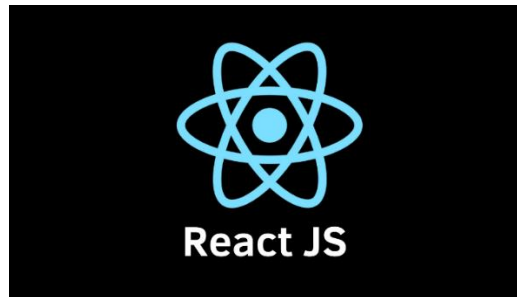
Fuente: [13]

Vue es un framework progresivo el cual se utiliza para la construcción de interfaces de usuario. Vue está diseñado desde cero para ser utilizado incrementalmente o hasta lograr sofisticadas aplicaciones. Vue es fácil de utilizar y se puede integrar con otras librerías o proyectos existentes. Adicional a esto Vue es capaz de impulsar sofisticadas SPA(Single Page Applications) utilizando herramientas modernas y librerías de apoyo [14] .

Vue se enmarca en las arquitecturas de componentes con una gestión de modelos basada en el patrón MVVM que es similar a las arquitecturas MVC, con la excepción del modelo de vista por lo que permite manipular los estados de aplicación [14].

## React

Ilustración 2. Logo de React



Fuente: [15]

React es un entorno de trabajo de JavaScript. Fue desarrollado inicialmente por ingenieros de Facebook para facilitar el desarrollo de interfaces de usuario complejas con conjuntos de datos que cambian constantemente con el tiempo. React nació en realidad en la organización de anuncios de Facebook, donde habían estado utilizando un controlador de vista de modelo tradicional del lado del cliente. Este tipo de aplicaciones normalmente se basan en un enlace de datos bidireccional junto con una plantilla de representación. React cambió la forma en que se crearon estas aplicaciones al realizar algunos avances audaces en el desarrollo web [16].

React realizó la introducción de muchos paradigmas nuevos y cambiando el status quo de lo que se necesita para crear aplicaciones JavaScript e interfaces de usuario escalables y mantenibles. Además del cambio en la mentalidad de desarrollo front-end, React cuenta con un amplio conjunto de características que facilitan del desarrollo de aplicaciones [16].

## Angular

Ilustración 3. Logo de Angular



Fuente: [17]

Angular es un framework de licencia libre desarrollado por Google con el objetivo de facilitar la creación y programación de aplicaciones web. Angular permite la creación aplicaciones de una sola página(SPA), esto quiere decir que permite cargar diferentes partes de la aplicación sin tener que recargar todo el contenido en el navegador [18].

Este framework divide completamente el front-end y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) proporcionando rapidez al momento del desarrollo de aplicaciones y a su vez facilitando la modificación y actualización de las mismas [18].

Angular utiliza la inyección de dependencias declarando controladores en el constructor de la clase de forma tal que el controlador puede utilizar estas dependencias como sea necesario. De esta forma el controlador no necesita saber cómo funciona la dependencia ni cuáles son las acciones que realiza para entregar los resultados [19].

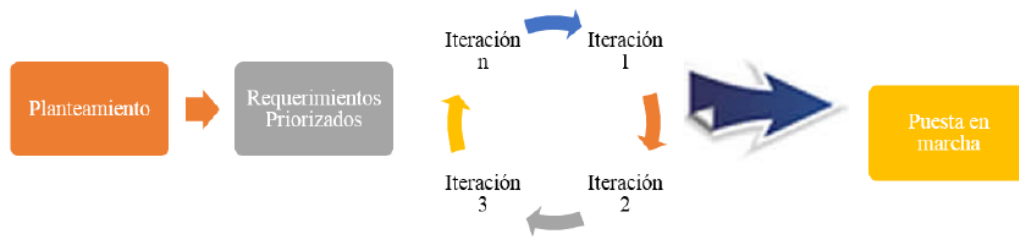
#### **1.2.4.4 Metodologías ágiles de desarrollo**

Las metodologías ágiles son metodologías adaptativas, estas permite realizar proyectos de desarrollo de software, adecuarlo a los cambios con el fin de mejorar el sistema y maximizar la satisfacción del cliente, considerando la gestión de cambios como un aspecto inherente al propio proceso de desarrollo software de este modo permitiendo mejorar la adaptación al entorno, incrementando la inversión y disminuyendo los costos, ya sea para modificar alguna funcionalidad o añadir otra nueva [20].

Para el desarrollo de software existen diversas propuestas de metodologías que influyen en distintas dimensiones del proceso, desde las metodologías tradicionales hasta las actuales metodologías ágiles que son las alternativas más usadas en el desarrollo de proyectos de software actuales [20].

Ilustración 4. Proceso de las metodologías ágiles

## Metodología de desarrollo Ágiles



Fuente: [21]

### Principales metodologías ágiles

#### Scrum

Scrum forma parte de las metodologías de desarrollo de software más utilizadas a nivel mundial, esta metodología data de los años 80 y fue introducida por Ikujiro Sutherland y Hirotaka Takeuchi, posteriormente fue utilizado por Jeff Sutherland y Ken Schwaber, estos últimos formalizaron un marco de trabajo y establecieron reglas que serían aplicadas al desarrollo de software [22].

Se escogió el nombre Scrum por su similitud con el juego de rugby y en los procesos que este juego proponía los cuales con: rápido, adaptable, auto organizable y con pocos descansos [21].

Entre los principios de la metodología scrum es lograr la simplicidad y escalabilidad, ya que no plantea prácticas de ingeniería del software, sino que combina fácilmente, con otras prácticas de ingeniería, metodologías de desarrollo o estándares. Se centra principalmente a nivel de las personas y equipo de desarrollo que construye el producto. El objetivo de la metodología Scrum es que los miembros que integran el equipo trabajen juntos y de forma eficaz obteniendo productos complejos y sofisticados [21].

Ilustración 5. Metodología SCRUM.



Fuente: [23]

Se puede interpretar que scrum es una clase de ingeniería social cuyo objetivo es el de poder satisfacer a todos los actores que participan en el desarrollo del producto. Obteniendo como resultado franqueza dentro del equipo, visibilidad del producto y evitando que puedan surgir posibles problemas que pongan en riesgo el desarrollo de software.

### Programación extrema (XP)

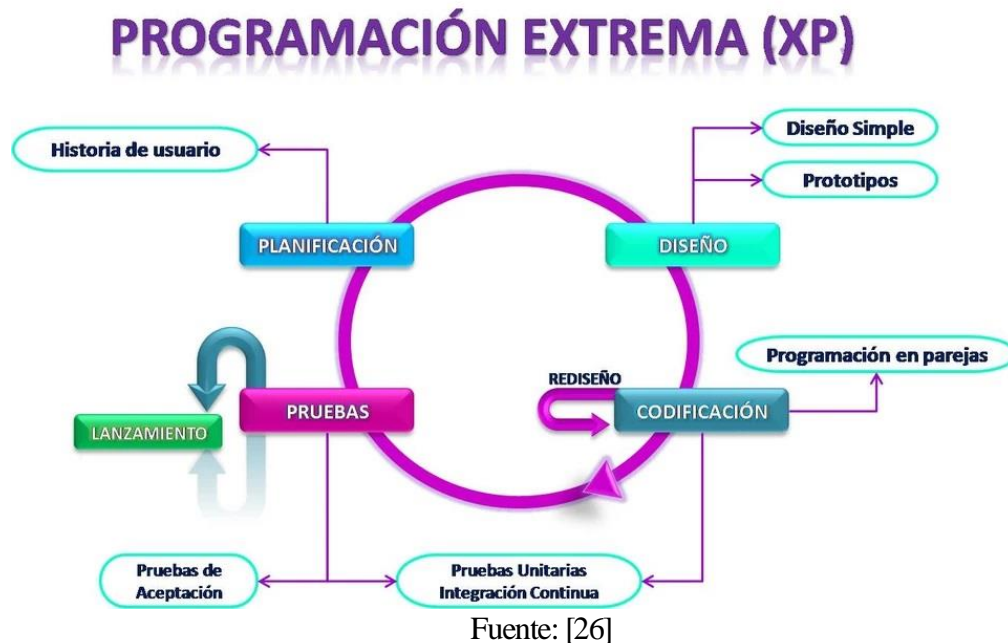
La programación extrema surge como nueva disciplina de desarrollo de software y ha generado un gran impacto en la comunidad de desarrolladores de software del mundo [24].

Esta metodología se basa en una sucesión de reglas y principios que se han ido generando a través de la historia de la ingeniería del software. Usadas en conjunto proporcionan una nueva metodología de desarrollo de software que se puede clasificar en las metodologías ligeras, en estas se da prioridad a las tareas que dan resultados directos [25]. Está basada en la simplicidad, la comunicación y el reciclado continuo de código, la programación extrema se puede clasificar como evolutiva.

Entre los principales objetivos de la programación extrema se encuentra la satisfacción del cliente esto tratando de dar al cliente el software que el necesita en el momento que

lo necesita, por ende se debe responder rápido a las necesidades del cliente y potenciar al máximo el trabajo en grupo de todos los actores que se encuentran involucrados en el proceso de desarrollo tales como los jefes del proyecto, los desarrolladores y los cliente [24].

Ilustración 6. Programación extrema



### 1.2.4.5 Node.js

Ilustración 7. Logo de Node.js



Fuente: [27]

Node.js fue ideado como un entorno de ejecución de JavaScript el cual está orientado a eventos asíncronos, con Node.js es posible crear aplicaciones escalables [28] de este modo permitiendo gestionar múltiples conexiones al mismo tiempo. Debido a esta característica no hay que preocuparse por el bloqueo de procesos ya que no existen bloqueos [29].



Node.js está influenciado y es similar en diseño a Event Machine de Ruby y Twisted de Python [28]. Node.js no tiene subprocesos sin embargo permite la utilización de múltiples núcleos en su entorno y compartir sockets entre procesos [29].

### Características de Node.js

**Velocidad:** Está construido sobre JavaScript V8 de Google Chrome, por lo cual su velocidad de ejecución es muy rápida [29].

**Asíncrono y contrato por eventos:** Node.js no espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarla lo que evita los bloqueos [29].

**Un subproceso escalable:** Debido al mecanismo de eventos que posee Node.js el servidor responde sin bloqueos lo que hace que el servidor sea altamente escalable [29].

#### 1.2.4.6 Base de datos MySQL

Ilustración 8. Logo de MySQL



Fuente: [30]

MySQL es una de las bases de datos más utilizadas a nivel mundial. Es una base de datos de código abierto, lo que significa que todo el mundo puede acceder al código fuente. Es un sistema de administración de bases de datos relacional [31].

MySQL es un sistema gestor de bases de datos que es desarrollado, distribuido y apoyado por Oracle [32].

## Características de MySQL.

- Escrito en C y C ++
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- Tablas hash en memoria temporales
- Es muy destacable su velocidad de respuesta.
- Brinda un sistema de contraseñas y privilegios
- Se puede utilizar incrustado en aplicaciones o como cliente-servidor.
- Soporta gran cantidad de datos. MySQL Server permite la creación de bases de datos de hasta 50 millones de registros.
- Total, soporte para operadores y funciones en cláusulas select y where.
- Es un motor de base de datos altamente confiable en cuanto a estabilidad se refiere.

### 1.2.4.7 Visual Studio Code

Ilustración 9. Logo de Visual Studio Code



Fuente: [33]

Visual Studio Code es un editor de código el cual permite trabajar con diversos lenguajes de programación, permite configurar tus propios atajos de teclado y refactorizar el código. Proporciona una gran utilidad puesto que permite descargar y gestionar extensiones con las que es posible personalizar y potenciar esta herramienta [34].

Las extensiones de visual studio code ayudan a tener una mayor satisfacción al momento de desarrollar aplicaciones debido a que es posible personalizar el IDE de desarrollo de forma que el usuario desee, permitiendo desde colorear determinado texto hasta montar un propio servidor dentro del IDE. También tiene extensiones que ayudan con los lenguajes de programación que se van a utilizar proporcionando atajos o auto completado.

### 1.2.4.8 GitHub

Ilustración 10. Logo de GitHub



Fuente: [35]

GitHub es un repositorio creado para alojar el código fuente de las aplicaciones de cualquier desarrollador, GitHub fue comprada por Microsoft en junio del 2018. La plataforma fue desarrollada con el fin de que los desarrolladores suban el código de sus aplicaciones y poder compartir el código para poder trabajar en conjunto con otros desarrolladores [36].

GitHub cuenta con un sistema de control de versiones Git que fue diseñado por Linus Torvalds. Este sistema de control de versiones permite a los desarrolladores administrar su proyecto, ordenando el código de cada una de las nuevas versiones que sacan de sus aplicaciones para evitar confusiones [36].

## 1.3 Objetivos

### 1.3.1 Objetivo General

Desarrollar una Aplicación Web Progresiva (PWA) para la gestión de procesos de cobro de la Junta de Agua Potable para la parroquia Huambaló caserío La Merced.

### 1.3.2 Objetivos Específicos

- Analizar los procesos relacionados a la gestión de cobros de la Junta de Agua Potable de la parroquia Huambaló caserío La Merced a fin de mejorar la toma de decisiones.
- Investigar el uso de diferentes tecnologías para crear una Aplicación Web Progresiva.
- Implantar la Aplicación Web Progresiva de gestión de procesos de cobro en la Junta de Agua Potable de la parroquia Huambaló caserío La Merced.

## **CAPÍTULO II**

### **2 METODOLOGÍA**

#### **2.1 Materiales**

##### **2.1.1 Institucionales**

- Repositorios Virtuales de la Universidad Técnica de Ambato.
- Instalaciones de la Junta de Agua Potable de la parroquia Huambaló caserío La Merced.

##### **2.1.2 Humano**

- Docente tutor de Tesis de la Universidad Técnica de Ambato.
- Personal administrativo de la Institución.
- Investigador.

##### **2.1.3 Otros**

- Suministros de oficina.
- Laptop.
- Internet.
- Visual Studio Code.
- Node.js.
- Base de datos MySQL.
- Impresora.
- Resma de hojas.

## 2.1.4 Económicos

Tabla 2. Cuadro de Presupuesto

No	Detalle	Unidad	Cant.	Valor Unitario	Valor total
1	Suministros de oficina	c/u	1	\$10.00	\$10.00
2	Laptop	c/u	1	\$1305.00	\$1305.00
3	Internet	c/mes	6	\$31.00	\$186.00
4	Visual Studio Code	c/u	1	\$0	\$0
5	Node.js	c/u	1	\$0	\$0
6	Base de datos MySQL	c/u	1	\$0	\$0
7	Impresora	c/u	1	\$300.00	\$300
8	Resma de hojas	c/u	1	\$5.00	\$5.00
				Sub Total	\$1806.00
				Imprevistos (10%)	\$180.60
				Total	\$1986.60

Elaborado por: El investigador

## 2.2 Métodos

### 2.2.1 Modalidad de la Investigación

Las modalidades de investigación que se manejarán para el desarrollo de la investigación serán Bibliográfica-Documental y de campo.

**Investigación Bibliográfica.** – La investigación será bibliográfica puesto que se utilizarán fuentes de artículos, documentos, tesis para la construcción del marco teórico.

**Investigación de campo.** – La investigación será de campo debido a que la investigación se realizará en la institución y de esta forma aportará información importante para el desarrollo del proyecto de investigación.

**Investigación Documental.** - La investigación será documental ya que se analizará el modelo de negocio para proponer una solución a la institución.

### 2.2.2 Población y Muestra

La población para la presente investigación está constituida de la siguiente manera:

Tabla 3. Población y Muestra

No	Tipo de Población	Descripción	Número
1	Presidente	Designado a dirigir la Junta de Agua Potable.	1
2	Vicepresidente	Ocupa un cargo inmediatamente inferior al cargo de presidente	1
3	Tesorero	Encargada de gestionar todo lo relacionado al movimiento de capital	1
4	Primer Vocal	Encargado de asistir a las Asambleas y sesiones de la Junta Ejecutiva con voz y voto	1
5	Segundo Vocal	Encargado de asistir a las Asambleas y sesiones de la Junta Ejecutiva con voz y voto	1
6	Recaudadora	Encargada de cobranzas de deudas.	1
		Total	6

Elaborado por: El investigador

Debido al tamaño de la población, no es necesario realizar una muestra ya que es menor o igual a 100 y se trabaja con toda la población establecida por el personal de la junta de agua potable de la parroquia Huambaló caserío La Merced.

### 2.2.3 Recolección de información

La técnica para emplearse será la entrevista dirigida específicamente a los miembros que conforman la junta de agua potable.

Tabla 4. Plan de recolección de la información

<b>Preguntas básicas</b>	<b>Explicación</b>
<b>¿Para qué?</b>	Para alcanzar los objetivos de la investigación
<b>¿De qué personas u objetos?</b>	Presidente Vicepresidente Tesorero Primer Vocal Segundo Vocal Recaudadora
<b>¿Sobre qué aspectos?</b>	Gestión de procesos de cobro
<b>¿Que, Quienes?</b>	Investigador: Alex Jair Villegas Alvarado
<b>¿Cuándo?</b>	septiembre 2021 – febrero 2022
<b>¿Donde?</b>	Junta de Agua Potable de la parroquia Huambaló caserío La Merced.
<b>¿Cuántas veces?</b>	Una
<b>¿Qué técnicas de recolección?</b>	Entrevista
<b>¿Con que?</b>	Cuestionario estructurado
<b>¿En qué situación?</b>	En un ambiente adecuado

Elaborado por: El investigador

#### **2.2.4 Procesamiento y análisis de datos**

Para la recolección y análisis de datos se realizará una reunión y posteriormente se procederá con una entrevista personal, específicamente con la Recaudadora, ya que es ella la persona que maneja los procesos de cobro. Con la ayuda de la metodología seleccionada se procederá con el desarrollo del proyecto.

### **2.2.5 Resultado de la entrevista**

Los datos adquiridos mediante la entrevista realizada a la recaudadora de la empresa, será analizada en base en la metodología XP en la fase I, debido a que se realiza el análisis de los datos para el desarrollo de software, realizando historias de usuario, para realizar los entregables en iteraciones.

### **2.2.6 Desarrollo del proyecto**

El proyecto se desarrollará en base a las fases de la metodología XP.

#### **Planificación**

- Levantamiento de información
- Análisis de resultados
- Definición de roles
- Modelado de procesos
- Historias de usuario
- Valoración de historias de usuario
- Estimación de historias de usuario
- Tareas
- Plan de entrega

#### **Diseño del proyecto**

- Metáfora de la aplicación
- Iteraciones

#### **Codificación**

- Desarrollo de las iteraciones establece sidas



## **Pruebas de Aceptación**

- Pruebas de Funcionalidad

## CAPÍTULO III

### 3 RESULTADOS Y DISCUSIÓN

#### 3.1 Análisis y discusión de los resultados

##### 3.1.1 Comparativa de metodologías ágiles

Tabla 5. Comparativa de metodologías ágiles

<b>Scrum</b>	<b>XP</b>
Es una metodología de desarrollo ágil basada en la administración del proyecto.	Es una metodología de desarrollo que está más centrada en la programación o creación del producto.
Cada miembro de del equipo trabaja de forma individual.	Los miembros del equipo programan en parejas.
Las iteraciones de entrega son de 1 a 4 semanas.	Las iteraciones de entrega son de 1 a 3 semanas
Lo que se termina, funciona y está bien, se aparta y ya no se toca	Se permite modificaciones durante el transcurso de todo el proyecto
Trata de seguir el orden de prioridad marcado por el producto owner en el sprint backlog, pero si es mejor modificar un sprint, es mejor modificar el orden de la prioridad	El equipo de desarrollo sigue estrictamente el orden de prioridad de las tareas definido por el cliente.
Recomendado para grandes empresas	Recomendado para pequeñas y grandes empresas

Elaborado por: El investigador

##### 3.1.2 Metodología de desarrollo seleccionada




Luego de analizar metodologías de desarrollo ágil en base a sus características. Teniendo en cuenta el uso de tecnologías PWA, se seleccionó el enfoque XP tomando en cuenta:

- Permite el rápido desarrollo de los proyectos al mantener una comunicación continua y duradera con las partes interesadas, logrando así un desarrollo óptimo e incremental al realizarse en serie. Esto facilita la retroalimentación y minimiza las tasas de error en el proyecto.

- Se enfoca en proyectos de desarrollo con grupos de trabajo pequeños y medianos, y basado en que los integrantes del grupo de desarrollo son pocos se consideró que la metodología XP es la más adecuada.
- Los tiempos de las iteraciones son cortos lo cual permite tener un mayor control con respecto a los avances realizados.

### 3.1.3 Comparativa de frameworks de desarrollo PWA

Tabla 6. Comparativa de frameworks de desarrollo PWA

<b>Características</b>	 <b>React</b>	 <b>Vue</b>	 <b>Angular</b>
<b>Modelo de Objetos del Documento (DOM)</b>	Virtual DOM	Virtual DOM	Regular DOM
<b>Línea de aprendizaje</b>	Lenta	Rápida	Rápida
<b>Paquete de componentes</b>	Fuerte	Fuerte	Fuerte
<b>Abstracción</b>	Medio	Medio	Fuerte
<b>Depuración General</b>	Buen HTML/Buen Js	Buen HTML/Medio Js	Buen HTML/Medio Js
<b>¿Etiqueta no cerrada mencionada?</b>	Si	Si	Si
<b>Enlace de datos</b>	Unidireccional, siendo los datos inmutables.	Bidireccional, donde los datos son mutables	Bidireccional, donde los datos son mutables
<b>Plantillas</b>	Si	Si	Si
<b>Modelo de componente</b>	Medio	Medio	Medio
<b>¿Compilación Móvil?</b>	React Native	Ionic Framework	Ionic Framework
<b>MVC</b>	vista de modelo	MVVM	Si
<b>Renderizado</b>	Lado del cliente	Lado del servidor	Lado del servidor
<b>Router</b>	React Router v4	vue-router	@angular/router

<b>Lenguaje</b>	JSX script, JS ES6+	JS,TypeScript	JS,TypeScript
<b>Componentes UI</b>	Material-UI Library	Vuetify	Material Desing
<b>Inyección de dependencias</b>	No soportado	Soportado	Soportado
<b>Comunidad</b>	Extensa	Extensa	Extensa
<b>Compañías</b>	Facebook, Twiter, Netflix, PayPal	Nintendo, Gitlab, Netlify	HBO, Apple, Adobe, Nike, Microsoft

Elaborado por: El investigador

### 3.1.4 Framework de desarrollo seleccionado

Después de analizar las características de los diferentes frameworks de programación, se seleccionó el framework Angular. Este framework cuenta con una extensa documentación técnica y una gran comunidad para resolver dudas y errores.

Con Angular se puede construir aplicaciones poderosas, escalables y rápidas, y la línea de aprendizaje es muy rápida, además cuenta con diversas librerías lo cual lo vuelve un framework potente al momento del desarrollo de aplicaciones.

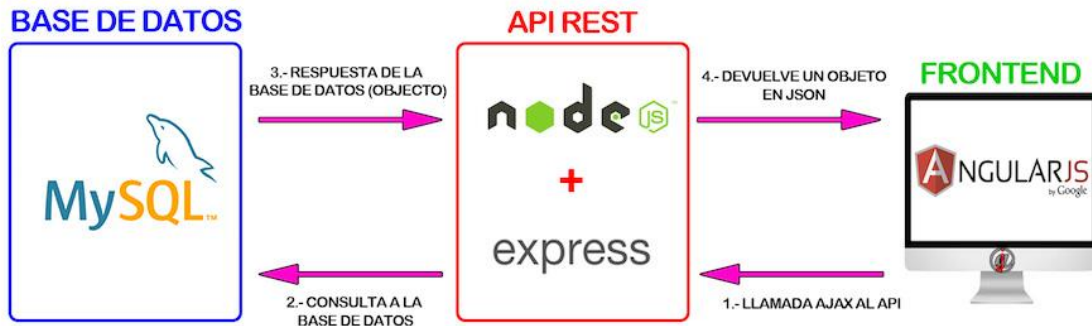
### 3.1.5 Determinación de la tecnología Back-End

La tecnología seleccionada para el presente proyecto de investigación será Node.js ya que es una tecnología relativamente fácil de utilizar y al mismo tiempo robusta, proporciona una gran escalabilidad para los proyectos ya que puede manejar un gran número de conexiones simultaneas además de ser rápido y eficiente.

### 3.1.6 Arquitectura de la aplicación

Para el desarrollo del proyecto de investigación se seleccionó la arquitectura cliente-servidor ya que en esta arquitectura las tareas se reparten entre el proveedor de los servicios (Node.js) y el cliente (PWA en Angular). Este modelo de arquitectura puede aplicarse a programas que se ejecutan en un mismo computador, aunque es más eficiente cuando se encuentra distribuido en diferentes computadores a través de una red.

Ilustración 11. Arquitectura Cliente-Servidor



Elaborado por: El investigador

## 3.2 Desarrollo de la propuesta

### 3.2.1 Fase I: Planificación

#### 3.2.1.1 Levantamiento de información

Tomando en cuenta que se cuenta con los recursos necesarios para el desarrollo de la aplicación web progresiva y una vez determinado que el proyecto es factible, se realizó una entrevista a la recaudadora de la Junta de agua potable de la parroquia Huambaló caserío La Merced la cual estuvo enfocada en las formas que utilizan para recolectar la información y los requerimientos necesarios para un buen desempeño de la aplicación. Por lo que se ha establecido con claridad cada uno de los requerimientos de los módulos a desarrollarse, con el fin de satisfacer las necesidades del usuario final.

Ilustración 12. Módulos definidos



Elaborado por: El investigador

### 3.2.1.2 Análisis de resultados

En la entrevista realizada a la recaudadora de la junta de agua potable parroquia Huambaló caserío La Merced, la Ing. Valeria Paredes, se pudo recolectar información tal como que la junta de agua potable maneja la mayor parte de la información de forma física y el sistema con el que actualmente cuenta es básico e ineficiente.

Por medio de la entrevista realizada también se pudo identificar los principales roles que regirán el sistema, de tal manera que existan jerarquías los cuales servirán para que cada usuario maneje de forma correcta sus respectivas funciones, los roles que se pudieron identificar son los siguiente:

- Administrador
- Recaudador
- Consultor

### 3.2.1.3 Definición de roles

Tabla 7. Definición de roles

Metodología	Rol	Responsable	Descripción
XP(Extreme Programming)	Usuario	Ing. Valeria Paredes	Definir los requerimientos y especificaciones de la Aplicación Web Progresiva
	Programador	Alex Villegas	Convertir los requerimientos en funcionalidades para el sistema, diseñar, programar, realizar pruebas unitarias de la PWA.
	Coach	Ing. Leonardo Torres	Verificar el cumplimiento de la planificación
	Tester	Ing. Valeria Paredes	Realizar las pruebas de aceptación

Elaborado por: El investigador

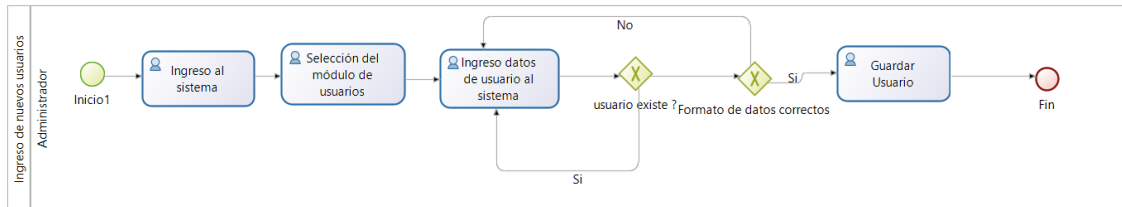
### 3.2.1.4 Modelado de procesos

En esta fase del proyecto se procederá a modelar los procesos que maneja la junta de agua potable de la parroquia Huambaló caserío La Merced.

## Procesos generales

Ingreso de usuarios al sistema:

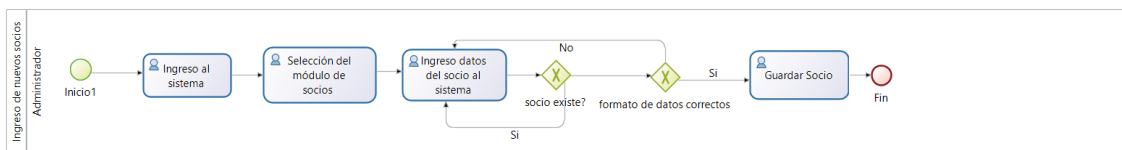
Ilustración 13. Ingreso de usuarios al sistema



Elaborado por: El investigador

Ingreso de socios al sistema

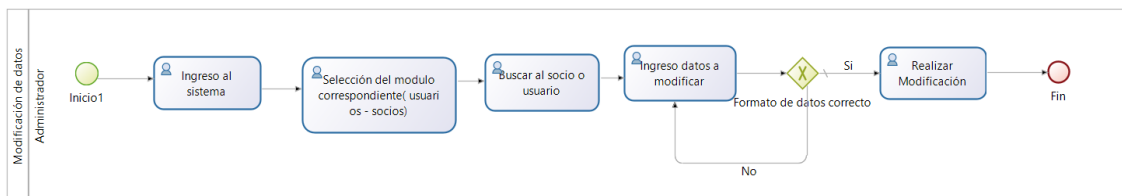
Ilustración 14. Ingreso de socios al sistema



Elaborado por: El investigador

Modificación de datos usuarios y socios:

Ilustración 15. Modificación de usuarios y socios

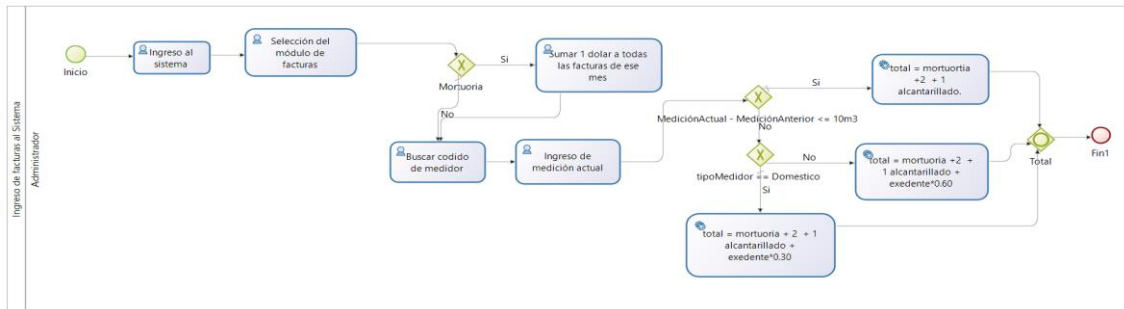


Elaborado por: El investigador



## Ingreso de mediciones al sistema

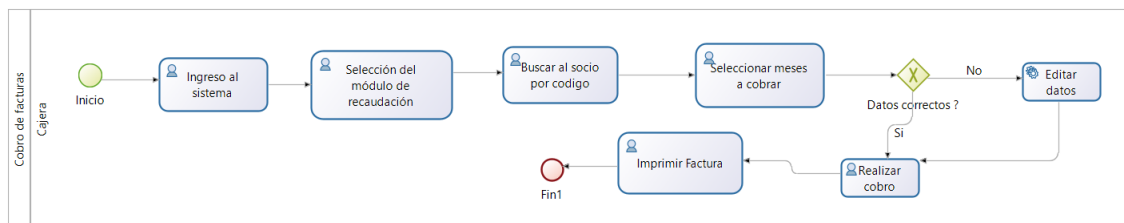
Ilustración 16. Ingreso de mediciones al sistema



Elaborado por: El investigador

## Cobro de facturas

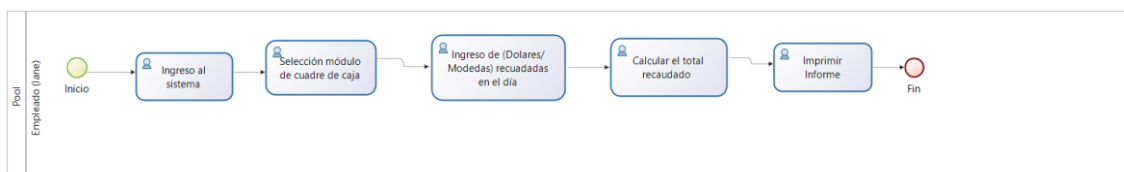
Ilustración 17. Cobro de facturas



Elaborado por: El investigador

## Cuadre de caja

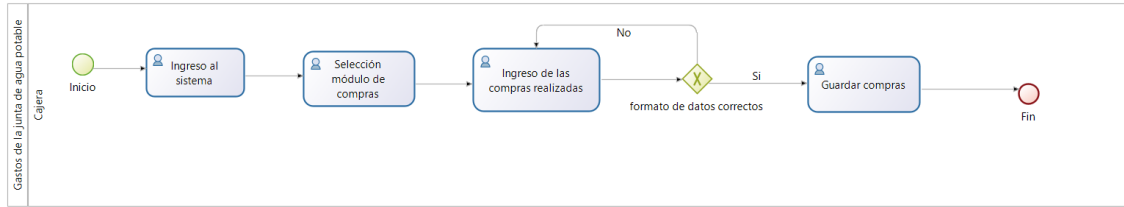
Ilustración 18. Cuadre de caja



Elaborado por: El investigador

## Compras

Ilustración 19. Compras



Elaborado por: El investigador

### 3.2.1.5 Historias de usuario

Es importante el realizar historias de usuario para llevar un control adecuado de cómo se está desarrollando el proyecto de desarrollo debido a que estas se especifica las funcionalidades que se implementaran el proyecto desde la perspectiva del usuario final.

Una vez realizadas las historias de usuario se procede a dividir las actividades para estimar el tiempo en que se cumplirán.

Tabla 8.Plantilla de historias de usuario

<b>Historia de usuario</b>	
<b>Identificación:</b>	<b>Usuario:</b>
<b>Nombre Historia:</b>	
<b>Prioridad en el negocio:</b>	<b>Riesgo en el desarrollo:</b>
<b>Puntos Estimados:</b>	<b>Iteración asignada:</b>
<b>Programador responsable:</b>	
<b>Descripción:</b>	
<b>Observación</b>	

Elaborado por: El investigador.

Esta plantilla está formada por los siguientes elementos:

**Identificador:** Es el identificador alfanumérico que se le asigna a cada historia usuario.

**Usuario:** Es la persona a la cual se hace referencia que redacta la historia.

**Nombre historia:** Es el título o nombre que se le da a la historia de usuario.

**Prioridad en el negocio:** Valores que se asignan de acuerdo a la necesidad del usuario final (Alta, Media y Baja).

**Riesgo en el desarrollo:** Valores que se asignan de acuerdo al riesgo que pueda tener al desarrollar la historia de usuario.

**Puntos estimados:** Es el número de días estimado en el que se desarrollará la historia de usuario.

**Iteración asignada:** Es el número de iteración en la que se establece la historia de usuario.

**Programador responsable:** Nombre de la persona encargada de desarrollar la historia de usuario.

**Descripción:** El usuario detalla con sus propias palabras lo que se va a hacer mas no el cómo hacer.

**Observación:** Se detalla los diferentes procesos que tiene relación con la historia de usuario.

## Historias de usuario

Tabla 9. Historia de usuario - Estructura del proyecto

<b>Historia de usuario</b>	
<b>Identificación:</b> H001	<b>Usuario:</b> Desarrollador
<b>Nombre Historia:</b> Estructurar el proyecto	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 5	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Establecer la estructura del proyecto en base al framework a utilizar (Angular)	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 10. Historia de usuario - Diseño de la base de datos

<b>Historia de usuario</b>	
<b>Identificación:</b> H002	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de la base de datos	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 4	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseñar la base de datos	
<b>Observación:</b> La base de datos se diseñará en MySql	

Elaborado por: El investigador

Tabla 11. Historia de usuario - Diseño de interfaz módulo usuarios

<b>Historia de usuario</b>	
<b>Identificación:</b> H003	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo usuarios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 1
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 12. Historia de usuario - Diseño de interfaz módulo socios

<b>Historia de usuario</b>	
<b>Identificación:</b> H004	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo socios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 13. Historia de usuario - Diseño de interfaz módulo lecturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H005	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo lecturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 14. Historia de usuario - Diseño de interfaz módulo de facturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H006	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo facturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 4	<b>Iteración asignada:</b> 2
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 15. Historia de usuario - Diseño de interfaz módulo recaudación

<b>Historia de usuario</b>	
<b>Identificación:</b> H007	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo recaudación	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 4	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 16. Historia de usuario - Diseño interfaz modulo mortuoria

<b>Historia de usuario</b>	
<b>Identificación:</b> H008	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo mortuoria	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 17. Historia de usuario - Diseño de interfaz módulo cuadro de caja

<b>Historia de usuario</b>	
<b>Identificación:</b> H009	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo cuadro de caja	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 18. Historia de usuario - Diseño de interfaz módulo compras

<b>Historia de usuario</b>	
<b>Identificación:</b> H010	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo compras	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 3
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 19. Historia de usuario - Diseño de interfaz módulo reportes

<b>Historia de usuario</b>	
<b>Identificación:</b> H011	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Diseño de interfaz módulo reportes	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 20. Historia de usuario - Inicio de sesión

<b>Historia de usuario</b>	
<b>Identificación:</b> H012	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Inicio de sesión	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 4	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> El usuario del sistema podrá acceder mediante un usuario y una contraseña	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 21. Historia de usuarios - Listado de usuarios

<b>Historia de usuario</b>	
<b>Identificación:</b> H013	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listado de usuarios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar los usuarios del sistema.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 22. Historia de usuario - Creación de usuarios

<b>Historia de usuario</b>	
<b>Identificación:</b> H014	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Creación de usuarios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 4
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir crear usuarios en el sistema.	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 23. Historia de usuarios - Edición de usuarios

<b>Historia de usuario</b>	
<b>Identificación:</b> H015	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Edición de usuarios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir editar los datos del usuario	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador



Tabla 24. Historia de usuario - Habilitar y deshabilitar de usuarios

<b>Historia de usuario</b>	
<b>Identificación:</b> H016	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Habilitar y deshabilitar usuarios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir habilitar y deshabilitar los usuarios del sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 25. Historia de usuario - Listar socios

<b>Historia de usuario</b>	
<b>Identificación:</b> H017	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listar socios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir listar los socios del sistema	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 26. Historia de usuario - Crear socio

<b>Historia de usuario</b>	
<b>Identificación:</b> H018	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Crear socio	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir crear socios	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 27. Historia de usuario - Editar socios

<b>Historia de usuario</b>	
<b>Identificación:</b> H019	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Editar socios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir editar socios del sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 28. Historia de usuario - Habilitar y deshabilitar socios

<b>Historia de usuario</b>	
<b>Identificación:</b> H020	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Habilitar y deshabilitar socios	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 5
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir habilitar y deshabilitar los socios del sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 29. Historia de usuario - Listar lecturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H021	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listar lecturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir listar lecturas del sistema	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 30. Historia de usuario - Agregar lecturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H022	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Agregar lecturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir agregar lecturas del sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador y recaudador.	

Elaborado por: El investigador

Tabla 31. Historia de usuario - Editar lecturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H023	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Editar lecturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir editar lecturas del sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 32. Historia de usuario - Eliminar lecturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H024	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Eliminar lecturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir eliminar lecturas del sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 33. Historia de usuario - Listar Facturas

<b>Historia de usuario</b>	
<b>Identificación:</b> H025	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listar facturas	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir eliminar facturas del sistema	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 34. Historia de usuario - Cobro de factura

<b>Historia de usuario</b>	
<b>Identificación:</b> H026	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Cobro de factura	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 5	<b>Iteración asignada:</b> 6
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir realizar el cobro de las lecturas que se encuentran registradas en el sistema.	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador y recaudador.	

Elaborado por: El investigador

Tabla 35. Historia de usuario - Listar mortuoria

<b>Historia de usuario</b>	
<b>Identificación:</b> H027	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listar mortuoria	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 7
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir listar el registro de mortuoria registrado en el sistema	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 36. Historia de usuario - Agregar mortuoria

<b>Historia de usuario</b>	
<b>Identificación:</b> H028	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Agregar mortuoria	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 7
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir agregar registros de mortuoria al sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador y recaudador.	

Elaborado por: El investigador

Tabla 37. Historia de usuario - Eliminar mortuoria

<b>Historia de usuario</b>	
<b>Identificación:</b> H029	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Eliminar mortuoria	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 7
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir eliminar registros de mortuoria al sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 38. Historia de usuario – Listar cuadro de caja

<b>Historia de usuario</b>	
<b>Identificación:</b> H030	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listar cuadro de caja	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 7
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir listar los cuadros de caja registrados en el sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador y recaudador.	

Elaborado por: El investigador

Tabla 39. Historia de usuario - Agregar cuadro de caja

<b>Historia de usuario</b>	
<b>Identificación:</b> H031	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Agregar cuadro de caja	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 7
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir realizar el cuadro de caja ingresando la cantidad por cada billete o dólar recaudado.	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador y recaudador.	

Elaborado por: El investigador

Tabla 40. Historia de usuario - Listar compras

<b>Historia de usuario</b>	
<b>Identificación:</b> H032	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Listar compras	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 8
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir listar las compras registradas en el sistema.	
<b>Observación:</b>	

Elaborado por: El investigador

Tabla 41. Historia de usuario - Agregar compra

<b>Historia de usuario</b>	
<b>Identificación:</b> H033	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Agregar compra	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 8
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir agregar cuadros de caja registrados en el sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador y recaudador.	

Elaborado por: El investigador

Tabla 42. Historia de usuario - Editar compra

<b>Historia de usuario</b>	
<b>Identificación:</b> H034	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Editar compra	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 8
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir editar los cuadros de caja registrados en el sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 43. Historia de usuario - Eliminar compra

<b>Historia de usuario</b>	
<b>Identificación:</b> H035	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Eliminar compra	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 2	<b>Iteración asignada:</b> 8
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir eliminar los cuadros de caja registrados en el sistema	
<b>Observación:</b> Esta funcionalidad solo estará disponible para el usuario administrador.	

Elaborado por: El investigador

Tabla 44. Historia de usuario - Generación de reportes

<b>Historia de usuario</b>	
<b>Identificación:</b> H036	<b>Usuario:</b> Usuario Final
<b>Nombre Historia:</b> Generación de reportes	
<b>Prioridad en el negocio:</b> Alta	<b>Riesgo en el desarrollo:</b> Alto
<b>Puntos Estimados:</b> 7	<b>Iteración asignada:</b> 8
<b>Programador responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de las facturaciones realizadas	
<b>Observación:</b>	

Elaborado por: El investigador

### 3.2.1.6 Valoración de las historias de usuario

Una vez detallados los requerimientos establecidos en las historias de usuario, tendrán valoraciones definidas para los cuales se consideró dedicar 5 horas de trabajo por día laborable.

### 3.2.1.7 Estimación de historias de usuario

Tabla 45. Estimación de historias de usuario

N°	Historia de usuario	Tiempo estimado	
		Días	Horas
H001	Estructurar el proyecto	5	25
H002	Diseño de la base de datos	4	20
H003	Diseño de interfaz modulo usuarios	3	15
H004	Diseño de interfaz modulo socios	3	15
H005	Diseño de interfaz modulo lecturas	3	15
H006	Diseño de interfaz modulo facturas	4	20
H007	Diseño de interfaz modulo recaudación	4	20
H008	Diseño de interfaz modulo mortuoria	3	15
H009	Diseño de interfaz modulo cuadro de caja	3	15
H010	Diseño de interfaz modulo compras	3	15
H011	Diseño de interfaz modulo reportes	3	15
H012	Inicio de sesión	4	20
H013	Listado de usuarios	2	10
H014	Creación de usuarios	2	10
H015	Edición de usuarios	2	10
H016	Habilitar y deshabilitar usuarios	2	10
H017	Listar socios	2	10
H018	Crear socio	2	10
H019	Editar socios	2	10
H020	Habilitar y deshabilitar socios	2	10
H021	Listas lecturas	2	10
H022	Agregar lecturas	2	10
H023	Editar lecturas	2	10
H024	Eliminar lecturas	2	10
H025	Listar facturas	2	10
H026	Cobro de factura	5	25
H027	Listar mortuoria	2	10
H028	Agregar mortuoria	2	10
H029	Eliminar mortuoria	2	10
H030	Listar cuadro de caja	2	10
H031	Agregar cuadro de caja	2	10
H032	Listar compras	2	10
H033	Agregar compra	2	10
H034	Editar compra	2	10
H035	Eliminar compra	2	10
H036	Generación de reportes	7	35
<b>Total:</b>		<b>98</b>	<b>490</b>

Elaborado por: El investigador



### 3.2.1.8 Tareas

Acorde a la metodología XP se realizarán tareas a partir de las historias de usuario, estas tareas están delimitadas por una fecha de inicio, así como también por un responsable el cual estará encargado de completarla en las fechas establecidas.

Tabla 46. Plantilla de tareas de historias de usuario

<b>TAREA</b>	
<b>Identificador de a tarea:</b>	<b>Identificador de la historia:</b>
<b>Denominación de la tarea:</b>	
<b>Tipo de tarea:</b>	<b>Puntos estimados:</b>
<b>Fecha inicio:</b>	<b>Fecha fin:</b>
<b>Responsable:</b>	
<b>Descripción:</b>	

Elaborado por: El investigador

Tabla 47. Tarea - Estructurar el proyecto

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T001	<b>Identificador de la historia:</b> H001
<b>Denominación de la tarea:</b> Estructurar el proyecto	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5
<b>Fecha inicio:</b> 04/04/2022	<b>Fecha fin:</b> 08/04/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Establecer la estructura del proyecto en base al framework a utilizar (Angular)	

Elaborado por: El investigador

Tabla 48. Tarea - Diseño de la base de datos

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T002	<b>Identificador de la historia:</b> H002
<b>Denominación de la tarea:</b> Diseño de la base de datos	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Fecha inicio:</b> 11/04/2022	<b>Fecha fin:</b> 14/04/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseñar la base de datos	

Elaborado por: El investigador

Tabla 49. Tarea - Diseño de interfaz módulo usuarios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T003	<b>Identificador de la historia:</b> H003
<b>Denominación de la tarea:</b> Diseño de interfaz módulo usuarios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 15/04/2022	<b>Fecha fin:</b> 18/04/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 50. Tarea - Diseño de interfaz módulo socios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T004	<b>Identificador de la historia:</b> H004
<b>Denominación de la tarea:</b> Diseño de interfaz módulo socios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 19/04/2022	<b>Fecha fin:</b> 21/04/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 51. Tarea - Diseño de interfaz módulo de lecturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T005	<b>Identificador de la historia:</b> H005
<b>Denominación de la tarea:</b> Diseño de interfaz módulo lecturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 22/04/2022	<b>Fecha fin:</b> 26/04/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 52. Tarea - Diseño de interfaz módulo facturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T006	<b>Identificador de la historia:</b> H006
<b>Denominación de la tarea:</b> Diseño de interfaz módulo facturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Fecha inicio:</b> 27/04/2022	<b>Fecha fin:</b> 02/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 53. Tarea - Diseño de interfaz módulo recaudación

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T007	<b>Identificador de la historia:</b> H007
<b>Denominación de la tarea:</b> Diseño de interfaz módulo recaudación	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Fecha inicio:</b> 03/05/2022	<b>Fecha fin:</b> 06/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 54. Tarea - Diseño de interfaz módulo cuadro de caja

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T008	<b>Identificador de la historia:</b> H008
<b>Denominación de la tarea:</b> Diseño de interfaz módulo de mortuoria	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 09/05/2022	<b>Fecha fin:</b> 11/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 55. Tarea - Diseño de interfaz módulo cuadro de caja

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T009	<b>Identificador de la historia:</b> H009
<b>Denominación de la tarea:</b> Diseño de interfaz módulo cuadro de caja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 12/05/2022	<b>Fecha fin:</b> 16/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 56. Tarea - Diseño de interfaz módulo compras

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T010	<b>Identificador de la historia:</b> H010
<b>Denominación de la tarea:</b> Diseño de interfaz módulo compras	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 17/05/2022	<b>Fecha fin:</b> 19/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 57. Tarea - Diseño de interfaz módulo reportes

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T011	<b>Identificador de la historia:</b> H011
<b>Denominación de la tarea:</b> Diseño de interfaz módulo reportes	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 20/05/2022	<b>Fecha fin:</b> 24/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Diseño de interfaces que tendrá el sistema para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 58. Tarea - Diseño de interfaz inicio de sesión

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T012	<b>Identificador de la historia:</b> H012
<b>Denominación de la tarea:</b> Diseño de interfaz inicio de sesión	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha inicio:</b> 25/05/2022	<b>Fecha fin:</b> 27/05/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Desarrollo backend inicio de sesión y seguridad	

Elaborado por: El investigador

Tabla 59. Tarea - Listado de usuarios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T013	<b>Identificador de la historia:</b> H013
<b>Denominación de la tarea:</b> Listado de usuarios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 28/05/2022	<b>Fecha fin:</b> 29/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Interfaz listar usuarios	

Elaborado por: El investigador

Tabla 60. Tarea - Creación de usuarios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T014	<b>Identificador de la historia:</b> H014
<b>Denominación de la tarea:</b> Creación de usuarios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 30/05/2022	<b>Fecha fin:</b> 01/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Interfaz para crear usuarios	

Elaborado por: El investigador

Tabla 61. Tarea - Edición de usuarios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T015	<b>Identificador de la historia:</b> H015
<b>Denominación de la tarea:</b> Edición de usuarios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 04/06/2022	<b>Fecha fin:</b> 05/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Interfaz para editar usuarios	

Elaborado por: El investigador

Tabla 62. Tarea - Habilitar y deshabilitar usuarios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T016	<b>Identificador de la historia:</b> H016
<b>Denominación de la tarea:</b> Habilitar y deshabilitar usuarios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 06/06/2022	<b>Fecha fin:</b> 07/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Habilitar y deshabilitar usuarios	

Elaborado por: El investigador

Tabla 63. Tarea - Listar socios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T017	<b>Identificador de la historia:</b> H017
<b>Denominación de la tarea:</b> Listar socios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 08/06/2022	<b>Fecha fin:</b> 11/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar socios	

Elaborado por: El investigador

Tabla 64. Tarea - Crear socio

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T018	<b>Identificador de la historia:</b> H018
<b>Denominación de la tarea:</b> Crear socio	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 12/06/2022	<b>Fecha fin:</b> 12/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Crear socios	

Elaborado por: El investigador

Tabla 65. Tarea - Editar socios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T019	<b>Identificador de la historia:</b> H019
<b>Denominación de la tarea:</b> Editar socios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 13/06/2022	<b>Fecha fin:</b> 14/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Editar socios	

Elaborado por: El investigador

Tabla 66. Tarea - Habilitar y deshabilitar socios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T020	<b>Identificador de la historia:</b> H020
<b>Denominación de la tarea:</b> Habilitar y deshabilitar socios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 15/06/2022	<b>Fecha fin:</b> 16/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Habilitar y deshabilitar socios	

Elaborado por: El investigador

Tabla 67. Tarea - Listar lecturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T021	<b>Identificador de la historia:</b> H021
<b>Denominación de la tarea:</b> Listar lecturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 17/06/2022	<b>Fecha fin:</b> 20/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar lecturas	

Elaborado por: El investigador

Tabla 68. Tarea - Agregar lecturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T022	<b>Identificador de la historia:</b> H022
<b>Denominación de la tarea:</b> Agregar lecturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 21/06/2022	<b>Fecha fin:</b> 22/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Agregar lecturas	

Elaborado por: El investigador

Tabla 69. Tarea - Editar lecturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T023	<b>Identificador de la historia:</b> H023
<b>Denominación de la tarea:</b> Editar lecturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 23/06/2022	<b>Fecha fin:</b> 24/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Editar lecturas	

Elaborado por: El investigador



Tabla 70. Tarea - Eliminar lecturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T024	<b>Identificador de la historia:</b> H024
<b>Denominación de la tarea:</b> Eliminar lecturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 27/06/2022	<b>Fecha fin:</b> 28/06/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Eliminar lecturas	

Elaborado por: El investigador

Tabla 71. Tarea - Listar facturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T025	<b>Identificador de la historia:</b> H025
<b>Denominación de la tarea:</b> Listar facturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 29/06/2022	<b>Fecha fin:</b> 30/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar facturas	

Elaborado por: El investigador

Tabla 72. Tarea - Cobro de facturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T026	<b>Identificador de la historia:</b> H026
<b>Denominación de la tarea:</b> Cobro de facturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5
<b>Fecha inicio:</b> 01/07/2022	<b>Fecha fin:</b> 07/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir realizar el cobro de las lecturas que se encuentran registradas en el sistema.	

Elaborado por: El investigador

Tabla 73. Tarea - Listar mortuoria

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T027	<b>Identificador de la historia:</b> H027
<b>Denominación de la tarea:</b> Listar mortuoria	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 08/07/2022	<b>Fecha fin:</b> 11/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar mortuoria.	

Elaborado por: El investigador

Tabla 74. Tarea - Agregar mortuoria

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T028	<b>Identificador de la historia:</b> H028
<b>Denominación de la tarea:</b> Agregar mortuoria	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 12/07/2022	<b>Fecha fin:</b> 13/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Agregar mortuoria	

Elaborado por: El investigador

Tabla 75. Tarea - Eliminar mortuoria

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T029	<b>Identificador de la historia:</b> H029
<b>Denominación de la tarea:</b> Eliminar mortuoria	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 14/07/2022	<b>Fecha fin:</b> 15/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Eliminar mortuoria	

Elaborado por: El investigador

Tabla 76. Tarea – Listar cuadro de caja

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T030	<b>Identificador de la historia:</b> H030
<b>Denominación de la tarea:</b> Listar cuadro de caja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 18/07/2022	<b>Fecha fin:</b> 19/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar cuadro de caja	

Elaborado por: El investigador

Tabla 77. Tarea – Agregar cuadro de caja

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T031	<b>Identificador de la historia:</b> H031
<b>Denominación de la tarea:</b> Agregar cuadro de caja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 20/07/2022	<b>Fecha fin:</b> 21/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir realizar el cuadro de caja ingresando la cantidad por cada billete o dólar recaudado	

Elaborado por: El investigador

Tabla 78. Tarea – Listar compras

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T032	<b>Identificador de la historia:</b> H032
<b>Denominación de la tarea:</b> Listar compras	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 22/07/2022	<b>Fecha fin:</b> 25/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Listar compras	

Elaborado por: El investigador

Tabla 79. Tarea – Agregar compra

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T033	<b>Identificador de la historia:</b> H023
<b>Denominación de la tarea:</b> Agregar compra	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 26/07/2022	<b>Fecha fin:</b> 27/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Agregar compra	

Elaborado por: El investigador

Tabla 80. Tarea – Editar compra

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T034	<b>Identificador de la historia:</b> H034
<b>Denominación de la tarea:</b> Editar compra	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 28/07/2022	<b>Fecha fin:</b> 29/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Editar compra	

Elaborado por: El investigador

Tabla 81. Tarea – Eliminar compra

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T035	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Eliminar compra	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha inicio:</b> 01/08/2022	<b>Fecha fin:</b> 02/08/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Eliminar compra	

Elaborado por: El investigador

Tabla 82. Tarea – Reportes Socios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T036	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes Socios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 03/08/2022	<b>Fecha fin:</b> 03/08/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de los socios	

Elaborado por: El investigador

Tabla 83. Tarea - Reportes usuarios

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T037	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes Usuarios	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 04/08/2022	<b>Fecha fin:</b> 04/08/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de los usuarios.	

Elaborado por: El investigador

Tabla 84. Tarea - Reporte lecturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T038	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes lecturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 05/08/2022	<b>Fecha fin:</b> 05/8/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de las lecturas ingresadas al sistema	

Elaborado por: El investigador

Tabla 85. Tarea – Reportes facturas

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T039	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes Facturas	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 08/07/2022	<b>Fecha fin:</b> 08/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de las facturaciones realizadas	

Elaborado por: El investigador

Tabla 86. Tarea - Reportes mortuoria

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T040	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes mortuoria	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 09/07/2022	<b>Fecha fin:</b> 09/07/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes mortuoria	

Elaborado por: El investigador

Tabla 87. Tarea – Reportes cuadro de caja

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T041	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes cuadro de caja	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 10/08/2022	<b>Fecha fin:</b> 10/08/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de cuadro de caja	

Elaborado por: El investigador

Tabla 88. Tarea - Reportes compras

<b>TAREA</b>	
<b>Identificador de a tarea:</b> T042	<b>Identificador de la historia:</b> H036
<b>Denominación de la tarea:</b> Reportes compras	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1
<b>Fecha inicio:</b> 11/08/2022	<b>Fecha fin:</b> 11/08/2022
<b>Responsable:</b> Alex Jair Villegas Alvarado	
<b>Descripción:</b> Permitir generar reportes de las compras realizadas	

Elaborado por: El investigador

### 3.2.1.9 Plan de entrega

El objetivo de crear un plan de entrega fue es establecer un tiempo de finalización para cada historia de usuario aplicando el enfoque XP teniendo en cuenta que cada iteración tendría una duración de 2 semanas y media.

Tabla 89. Planificación de entrega

N°	Historia de usuario	Tiempo estimado		Iteración asignada									
		Días	Horas	1	2	3	4	5	6	7	8		
H001	Estructurar el proyecto	5	25	x									
H002	Diseño de la base de datos	4	20	x									
H003	Diseño de interfaz modulo usuarios	3	15	x									
H004	Diseño de interfaz modulo socios	3	15		x								
H005	Diseño de interfaz modulo lecturas	3	15		x								
H006	Diseño de interfaz modulo facturas	4	20		x								
H007	Diseño de interfaz modulo recaudación	4	20			x							
H008	Diseño de interfaz modulo mortuoria	3	15			x							
H009	Diseño de interfaz modulo cuadro de caja	3	15			x							
H010	Diseño de interfaz modulo compras	3	15			x							
H011	Diseño de interfaz modulo reportes	3	15				x						
H012	Inicio de sesión	4	20				x						
H013	Listado de usuarios	2	10				x						
H014	Creación de usuarios	2	10				x						
H015	Edición de usuarios	2	10					x					
H016	Habilitar y deshabilitar usuarios	2	10					x					
H017	Listar socios	2	10					x					
H018	Crear socio	2	10					x					
H019	Editar socios	2	10					x					
H020	Habilitar y deshabilitar socios	2	10					x					
H021	Listas lecturas	2	10							x			
H022	Agregar lecturas	2	10							x			
H023	Editar lecturas	2	10							x			
H024	Eliminar lecturas	2	10							x			
H025	Listar facturas	2	10							x			
H026	Cobro de factura	5	25							x			
H027	Listar mortuoria	2	10									x	
H028	Agregar mortuoria	2	10									x	
H029	Eliminar mortuoria	2	10									x	
H030	Listar cuadro de caja	2	10									x	
H031	Agregar cuadro de caja	2	10									x	
H032	Listar compras	2	10										x
H033	Agregar compra	2	10										x
H034	Editar compra	2	10										x
H035	Eliminar compra	2	10										x
H036	Generación de reportes	7	35										x

Elaborado por: El investigador



### **3.2.2 Fase II: Diseño del proyecto**

Una vez recolectados los datos los cuales indican las necesidades del usuario, el objetivo de la segunda fase de la metodología XP se centra en servir de guía y dar soporte para iniciar con la codificación del proyecto.

#### **3.2.2.1 Metáfora de la aplicación**

En la metodología de desarrollo XP, diseñar un proyecto implica desarrollar una metáfora de aplicación. En este caso la aplicación se ha dividido en diferentes módulos en base a la información recolectada. Se procede a elaborar una metáfora para cada módulo porque es imposible dar una metáfora para todo el proyecto puesto que está compuesto por diversas secciones.

Para el módulo de usuarios permitirá gestionar los usuarios que utilicen la aplicación, así como también asignarles determinados roles, también permitirá añadir, editar y eliminar los usuarios. Estas funciones estarán disponibles únicamente para el usuario administrador.

El módulo de socios permitirá gestionar los socios que estén registrados en el sistema, así como también añadir, editar y eliminar a los socios. Esta función solo estará disponible para el usuario administrador.

El siguiente módulo corresponde a las lecturas se permitirá el ingreso de las lecturas mensuales de todos los socios que se encuentren registrados en el sistema, también permitirá editar las mismas y su eliminación. Esta función solo estará disponible para el usuario administrador y recaudador.

El módulo correspondiente a las facturas permitirá filtrar las facturas por socios y fechas de pago así mismo como revisar el detalle de cada una de las facturas.

En el módulo de recaudación permitirá cambiar el estado de las lecturas previamente ingresados al sistema una vez se halla cobrado el valor pendiente.

A continuación, en módulo cuadro de caja dará la facilidad al usuario de imprimir un reporte de la recaudación realizada en el día, además de permitirle ingresar el valor de los billetes y monedas para realizar la correspondiente suma la cual tiene que ser igual al valor que consta en el reporte de la recaudación diaria.

El módulo de mortuoria permitirá ingresar las contribuciones a mortuoria, esta contribución servirá para cuando un socio de la junta fallezca colaborar para los gastos funerarios.

También se cuenta con el módulo de compras en este módulo se realizará el ingreso de las facturas de los gastos realizados por la junta de agua potable para así a fin de mes poder imprimir un reporte con los ingresos y los egresos que a realizado la junta de agua potable.

El ultimo módulo de la aplicación es el de reportes en cuyo módulo se generan los reportes tanto de usuarios y socios. También se podrá realizar la impresión de las facturas entre determinadas fechas, de los socios que lleven atraso en el pago de sus facturas, reporte de los ingresos y egresos de la junta de agua potable, etc.

### 3.2.2.2 Iteración I

Tabla 90. Estimación de la primera iteración

Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
1	H001	Estructurar el proyecto	5	25
1	H002	Diseño de la base de datos	4	20
1	H003	Diseño de interfaz módulo usuarios	3	15
Total:			12	60

Elaborado por: El investigador

## Establecer la estructura del proyecto

### Hardware

- Laptop Dell G5 SE
- Procesador AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz
- Memoria RAM 16,0 GB

### Software

- Sistema operativo de 64 bits, procesador basado en x64
- Angular CLI: 13.3.1
- Node: 14.15.0
- Package Manager: npm 6.14.8
- Paquetes de Node: cors, express, fs-extra, jsonwebtoken, morgan, multer, promise-mysql, uuid, bcryptjs, class-validator, helmet, mysql, typeorm
- Editor de código fuente Visual Studio Code.
- Extensiones para Visual Studio Code: Bracket Pair Colorizer 2, Angular Language Service, Angular Snippets (Version 13), Beautify, JSON to TS, Material Icon Theme, mysql-inline-decorator, Spanish Language Pack for Visual Studio Code

Una vez determinado el ambiente de desarrollo se procede con el desarrollo del mismo:

### Creación del proyecto

Se ingresa el comando **ng new (nombre del proyecto)** para crear un nuevo proyecto, una vez realizado esto preguntara si desea trabajar con rutas y como último se debe seleccionar el tipo de archivos CSS con el que se va a trabajar en el proyecto.

## Ilustración 20. Inicialización del proyecto

```
C:\Users\Alex\Desktop\Trabajos\Angular>ng new JAP
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS [ https://sass-lang.com/documentation/syntax#scss ]
Sass [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less [ http://lesscss.org ]
```

Elaborado por: El investigador

Entrar a la carpeta del proyecto que se creó y levantar el servidor de Angular que usará el puerto 4200 por defecto siendo este modificable.

## Ilustración 21. Levantar el servidor de Angular

```
C:\Users\Alex\Desktop\Trabajos\Angular>cd JAP
C:\Users\Alex\Desktop\Trabajos\Angular\JAP>ng serve -o
```

Elaborado por: El investigador

## Organización de directorios del proyecto

De acuerdo al framework seleccionado Angular, este ya define una estructura básica del proyecto, pero se agregaron nuevos directorios para la correcta estructuración del proyecto los cuales son:

**Layouts:** En este directorio se encontrarán declarados los componentes de la aplicación y sus respectivas rutas.

**Services:** En este directorio se encontrarán los servicios de la aplicación los cuales servirán para comunicarse con el Backend.

**Services/Guards:** Aquí se protegerán las rutas a las cuales solo una persona que se encuentre autenticada puede ingresar.

**Shared:** En este directorio se encontrarán los componentes compartidos que van a ser utilizados en diferentes pantallas.

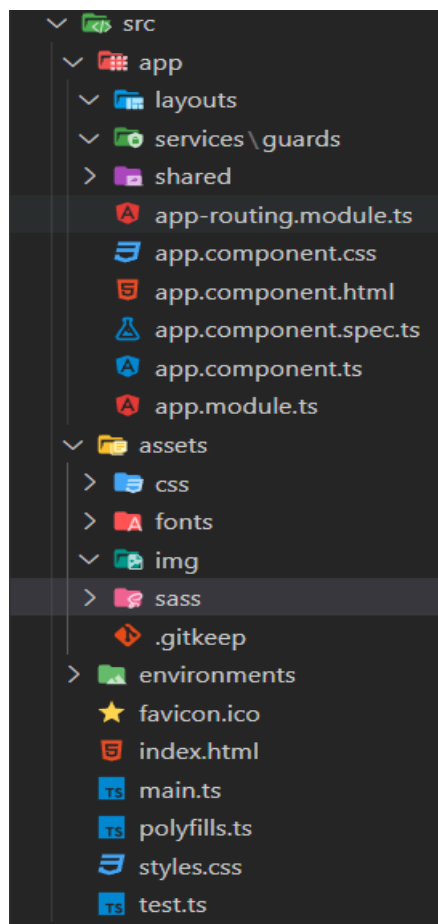
**Assets/css:** En donde se encontrarán los estilos css que se aplicarán de forma global.

**Assets/fonst:** En este directorio se guardarán los tipos de letra que serán utilizados en la aplicación.

**Assets/img:** En donde se encontrarán las imágenes que serán utilizadas en la aplicación.

**Assets/sass:** En donde se encontrarán los estilos en cascada que se aplicarán de forma global.

Ilustración 22. Organización de directorios del proyecto

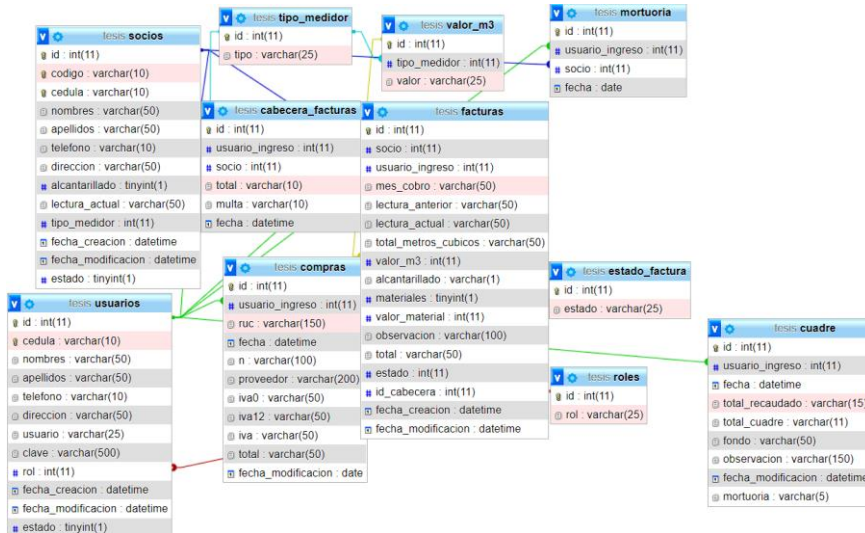


Elaborado por: El investigador

## Diseño de la base de datos

El diseño de la base de datos se realizó según la información proporcionada por los miembros de la junta de agua potable Huambaló caserío La Merced con el fin de cumplir todos sus requerimientos.

Ilustración 23. Diseño de la base de datos

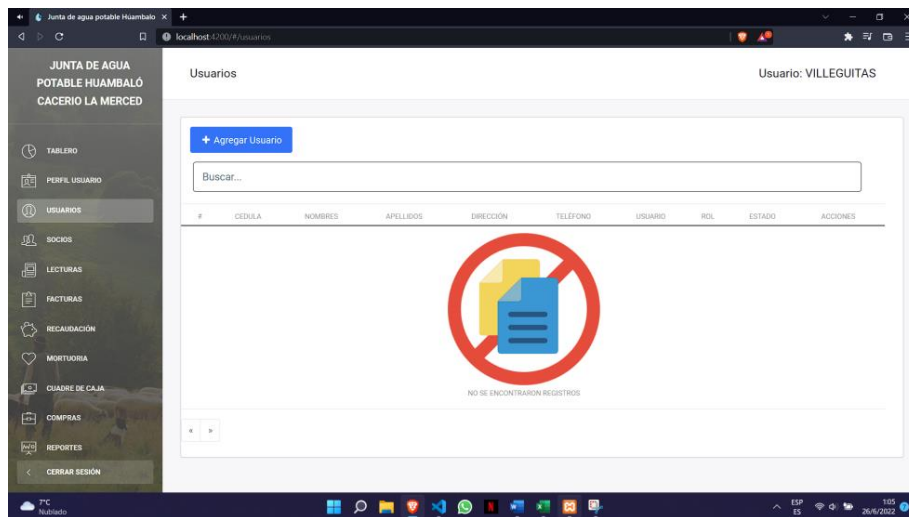


Elaborado por: El investigador

## Diseño de interfaz módulo usuarios

En la presente pantalla se mostrará el listado de los usuarios que se encuentren registrados en el sistema tanto si se encuentran activos como inactivos, al dar click sobre el botón **agregar usuario** se mostrara un modal en el cual se deberán ingresar los datos del nuevo usuario, al dar click sobre el botón editar le permitirá al usuario editar dicho usuario y al presionar el botón de habilitar y deshabilitar usuario se le permitirá cambiar el estado actual del usuario.

Ilustración 24. Interfaz módulo usuarios



Elaborado por: El investigador

### 3.2.2.3 Iteración II

Tabla 91. Estimación de la segunda iteración

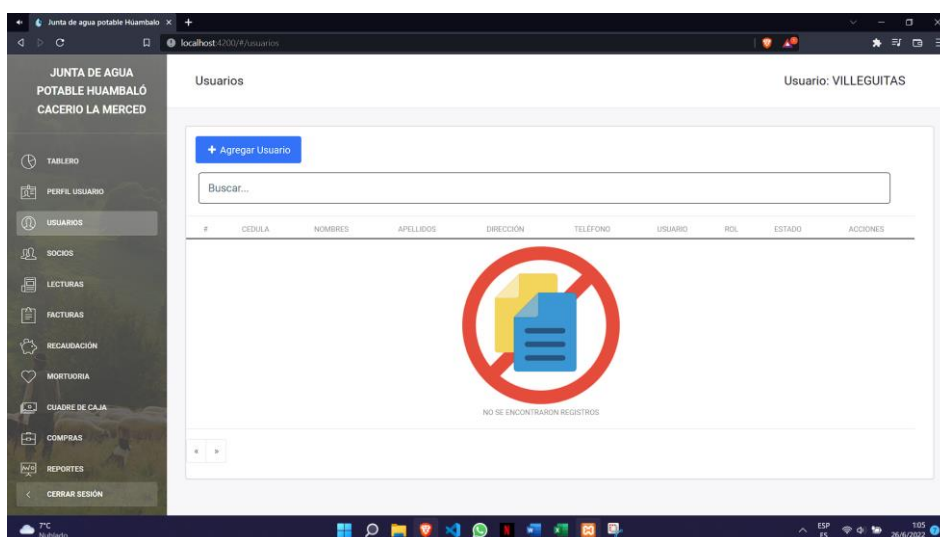
Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
2	H004	Diseño de interfaz módulo socios	3	15
2	H005	Diseño de interfaz módulo lecturas	3	15
2	H006	Diseño de interfaz módulo facturas	4	20
Total:			10	50

Elaborado por: El investigador

## Diseño de interfaz módulo socios

En la presente pantalla se mostrará el listado de los socios que se encuentren registrados en el sistema tanto si se encuentran activos como inactivos, al dar click sobre el botón **agregar socio** se mostrara un modal en el cual se deberán ingresar los datos del nuevo socio, al dar click sobre el botón editar le permitirá al usuario editar dicho socio y al presionar el botón de habilitar y deshabilitar socio se le permitirá cambiar el estado actual del socio.

Ilustración 25. Interfaz módulo socios



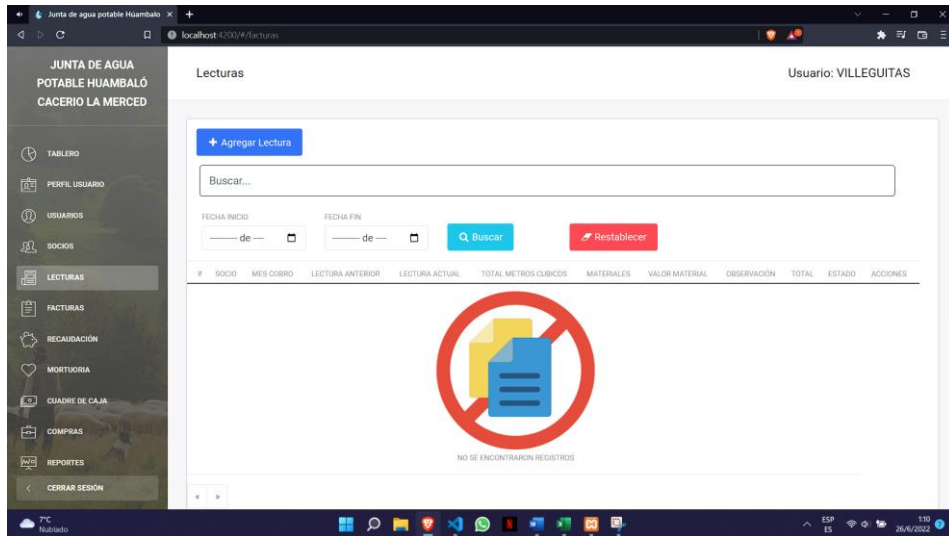
Elaborado por: El investigador

## Diseño de interfaz módulo lecturas

En la presente pantalla se mostrará el listado de las lecturas que se encuentren registrados en el sistema tanto si se encuentran pendientes o no de pago, al dar click sobre el botón **agregar lectura** se mostrara un modal en el cual se deberán ingresar los datos de la nueva lectura, al dar click sobre el botón editar le permitirá al usuario editar dicha factura y al presionar el botón eliminar procederá a eliminar la lectura del sistema.



Ilustración 26. Interfaz módulo lecturas

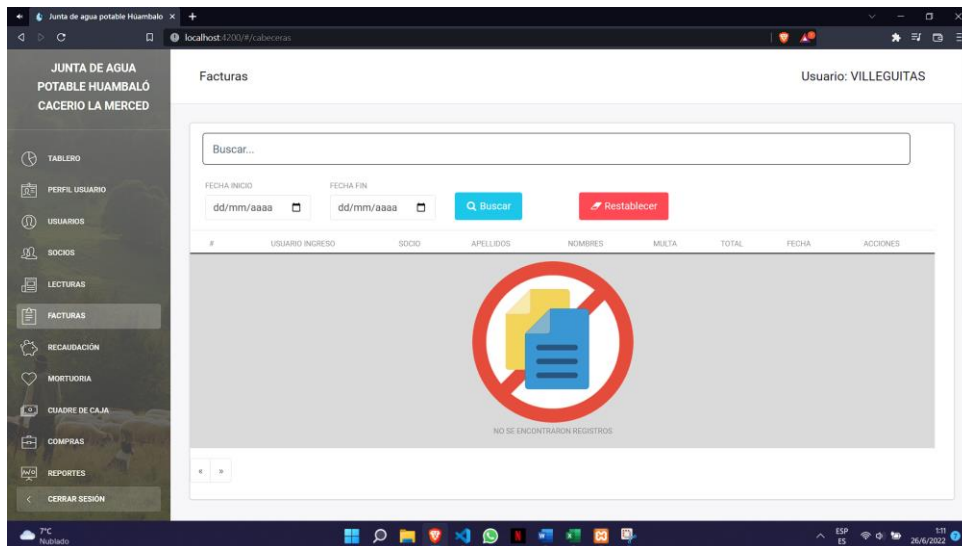


Elaborado por: El investigador

### Diseño de interfaz módulo facturas

En la presente pantalla se mostrará el listado de las facturas que se encuentren registrados en el sistema.

Ilustración 27. Interfaz módulo de facturas



Elaborado por: El investigador

### 3.2.2.4 Iteración III

Tabla 92. Estimación de la tercera iteración

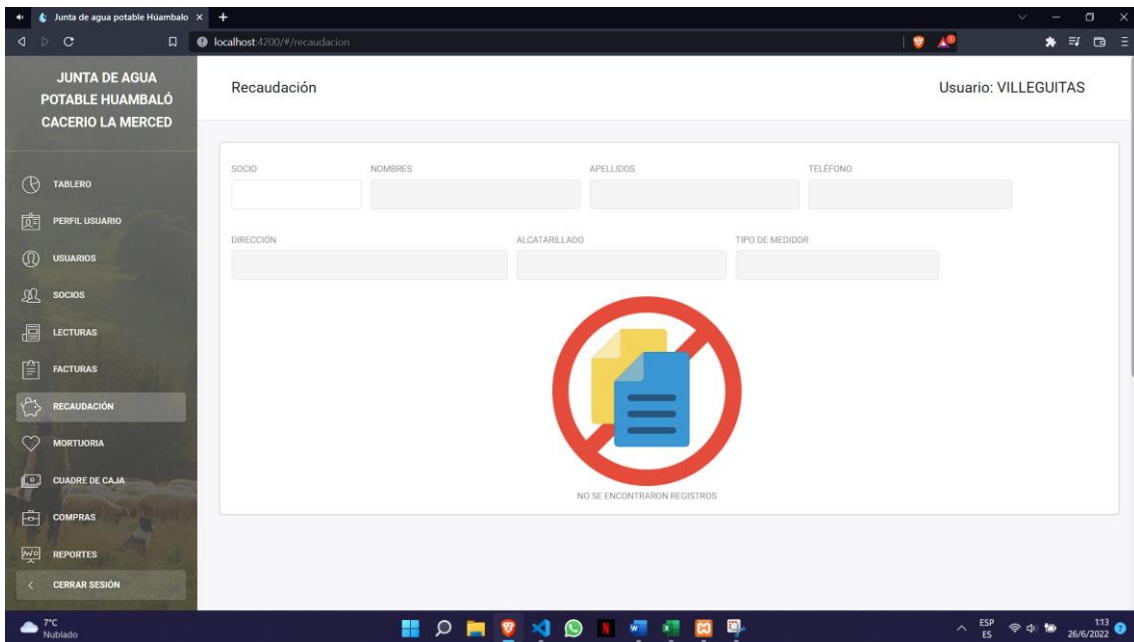
Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
3	H007	Diseño de interfaz modulo recaudación	4	20
3	H008	Diseño de interfaz modulo mortuoria	3	15
3	H009	Diseño de interfaz modulo cuadro de caja	3	15
3	H010	Diseño de interfaz modulo compras	3	15
Total:			13	65

Elaborado por: El investigador

#### Diseño de interfaz módulo recaudación

En la presente pantalla se mostrará el listado de las facturas que se encuentren registrados en el sistema a nombre del socio respectivo y que se encuentren pendientes de pago.

Ilustración 28. Interfaz módulo recaudación

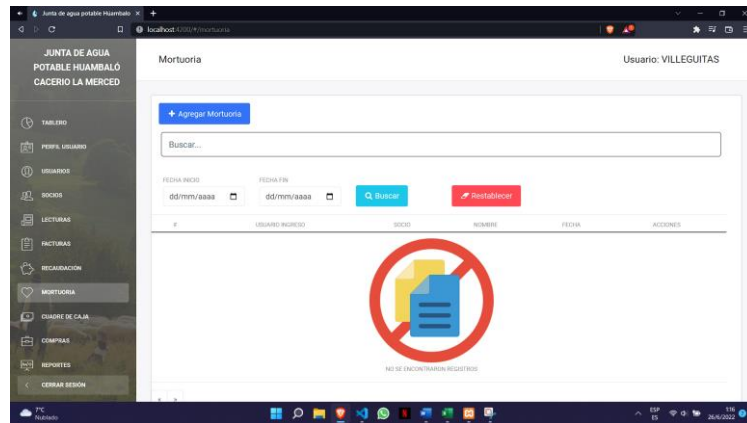


Elaborado por: El investigador

## Diseño de interfaz módulo mortuoria

En la presente pantalla se mostrará el listado mortuorio que se encuentren registrados en el sistema.

Ilustración 29. Interfaz módulo mortuoria

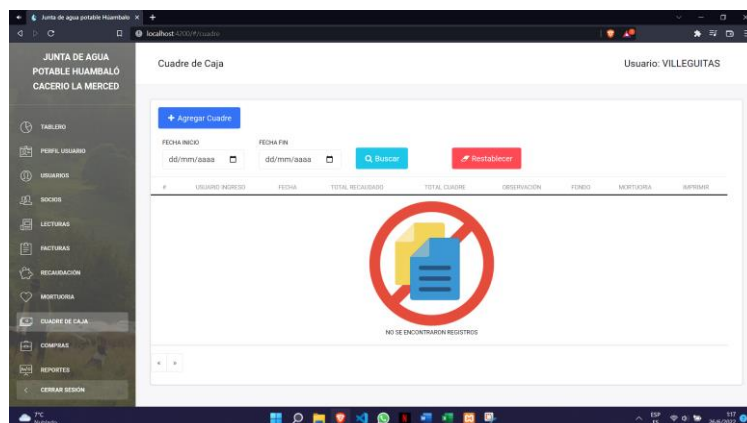


Elaborado por: El investigador

## Diseño de interfaz módulo cuadro de caja

En la presente pantalla se mostrará el listado de los cuadros de caja que se encuentren registrados en el sistema, adicionalmente se puede realizar la revisión de cada una de los cuadros de caja. También se pueden agregar nuevos registros al dar click sobre el botón **Agregar Cuadre**.

Ilustración 30. Interfaz módulo cuadro de caja

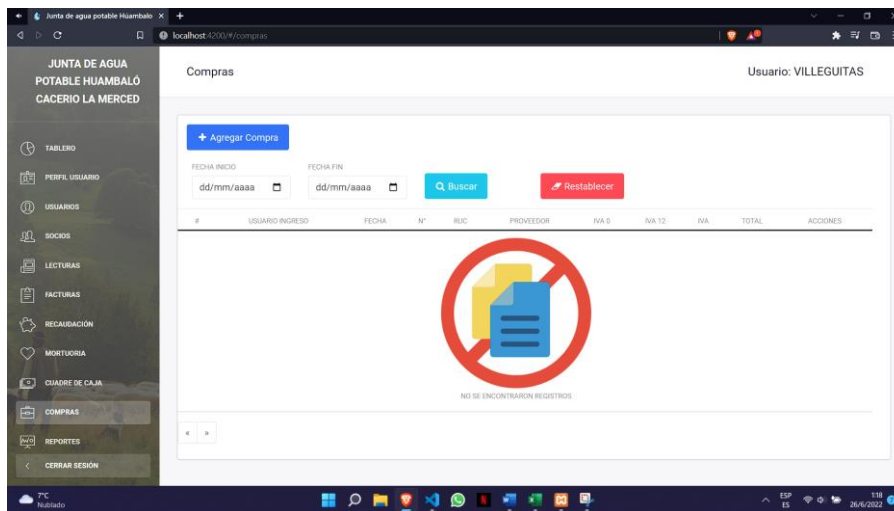


Elaborado por: El investigador

## Diseño de interfaz módulo compras

En la presente pantalla se mostrará el listado de las compras encuentren registradas en el sistema, adicionalmente se puede realizar la revisión de cada una de las compras y eliminarlas. También se pueden agregar nuevos registros al dar click sobre el botón **Agregar Compra**.

Ilustración 31. Interfaz módulo compras



Elaborado por: El investigador

### 3.2.2.5 Iteración IV

Tabla 93. Estimación de la cuarta iteración

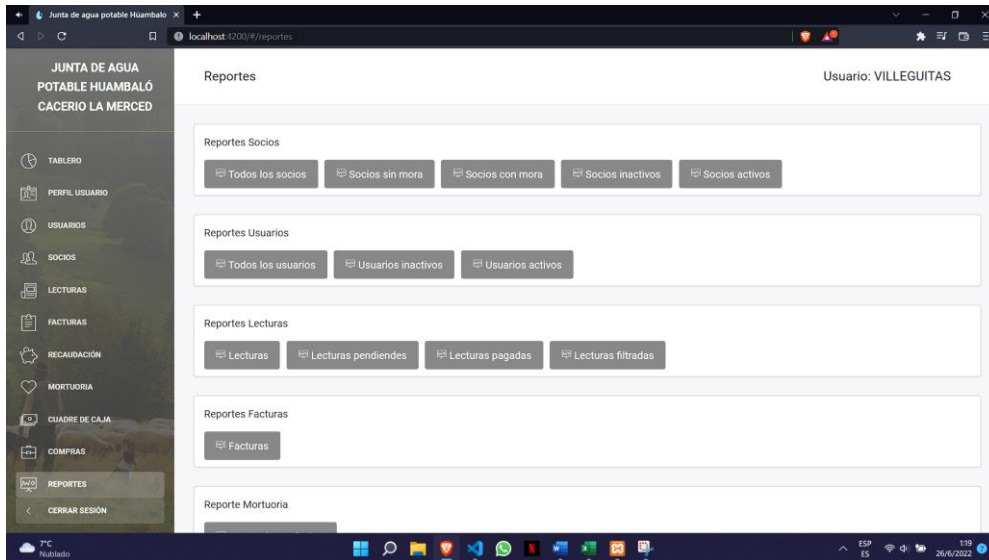
Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
4	H011	Diseño de interfaz modulo reportes	3	15
4	H012	Inicio de sesión	4	20
4	H013	Listado de usuarios	2	10
4	H014	Creación de usuarios	2	10
Total:			11	55

Elaborado por: El investigador

## Diseño de interfaz módulo reportes

En el módulo de reportes estarán presentes todas las variantes de los reportes que podrán ser generados por el sistema.

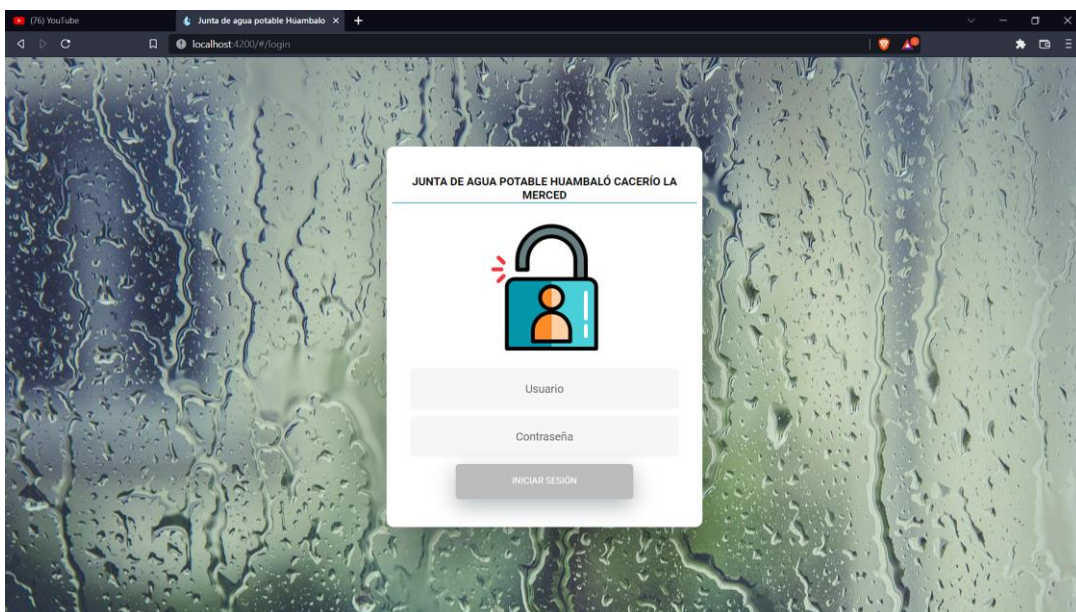
Ilustración 32. Interfaz módulo reportes



Elaborado por: El investigador

## Inicio de sesión

Ilustración 33. Interfaz Login



Elaborado por: el investigador

## Listado de usuarios

Listado de usuarios registrados en el sistema

Ilustración 34. Listado de usuarios

#	CEDULA	NOMBRES	APELLIDOS	DIRECCIÓN	TELÉFONO	USUARIO	ROL	ESTADO	ACCIONES
1	1801977123	Valeria	Paredes	Ambato	0999952397	Valeria	recaudador	Activo	
2	2350495194	Javier	Ramos	Ambato	0999952397	javier	consultor	Activo	
3	2300687510	Alex	Villegas	Ambato	0999952397	villeguitas	admin	Activo	

Elaborado por: el investigador

## Creación de usuarios

A continuación, se presenta el modal de agregar usuario en el cual pedirá ingresar los respectivos datos del usuario, una vez se verifique que los datos han sido ingresados de forma correcta permitirá realizar la acción de guardar, esta acción solo estará permitida para el usuario administrador.

Ilustración 35. Creación de usuarios

Agregar usuario

Cedula:

Nombres:

Apellidos:

Dirección:

Teléfono:

Rol:

Estado:  Activo  Inactivo

Usuario:

Contraseña:

Elaborado por: el investigador

### 3.2.2.6 Iteración V

Tabla 94. Estimación de la quinta iteración

Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
5	H015	Edición de usuarios	2	10
5	H016	Habilitar y deshabilitar usuarios	2	10
5	H017	Listar socios	2	10
5	H018	Crear socio	2	10
5	H019	Editar socios	2	10
5	H020	Habilitar y deshabilitar socios	2	10
Total:			12	60

Elaborado por: El investigador

#### Edición de usuarios

En el presente modal se puede realizar la modificación de datos del usuario, esta acción solo estará permitida para el usuario administrador.

Ilustración 36. Edición de usuarios

The image shows a modal window titled "Editar usuario" with a close button (X) in the top right corner. The form contains the following fields and controls:

- Cedula:** Text input with value "2300687510".
- Nombres:** Text input with value "Alex".
- Apellidos:** Text input with value "Villegas".
- Dirección:** Text input with value "Ambato".
- Teléfono:** Text input with value "0999952397".
- Rol:** Dropdown menu with "admin" selected.
- Estado:** Two buttons: "Activo" (green) and "Inactivo" (red).
- Usuario:** Text input with value "villeguitas".

At the bottom right of the modal, there are two buttons: "Cancelar" (blue) and "Guardar" (green).

Elaborado por: El investigador

## Habilitar y deshabilitar usuarios

En el presente modal se podrá habilitar y deshabilitar los usuarios para permitir o restringir el acceso al sistema, esta acción solo estará permitida para el usuario administrador.

Ilustración 37. Habilitar y deshabilitar usuarios



Elaborado por: El investigador

## Listar socios

Listado de socios registrados en el sistema.

Ilustración 38. Listar socios

Socios Usuario: VILLEGUITAS

The screenshot shows a list of registered partners. The interface includes a search bar and a table with columns for ID, code, ID card, names, address, phone, status, and actions. The table contains the following data:

#	CODIGO	CEDULA	APELLIDOS	NOMBRES	DIRECCIÓN	TELÉFONO	ALCANTARILLADO	ESTADO	TIPO DE MEDIDOR	ACCIONES
1	LM001	1800281832	Altamirano Martinez	Mesias Faraón	La Merced	0987541259	✓	Activo	Domestico	[Edit] [Delete]
2	LM002	1801977123	Altamirano Martinez	David Benjamín	La Merced	0954856145	✓	Activo	Domestico	[Edit] [Delete]
3	LM003	1800282673	Aponte Martinez	Luz Marieta	La Merced	0985514558	✓	Activo	Domestico	[Edit] [Delete]
4	LM004	1803988201	Arcos Villegas	Cesar Manuel	La Merced	0958745587	✓	Activo	Domestico	[Edit] [Delete]
5	LM005	1802500346	Campos	Iván Alcidas	La Merced	0998952397	✓	Activo	Domestico	[Edit] [Delete]
6	LM006	1802012425	Campos	Jesús Eduardo	La Merced	0985475478	✓	Activo	Domestico	[Edit] [Delete]
7	LM007	1800000000	Campos	Jesús Eduardo	La Merced	0985475478	✓	Activo	Domestico	[Edit] [Delete]

Elaborado por: El investigador



## Crear socio

A continuación, se presenta el modal de agregar socio en el cual pedirá ingresar los respectivos datos del socio, una vez se verifique que los datos han sido ingresados de forma correcta permitirá realizar la acción de guardar, esta acción solo estará permitida para el usuario administrador.

Ilustración 39. Crear socio

Socios

Usuari

Agregar socio

Codigo: LM000 Cedula: 2300687510

Nombres: Alex Jair Apellidos: Villegas Alvarado

Dirección: La Merced Teléfono: 0999952397

Alcantarillado:  Si  No Tipo Medidor: Domestico Estado:  Activo  Inactivo Lectura actual:

Cancelar Guardar

#	Codigo	Cedula	Nombres	Apellidos	Dirección	Teléfono	Alcantarillado	Tipo Medidor	Estado	Lectura actual
1	LM000	2300687510	Alex Jair	Villegas Alvarado	La Merced	0999952397	<input type="radio"/> Si <input type="radio"/> No	Domestico	<input type="radio"/> Activo <input type="radio"/> Inactivo	
2	LM006	1802012425	Campos	Jesús Eduardo	La Merced	0985475478	<input type="radio"/> Si <input type="radio"/> No	Domestico	<input checked="" type="radio"/> Activo <input type="radio"/> Inactivo	

Elaborado por: El investigador

## Editar socios

En el presente modal se puede realizar la modificación de datos del socio, esta acción solo estará permitida para el usuario administrador.

Ilustración 40. Editar socio

Socios

Usuari

Editar socio

Codigo: LM001 Cedula: 1800281832

Nombres: Mesías Faraón Apellidos: Altamirano Martínez

Dirección: La Merced Teléfono: 0987541259

Alcantarillado:  Si  No Tipo Medidor: Domestico Estado:  Activo  Inactivo Lectura actual: 40

Cancelar Guardar

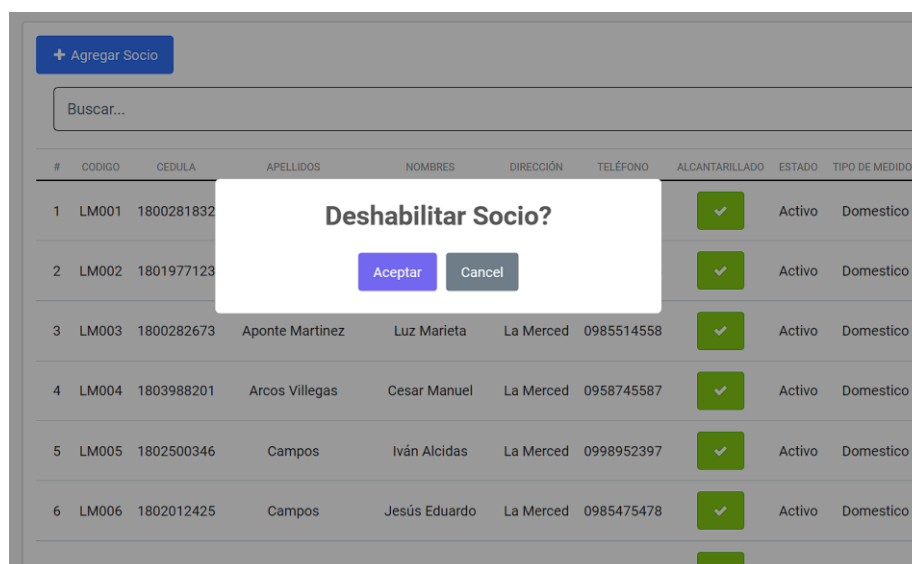
#	Codigo	Cedula	Nombres	Apellidos	Dirección	Teléfono	Alcantarillado	Tipo Medidor	Estado	Lectura actual
1	LM001	1800281832	Mesías Faraón	Altamirano Martínez	La Merced	0987541259	<input type="radio"/> Si <input type="radio"/> No	Domestico	<input type="radio"/> Activo <input type="radio"/> Inactivo	40
2	LM006	1802012425	Campos	Jesús Eduardo	La Merced	0985475478	<input type="radio"/> Si <input type="radio"/> No	Domestico	<input checked="" type="radio"/> Activo <input type="radio"/> Inactivo	

Elaborado por: El investigador

## Habilitar y deshabilitar socios

En el presente modal se podrá habilitar y deshabilitar los socios esto servirá para que no se le sume más recargos al socio en caso de que deje de ser parte de la junta de agua potable, esta acción solo estará permitida para el usuario administrador.

Ilustración 41. Habilitar y deshabilitar socios



Elaborado por: El investigador

### 3.2.2.7 Iteración VI

Tabla 95. Estimación de la sexta iteración

Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
6	H021	Listas lecturas	2	10
6	H022	Agregar lecturas	2	10
6	H023	Editar lecturas	2	10
6	H024	Eliminar lecturas	2	10
6	H025	Listar facturas	2	10
6	H026	Cobro de factura	5	25
Total:			15	75

Elaborado por: El investigador

## Listas lecturas

Lista de lecturas registradas en el sistema.

Ilustración 42. Lista de lecturas

Lecturas Usuario: VILLEGUITAS

[+ Agregar Lectura](#)

Buscar...

FECHA INICIO: de -- de -- FECHA FIN: de -- de -- [Buscar](#) [Restablecer](#)

#	SOCIO	MES COBRO	LECTURA ANTERIOR	LECTURA ACTUAL	TOTAL METROS CUBICOS	MATERIALES	VALOR MATERIAL	OBSERVACION	TOTAL	ESTADO	ACCIONES
1	LM001	2022-06	30	40	10	No			\$4.00	Pagado	<a href="#">+</a> <a href="#">-</a> <a href="#">B</a> <a href="#">+</a> <a href="#">-</a> <a href="#">B</a>
2	LM001	2022-04	20	30	10	No			\$4.00	Pagado	<a href="#">+</a> <a href="#">-</a> <a href="#">B</a> <a href="#">+</a> <a href="#">-</a> <a href="#">B</a>
3	LM001	2022-03	10	20	10	No			\$4.00	Pagado	<a href="#">+</a> <a href="#">-</a> <a href="#">B</a> <a href="#">+</a> <a href="#">-</a> <a href="#">B</a>

Elaborado por: El investigador

## Agregar lecturas

A continuación, se presenta el modal de agregar lectura en el cual pedirá ingresar los respectivos datos de la lectura, una vez se verifique que los datos han sido ingresados de forma correcta permitirá realizar la acción de guardar, esta acción solo estará permitida para el usuario administrador y recaudador.

Ilustración 43. Agregar lecturas

Agregar lectura ×

Mes de cobro: de -- de -- Estado: Pendiente

[← Anterior](#) [Siguiente →](#)

Socio:  Nombres:  Apellidos:

Lectura anterior:  Lectura actual:  Total metros cubicos:  Tipo Medidor:

Valor del m3:  Alcantarillado:  Materiales: Si No

Observación:  Total:

[Cancelar](#) [Guardar](#)

Elaborado por: El investigador

### Editar lecturas

En el presente modal brindará la posibilidad de editar las lecturas registradas, solo tendrá acceso el usuario administrador.

Ilustración 44. Editar lecturas

Editar lectura

Mes de cobro: junio de 2022 Estado: Pagado

Socio: LM001 Nombres: Mesías Faraón Apellidos: Altamirano Martinez

Lectura anterior: 30 Lectura actual: 40 Total metros cubicos: 10 Tipo Medidor: Domestico

Valor del m3: 0.30 Alcantarillado: Si Materiales: Total: 4.00

Observación:

Cancelar Guardar

Elaborado por: El investigador

### Eliminar lecturas

El presente modal permite eliminar las lecturas registradas en el sistema, esta función solo se encontrará disponible para el usuario administrador.

Ilustración 45. Eliminar lecturas

FECHA INICIO de

**Eliminar lectura**

La lectura se eliminara permanentemente de la base de datos

Aceptar Cancel

#	SOCIO	MES COBRO	OBSERVACIÓN	TOTAL	ESTADO
1	LM001	2022-06		\$4.00	Pagado
2	LM001	2022-04	20 30 10 No	\$4.00	Pagado
3	LM001	2022-03	10 20 10 No	\$4.00	Pagado

Elaborado por: El investigador

## Listar facturas

Listado de facturas registradas en el sistema.

Ilustración 46. Listar facturas

Facturas Usuario: VILLEGUITAS

Buscar...

FECHA INICIO: dd/mm/aaaa  FECHA FIN: dd/mm/aaaa

#	USUARIO INGRESO	SOCIO	APELLIDOS	NOMBRES	MIXTA	TOTAL	FECHA	ACCIONES
1	villeguitas	LM008	Campos Campos	Ivan	0	\$4	23/junio/2022, 2:10 a. m.	<input type="button" value="↓"/>
2	villeguitas	LM002	Altamirano Martinez	David Benjamín	0	\$4	22/junio/2022, 2:04 a. m.	<input type="button" value="↓"/>
3	villeguitas	LM001	Altamirano Martinez	Mesías Faraón	3	\$19	22/junio/2022, 2:03 a. m.	<input type="button" value="↓"/>
4	villeguitas	LM011	Campos Campos	Ines Balvina	0	\$4.54	22/junio/2022, 2:02 a. m.	<input type="button" value="↓"/>
5	villeguitas	Lm010	Campos Campos	Rosa Emitela	0	\$7.5	22/junio/2022, 2:02 a. m.	<input type="button" value="↓"/>
6	villeguitas	LM008	Campos Campos	Ivan	0	\$4	22/junio/2022, 2:00 a. m.	<input type="button" value="↓"/>

Elaborado por: El investigador

## Cobro de factura

En la siguiente pantalla ingresar el código del socio se cargarán sus respectivos datos y la lista de lecturas pendientes de pago.

Ilustración 47. Cobro de facturas

Recaudación Usuario: VILLEGUITAS

SOCIO: Im006 NOMBRES: Jesús Eduardo APELLIDOS: Campos TELÉFONO: 0985475478

DIRECCIÓN: La Merced ALCATARRILLADO: Si TIPO DE MEDIDOR: Domestico

#	SOCIO	MES COBRO	LECTURA ANTERIOR	LECTURA ACTUAL	TOTAL M3	MATERIALES	VALOR MATERIALES	OBSERVACIÓN	ESTADO	TOTAL	ACCIONES
1	LM006	2022-05	0	30	30	No			Pendiente	\$10.00	<input type="button" value="❌"/>

Total: \$10

Elaborado por: El investigador

### 3.2.2.8 Iteración VII

Tabla 96. Estimación de la séptima iteración

Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
7	H027	Listar mortuoria	2	10
7	H028	Agregar mortuoria	2	10
7	H029	Eliminar mortuoria	2	10
7	H030	Listar cuadro de caja	2	10
7	H031	Agregar cuadro de caja	2	10
Total:			10	50

Elaborado por: El investigador

### Listar mortuoria

Listado mortuorio registrado en el sistema.

Ilustración 48. Lisa mortuoria

Mortuoria Usuario: VILLEGUITAS

[+ Agregar Mortuoria](#)

FECHA INICIO

FECHA FIN

🔍 Buscar

🔄 Restablecer

#	USUARIO INGRESO	SOCIO	NOMBRE	FECHA	ACCIONES
1	villeguitas	LM001	Mesías Faraón Altamirano Martinez	22/junio/2022	<span style="background-color: #dc3545; color: white; padding: 2px 5px; border-radius: 5px;">🗑️</span> <span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 5px;">📄</span>

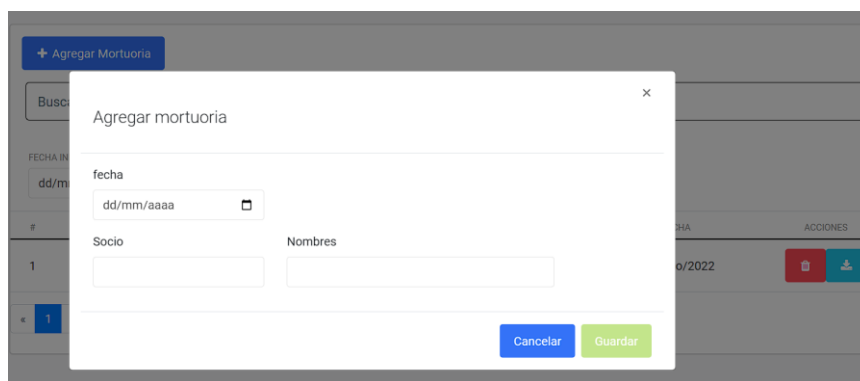
« 1 »

Elaborado por: El investigador

## Agregar mortuoria

A continuación, se presenta el modal de agregar mortuoria en el cual pedirá ingresar los respectivos datos de la mortuoria, una vez se verifique que los datos han sido ingresados de forma correcta permitirá realizar la acción de guardar, esta acción solo estará permitida para el usuario administrador y recaudador.

Ilustración 49. Agregar mortuoria



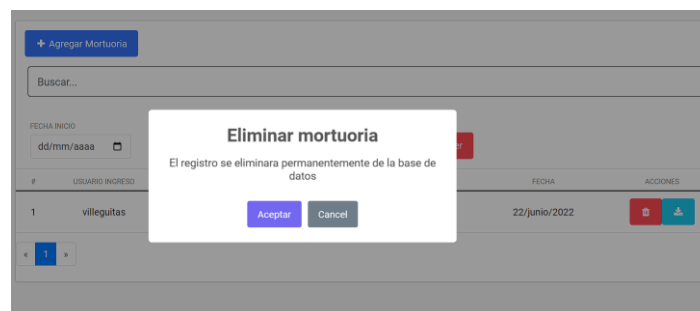
The image shows a web application interface with a modal window titled "Agregar mortuoria". The modal contains a search bar at the top, followed by a "fecha" field with a date picker showing "dd/mm/aaaa". Below this are two input fields labeled "Socio" and "Nombres". At the bottom right of the modal are two buttons: "Cancelar" (blue) and "Guardar" (green). The background shows a table with columns for "#", "FECHA INICIO", "USUARIO INGRESO", "FECHA", and "ACCIONES". A row is visible with the value "1" in the first column and "villeguitas" in the "USUARIO INGRESO" column.

Elaborado por: El investigador

## Eliminar mortuoria

El modal a continuación brinda la opción de eliminar las mortuorias registradas en el sistema, esta función solo estará despoñlé para el usuario administrador.

Ilustración 50. Eliminar mortuoria



The image shows the same web application interface as before, but with a modal dialog titled "Eliminar mortuoria". The dialog contains the text "El registro se eliminara permanentemente de la base de datos" and two buttons: "Aceptar" (blue) and "Cancel" (grey). The background table is visible, showing a row with "1" in the first column and "villeguitas" in the "USUARIO INGRESO" column. The date "22/junio/2022" is visible in the "FECHA" column.

Elaborado por: El investigador

## Listar cuadro de caja

Listado de cuadros de cajas registrados en el sistema.

Ilustración 51. Listar cuadro de caja

Cuadro de Caja Usuario: VILLEGUITAS

[+ Agregar Cuadro](#)

FECHA INICIO:  FECHA FIN:  [Buscar](#) [Restablecer](#)

#	USUARIO INGRESO	FECHA	TOTAL RECAUDADO	TOTAL CUADRO	OBSERVACIÓN	FONDO	MORTUORIA	IMPRIMIR
1	villeguitas	7/junio/2022	\$4	\$4			\$1	<a href="#">↓</a>
2	villeguitas	7/junio/2022	\$19	\$19			\$0	<a href="#">↓</a>
3	villeguitas	2/mayo/2022	\$120	\$120	1	\$1		<a href="#">↓</a>
4	villeguitas	2/mayo/2022	\$120	\$120				<a href="#">↓</a>
5	villeguitas	2/mayo/2022	\$	\$500	Nada	\$52		<a href="#">↓</a>

[<](#) [1](#) [>](#)

Elaborado por: El investigador

## Agregar cuadro de caja

En el presente se puede realizar el cuadro de caja, una vez se verifique que los datos ingresados el formulario permitirá guardar el registro.

Ilustración 52. Agregar cuadro de caja

Fecha:

<b>Billetes</b>	<b>Total</b>	<b>Monedas</b>	<b>Total</b>
100 USD: <input type="text"/>	<input type="text"/>	1 USD: <input type="text"/>	<input type="text"/>
50 USD: <input type="text"/>	<input type="text"/>	0.50 USD: <input type="text"/>	<input type="text"/>
20 USD: <input type="text"/>	<input type="text"/>	0.25 USD: <input type="text"/>	<input type="text"/>
10 USD: <input type="text"/>	<input type="text"/>	0.10 USD: <input type="text"/>	<input type="text"/>
5 USD: <input type="text"/>	<input type="text"/>	0.05 USD: <input type="text"/>	<input type="text"/>
1 USD: <input type="text"/>	<input type="text"/>	0.01 USD: <input type="text"/>	<input type="text"/>
Total recaudado en el día: <input type="text"/>		Total cuadro de caja: <input type="text"/>	
Total recaudado mortuoria: <input type="text"/>			
Observación: <input type="text"/>			
Fondo de caja: <input type="text"/>			

[Cancelar](#) [Guardar](#)

Elaborado por: El investigador



### 3.2.2.9 Iteración VIII

Tabla 97. Estimación de la octava iteración







Iteración	Identificador	Nombre historia	Tiempo estimado	
			Días	Horas
7	H021	Listar compras	2	10
7	H022	Agregar compra	2	10
7	H023	Editar compra	2	10
7	H024	Eliminar compra	2	10
7	H025	Generación de reportes	7	10
Total:			10	50

Elaborado por: El investigador

#### Listar compras

Listado de compras registradas en el sistema.

Ilustración 53. Listar compras

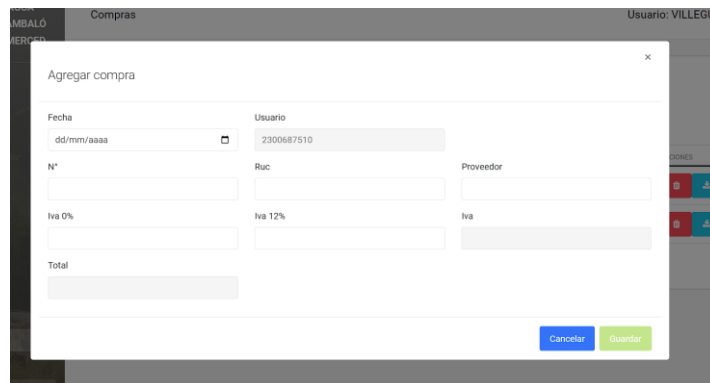
#	USUARIO INGRESO	FECHA	N°	RUC	PROVEEDOR	IVA 0	IVA 12	IVA	TOTAL	ACCIONES
1	villeguitas	4/mayo/2022	75548	784845577	ssss	\$1	\$0.12	\$1.12		  
2	villeguitas	6/mayo/2022	7777	77777	7777	\$77	\$0.00	\$77.00		  

Elaborado por: El investigador

#### Agregar compra

A continuación, se presenta el modal en el cual se puede añadir las compras realizadas por la junta de agua potable, una vez se ingresen todos los datos el formulario permitirá realizar la acción de guardar, esta función solo se encontrará deshabilitada para el usuario administrador y recaudador.

## Ilustración 54. Agregar compras



The screenshot shows a modal window titled 'Agregar compra' with a close button (X) in the top right corner. The form contains the following fields:

- Fecha:** A date picker showing 'dd/mm/aaaa' with a calendar icon.
- Usuario:** A text input field containing '2300687510'.
- N°:** A text input field.
- Ruc:** A text input field.
- Proveedor:** A text input field.
- Iva 0%:** A text input field.
- Iva 12%:** A text input field.
- Iva:** A text input field.
- Total:** A text input field.

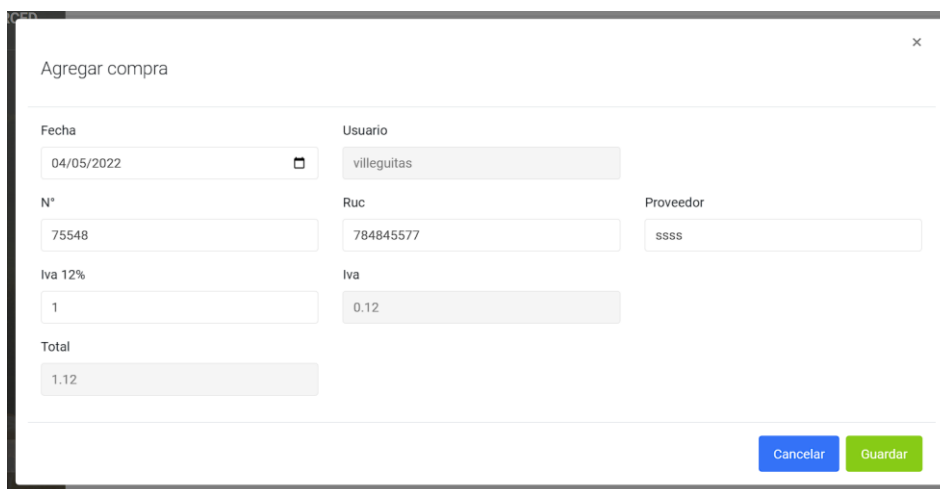
At the bottom right of the modal, there are two buttons: 'Cancelar' (blue) and 'Guardar' (green).

Elaborado por: El investigador

## Editar compra

En el presente modal brindará la posibilidad de editar las compras registradas, solo tendrá acceso el usuario administrador.

## Ilustración 55. Editar compra



The screenshot shows the same 'Agregar compra' modal window, but with pre-filled data in the input fields:

- Fecha:** 04/05/2022
- Usuario:** villeguitas
- N°:** 75548
- Ruc:** 784845577
- Proveedor:** ssss
- Iva 12%:** 1
- Iva:** 0.12
- Total:** 1.12

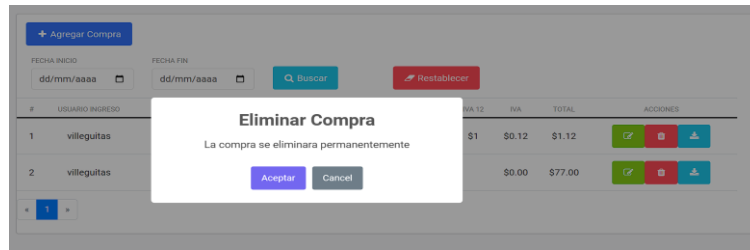
The 'Cancelar' and 'Guardar' buttons are still present at the bottom right.

Elaborado por: El investigador

## Eliminar compra

En el presente modal permitirá realizar la eliminación de las compras registradas, esta función de encontrar desoplé únicamente para el usuario administrador.

## Ilustración 56. Eliminar compra



Elaborado por: El investigador

## Generación de reportes

## Ilustración 57. Reporte de socios

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERÍO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERÍO LA MERCED VIA A SURANGAY S/N  
CEL:09977762490 • SAN PEDRO DE PELILEO - ECUADOR  
LISTADO DE SOCIOS

#	Codigo	Cedula	Nombres	Apellidos	Dirección	Teléfono	Alcantarillado	Estado	Tipo de medidor
1	LM001	1800281832	Mesías Faraón	Altamirano Martínez	La Merced	0987541259	Si	Activo	Domestico
2	LM002	1801977123	David Benjamín	Altamirano Martínez	La Merced	0954856145	Si	Activo	Domestico
3	LM003	1800282673	Luz Marieta	Aponte Martínez	La Merced	0985514558	Si	Activo	Domestico
4	LM004	1803988201	Cesar Manuel	Arco Villegas	La Merced	0958725587	Si	Activo	Domestico
5	LM005	1802500346	Iván Alcidas	Campos	La Merced	0998952397	Si	Activo	Domestico
6	LM006	1802012425	Jesús Eduardo	Campos	La Merced	0985475478	Si	Activo	Domestico
7	LM007	1802884898	Rene	Campos	La Merced	0965231454	Si	Activo	Industrial

Elaborado por: El investigador

## Ilustración 58. Reporte de usuarios

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERIO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERÍO LA MERCED VIA A SURANGAY S/N  
CEL:09977762490 \* SAN PEDRO DE PELILEO - ECUADOR  
LISTADO DE USUARIOS

#	Cedula	Nombres	Apellidos	Dirección	Teléfono	Rol	Estado
1	1801977123	Valeria	Paredes	Ambato	0999952397	recaudador	Activo
2	2350495194	Javier	Ramos	Ambato	0999952397	consultor	Activo
3	2300687510	Alex	Villegas	Ambato	0999952397	admin	Activo

Elaborado por: El investigador  
Ilustración 59. Reporte de lecturas

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERIO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERÍO LA MERCED VIA A SURANGAY S/N  
CEL:09977762490 \* SAN PEDRO DE PELILEO - ECUADOR  
LISTADO DE LECTURAS

#	Socio	Mes cobro	Lectura anterior	Lectura actual	Total m3	Materiales	Multa	Observación	Total	Estado
1	LM001	mayo/2022	30	40	10	No	No		\$4.00	Pagado
2	LM001	marzo/2022	20	30	10	No	No		\$4.00	Pagado
3	LM001	febrero/2022	10	20	10	No	No		\$4.00	Pagado
4	LM011	abril/2022	0	11.8	11.8	No	No		\$4.54	Pendiente
5	Lm010	abril/2022	0	15	15	No	No		\$7.50	Pendiente
6	LM009	abril/2022	0	1	1	No	No		\$4.00	Pendiente
7	LM008	abril/2022	0	1	1	No	No		\$4.00	Pagado
8	LM007	abril/2022	0	5	5	No	No		\$4.00	Pendiente
9	LM006	abril/2022	0	30	30	No	No		\$10.00	Pendiente
10	LM005	abril/2022	0	20	20	No	No		\$7.00	Pendiente
11	LM004	abril/2022	0	3	3	No	No		\$4.00	Pendiente

Elaborado por: El investigador

## Ilustración 60. Reporte de facturas

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERIO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERÍO LA MERCED VIA A SURANGAY S/N  
CEL.09977762490 \* SAN PEDRO DE PELILEO - ECUADOR  
LISTADO DE FACTURAS

#	Usuario Ingreso	Socio	Apellidos	Nombres	Multa	Total	Fecha
1	villeguitas	LM008	Campos Campos	Ivan	0	\$4	23/junio/2022, 2:10 a. m.
2	villeguitas	LM002	Altamirano Martinez	David Benjamín	0	\$4	22/junio/2022, 2:04 a. m.
3	villeguitas	LM001	Altamirano Martinez	Mesías Faraón	3	\$19	22/junio/2022, 2:03 a. m.
4	villeguitas	LM011	Campos Campos	Ines Balvina	0	\$4.54	22/junio/2022, 2:02 a. m.
5	villeguitas	Lm010	Campos Campos	Rosa Emitela	0	\$7.5	22/junio/2022, 2:02 a. m.
6	villeguitas	LM008	Campos Campos	Ivan	0	\$4	22/junio/2022, 2:00 a. m.
7	villeguitas	LM007	Campos	Rene	0	\$4	22/junio/2022, 1:59 a. m.
8	villeguitas	LM006	Campos	Jesús Eduardo	0	\$10	22/junio/2022, 1:57 a. m.
9	villeguitas	LM005	Campos	Iván Alcidas	0	\$7	22/junio/2022, 1:55 a. m.
10	villeguitas	LM002	Altamirano Martinez	David Benjamín	0	\$4	22/junio/2022, 1:54 a. m.

Elaborado por: El investigador

## Ilustración 61. Reporte de mortuorias

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERIO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERÍO LA MERCED VIA A SURANGAY S/N  
MORTUORIA

#	Usuario ingreso	Socio	Nombre	Fecha
1	villeguitas	LM001	Mesías Faraón Altamirano Martinez	22/junio/2022

Total recaudado de mortuoria : \$1

Elaborado por: El investigador

## Ilustración 62. Reporte de cuadros

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERIO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERIO LA MERCED VIA A SURANGAY S/N  
CEL:09977762490 \* SAN PEDRO DE PELILEO - ECUADOR  
CUADRE DE CAJA POR FECHAS

#	Usuario ingreso	Nombres	Fecha	Total Recaudado	Total Cuadre	Observación	Fondo	Mortuoria
1	villeguitas	Alex Villegas	7/6/22 1:37	4	4			1
2	villeguitas	Alex Villegas	7/6/22 1:35	19	19			0
3	villeguitas	Alex Villegas	2/5/22 23:49	120	120	1	1	
4	villeguitas	Alex Villegas	2/5/22 23:48	120	120			
5	villeguitas	Alex Villegas	2/5/22 10:01		500	Nada	52	

Elaborado por: El investigador

## Ilustración 63. Reporte de compras

JUNTA ADMINISTRATIVA DE AGUA POTABLE HUAMBALO  
CACERIO LA MERCED  
PROVINCIA DE TUNGURAHUA  
DIRECCIÓN: CACERIO LA MERCED VIA A SURANGAY S/N  
CEL:09977762490 \* SAN PEDRO DE PELILEO - ECUADOR  
LISTADO DE COMPRAS POR FECHAS

#	Usuario	Nombres	Fecha	N°	Ruc	Proveedor	Iva 0%	Iva 12%	Iva	Total
1	villeguitas	Alex Villegas	4/mayo/2022	75548	784845577	ssss		1	0.12	1.12
2	villeguitas	Alex Villegas	6/mayo/2022	7777	77777	7777	77		0.00	77.00

Elaborado por: El investigador

### 3.2.3 Fase III: Codificación

#### 3.2.3.1 Métodos del backend (Node.js)

- **Autenticación**

Este método valida las credenciales del usuario, si las credenciales son correctas genera un token de acceso al sistema, en caso contrario emite un mensaje de error.

Ilustración 64. Autenticación

```
public async login(req: Request, res: Response): Promise<void> {
  let { usuario, clave } = req.body;
  if (!(usuario && clave)) {
    res.status(400).json({ message: "Usuario & contraseña son requeridas!" });
  }
  const result = await pool.query(
    `SELECT u.id as id,u.usuario as usuario,u.clave, r.rol as rol from usuarios as u,roles as r where r.id = u.rol and usuario = [usuario]`
  );
  if (result.length === 0) {
    res.status(400).json({ message: "Usuario o contraseña incorrecto!" });
  } else {
    if (!bcryptjs.compareSync(clave, result[0].clave)) {
      res.status(400).json({ message: "Usuario o contraseña incorrecto!" });
    } else {
      const token = jwt.sign(
        {
          userId: result[0].id,
          username: result[0].usuario,
        },
        config.jwtSecret
      );
      res.json({
        message: "OK",
        token,
        usuario: result[0].usuario,
        userId: result[0].id,
        role: result[0].rol,
      });
    }
  }
}
```

Elaborado por: El investigador

- **Verificación de JSON Web Token**

Este método valida que el usuario que quiera realizar una acción dentro del sistema tenga un token, caso contrario no se le permitirá realizar ninguna acción dentro del mismo.

Ilustración 65. Verificación de JSON Web Token

```
export const checkJwt = (req: Request, res: Response, next: NextFunction) => {
  /* console.log('REQ---', req.headers); */
  const token = <string> req.headers['auth'];
  let jwtPayload;
  try {
    jwtPayload = <any>jwt.verify(token, config.jwtSecret);
    res.locals.jwtPayload = jwtPayload;
  } catch (error) {
    return res.status(401).json({message: 'No autorizado'});
  }
  const {userId, username} = jwtPayload;
  const newToken = jwt.sign({userId, username}, config.jwtSecret);
  res.setHeader('token', newToken);
  next();
}
```

Elaborado por: El investigador

- **Verificación del rol del usuario**

Este método valida que el usuario tenga el rol necesario para realizar las acciones necesarias.

Ilustración 66. Verificación del rol del usuario

```
export const checkRole = (roles: Array<string>) => {
  return async (req: Request, res: Response, next: NextFunction) => {
    const { userId } = res.locals.jwtPayload;
    let user: any;
    let rol: string;
    try {
      let user = await pool.query(
        "SELECT r.rol as rol from usuarios as u, roles as r where r.id = u.rol and u.id = ? ",
        [userId]
      );
      rol = user[0].rol;
    } catch (error) {
      return res.status(401).json({ message: "No autorizado" });
    }
    if (roles.includes(rol)) {
      next();
    } else {
      res.status(401).json({ message: "No autorizado" });
    }
  };
};
```

Elaborado por: El investigador



- **Cambio de contraseña**

Este método valida que la contraseña actual coincida con la contraseña enviada por el usuario, si la contraseña coincide permite la actualización de la contraseña.

Ilustración 67. Cambio de contraseña

```
public async cambiarClave(req: Request, res: Response): Promise<void> {
  let { id, clave, nueva } = req.body;
  let user = await pool.query("SELECT * from usuarios where id = ? ", [
    id,
  ]);
  if (!bcryptjs.compareSync(clave, user[0].clave)) {
    res.status(400).json({ message: "Contraseña actual incorrecta!" });
  } else {
    const salt = bcryptjs.genSaltSync(10);
    const nuevaClave = bcryptjs.hashSync(nueva, salt);
    try {
      await pool.query(`UPDATE usuarios set clave=? WHERE id = ? `, [
        nuevaClave,
        id
      ]);
      res.json({ message: "Ok" });
    } catch (error) {
      res.status(404).json({ message: "Error" });
    }
  }
}
```

Elaborado por: El investigador

- **Listar usuarios**

Este método permite listar los usuarios que se encuentran registrados en el sistema.

Ilustración 68. Listado de usuarios

```
public async list(req: Request, res: Response): Promise<void> {
  const result = await pool.query(
    `select u.id,u.cedula, u.nombres, u.clave,u.apellidos,u.telefono,u.direccion,u.usuario,r.rol, u.estado
    from usuarios as u, roles as r WHERE u.rol = r.id ORDER BY u.apellidos ASC`
  );
  res.json(result);
}
```

Elaborado por: El investigador

- **Agregar usuarios**

Este método permite el ingreso de nuevos usuarios en el sistema, valida que el número de cedula del usuario no se encuentre repetido.

Ilustración 69. Agregar usuario

```
public async create(req: Request, res: Response): Promise<void> {
  const fecha = new Date();

  req.body.fecha_creacion = fecha;
  req.body.fecha_modificacion = fecha;
  const salt = bcryptjs.genSaltSync(10);
  req.body.clave = bcryptjs.hashSync(req.body.clave, salt);
  const result = await pool.query(
    `SELECT * FROM usuarios where cedula=?`,
    req.body.cedula
  );
  if (result.length > 0) {
    res
      .status(404)
      .json({ message: "El número de cedula ya se encuentra registrado" });
    return;
  }
  try {
    const result = await pool.query(`INSERT INTO usuarios set ?`, [
      req.body,
    ]);
    res.json({ message: "OK" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Editar usuarios**

Este método permite editar los usuarios del sistema.

Ilustración 70. Editar usuario

```
public async update(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  const fecha = new Date();
  req.body.fecha_modificacion = fecha;
  try {
    await pool.query(`UPDATE usuarios set ? WHERE id = ?`, [req.body, id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Buscar usuarios**

Este método permite buscar los a los usuarios según cedula, nombre, apellido, teléfono, dirección, usuario y rol.

Ilustración 71. Buscar usuario

```
public async buscarUsuario(req: Request, res: Response): Promise<void> {
  const { busqueda } = req.body;
  try {
    const result =
      await pool.query(`select u.id,u.cedula, u.nombres, u.clave,u.apellidos,u.telefono,u.direccion,u.usuario,r.rol, u.estado
      from usuarios as u, roles as r WHERE u.rol = r.id and concat(u.cedula,u.nombres,u.clave,u.apellidos,u.telefono,u.direccion,
      u.usuario,r.rol) LIKE '%${busqueda}%' ORDER BY u.apellidos ASC`);
    res.json(result);
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Listar socios**

Este método permite listar los socios que se encuentran registrado en el sistema.

Ilustración 72. Listar socios

```
public async list(req: Request, res: Response): Promise<void> {
  const result = await pool.query(
    `SELECT s.id, s.codigo,s.cedula,s.nombres,s.apellidos,s.telefono,s.lectura_actual, s.direccion, s.alcantarillado,
    tm.tipo as tipo_medidor,s.estado FROM socios as s,tipo_medidor as tm where tm.id=s.tipo_medidor order by s.apellidos ASC`
  );
  res.json(result);
}
```

Elaborado por: El investigador

- **Agregar socios**

Este método permite el ingreso de nuevos socios en el sistema, valida que el número de cedula del socio no se encuentre repetido.

### Ilustración 73. Agregar socio

```
public async create(req: Request, res: Response): Promise<void> {
  const fecha = new Date();
  req.body.fecha_creacion = fecha;
  req.body.fecha_modificacion = fecha;
  const result = await pool.query(
    `SELECT * FROM socios where cedula=?`,
    req.body.cedula
  );
  if (result.length > 0) {
    res
      .status(404)
      .json({ message: "El número de cedula ya se encuentra registrado" });
    return;
  }
  try {
    const result = await pool.query(`INSERT INTO socios set ?`, [req.body]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Editar socios**

Este método permite editar los usuarios del sistema.

### Ilustración 74. Editar socio

```
public async update(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  const fecha = new Date();
  req.body.fecha_modificacion = fecha;
  try {
    await pool.query(`UPDATE socios set ? WHERE id = ?`, [req.body, id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Buscar socio**

Este método permite buscar los a los socios según cedula, nombre, apellido, teléfono, dirección y tipo de medidor.

## Ilustración 75. Buscar socio

```
public async buscarSocios(req: Request, res: Response): Promise<void> {
  const { busqueda } = req.body;
  const result = await pool.query(
    `SELECT s.id, s.codigo,s.cedula,s.nombres,s.apellidos,s.telefono,s.lectura_actual, s.direccion, s.alcantarillado,
    tm.tipo as tipo_medidor,s.estado FROM socios as s,tipo_medidor as tm where tm.id=s.tipo_medidor and
    concat(s.codigo,s.nombres,s.apellidos,s.telefono,s.direccion,tm.tipo) LIKE '%${busqueda}%' order by s.apellidos ASC`
  );
  res.json(result);
}
```

Elaborado por: El investigador

- **Listar lecturas**

Este método permite listar las lecturas que se encuentren en el sistema.

## Ilustración 76. Listar lecturas

```
public async list(req: Request, res: Response): Promise<void> {
  const result =
    await pool.query(`SELECT f.id,s.codigo as socio, CONCAT(s.nombres, ' ', s.apellidos) nombres,
    CONCAT(u.nombres, ' ', u.apellidos) usuario_ingreso,u.usuario ,
    f.mes_cobro,f.alcantarillado, f.lectura_anterior, f.lectura_actual, f.total_metros_cubicos,v.valor,
    f.materiales,f.valor_material,f.observacion,f.total,e.estado,f.fecha_creacion,f.fecha_modificacion
    FROM facturas as f INNER JOIN socios as s on(f.socio=s.id)
    INNER join usuarios as u on(f.usuario_ingreso=u.id)
    INNER JOIN valor_m3 as v on(f.valor_m3 = v.id)
    INNER JOIN estado_factura as e on(f.estado=e.id)
    ORDER BY f.id DESC`);
  res.json(result);
}
```

Elaborado por: El investigador

- **Agregar lectura**

Este método permite agregar nuevas lecturas al sistema, valida que en el mes solo se puede ingresar una lectura por socio.

### Ilustración 77. Agregar lectura

```
public async create(req: Request, res: Response): Promise<void> {
  const { socio, mes_cobro } = req.body;
  const result = await pool.query(
    `SELECT * FROM facturas WHERE socio = ? AND mes_cobro = ?`,
    [socio, mes_cobro]
  );

  if (result.length > 0) {
    res
      .status(404)
      .json({
        message: "El socio ya tiene una factura registrada en este mes",
      });
  } else {
    const fecha = new Date();
    req.body.fecha_creacion = fecha;
    req.body.fecha_modificacion = fecha;
    const result = await pool.query("INSERT INTO facturas set ?", [req.body]);
    res.json({ message: "Ok" });
  }
}
```

Elaborado por: El investigador

- **Editar lecturas**

Este método permite editar las lecturas.

### Ilustración 78. Editar lectura

```
public async update(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  const fecha = new Date();
  req.body.fecha_modificacion = fecha;
  try {
    await pool.query("UPDATE facturas set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Eliminar lecturas**

Este método permite eliminar las lecturas

### Ilustración 79. Eliminar lectura

```
public async delete(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  try {
    await pool.query("DELETE FROM facturas WHERE id = ?", [id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Listar facturas**

Este método permite listar las facturas.

### Ilustración 80. Listar facturas

```
public async list(req: Request, res: Response): Promise<void> {
  const result =
    await pool.query(`SELECT cf.id,u.cedula,u.usuario,s.codigo as socio,s.nombres,s.apellidos,
    s.alcantarillado,tp.tipo, cf.multa,cf.total,cf.fecha FROM cabecera_facturas as cf INNER JOIN usuarios as u
    ON(u.id=cf.usuario_ingreso) INNER JOIN socios as s ON(s.id=cf.socio) INNER JOIN tipo_medidor as tp on(tp.id=s.tipo_medidor)
    ORDER BY cf.id DESC`);
  res.json(result);
}
```

Elaborado por: El investigador

- **Agregar factura**

Este método permite agregar nuevas facturas al sistema.

### Ilustración 81. Agregar lectura

```
public async create(req: Request, res: Response): Promise<void> {
  const fecha = new Date();
  req.body.fecha = fecha;
  const result = await pool.query("INSERT INTO cabecera_facturas set ?", [
    req.body,
  ]);
  const result2 = await pool.query(
    `SELECT MAX(id) AS id FROM cabecera_facturas`
  );
  res.json({ id: result2[0].id });
}
```

Elaborado por: El investigador

- **Buscar facturas por fechas**

Este método permite buscar las facturas según la fecha.

Ilustración 82. Buscar facturas por fechas

```
public async getFacturasCabeceraByFechas(  
    req: Request,  
    res: Response  
): Promise<void> {  
    const { fecha } = req.body;  
    const result =  
        await pool.query(`SELECT cf.id,u.cedula,u.usuario,s.codigo as socio,cf.multa,cf.total,cf.fecha FROM cabecera_facturas as cf  
INNER JOIN usuarios as u ON(u.id=cf.usuario_ingreso) INNER JOIN socios as s ON(s.id=cf.socio)  
WHERE f.fecha LIKE '${fecha}%'  
ORDER BY f.id DESC`);  
    res.json(result);  
}
```

Elaborado por: El investigador

- **Buscar lecturas por facturas**

Este método permite buscar las lecturas según la cabecera a la que están ligadas.

Ilustración 83. Buscar lecturas por factura

```
public async getLecturasByCabecera(  
    req: Request,  
    res: Response  
): Promise<void> {  
    const { id } = req.body;  
    const result = await pool.query(  
        `SELECT f.id,s.codigo as socio, CONCAT(s.nombres, ' ', s.apellidos) nombres,  
CONCAT(u.nombres, ' ', u.apellidos)  
usuario_ingreso,  
f.mes_cobro, f.lectura_anterior, f.lectura_actual, f.total_metros_cubicos,v.valor,  
f.materiales,f.valor_material,f.observacion,f.total,e.estado,f.fecha_creacion,f.fecha_modificacion  
FROM facturas as f INNER JOIN socios as s on(f.socio=s.id)  
INNER join usuarios as u on(f.usuario_ingreso=u.id)  
INNER JOIN valor_m3 as v on(f.valor_m3 = v.id)  
INNER JOIN estado_factura as e on(f.estado=e.id)  
WHERE f.id_cabecera=?  
ORDER BY f.id DESC`,  
        [id]  
    );  
    res.json(result);  
}
```

Elaborado por: El investigador



- **Buscar lectura por socio**

Este método permite buscar las lecturas según un socio indicado.

Ilustración 84. Buscar lecturas por socio

```
public async getLecturasBySocio(req: Request, res: Response): Promise<void> {
  const { socio } = req.body;
  const result =
    await pool.query(`SELECT cf.id,u.cedula,u.usuario,s.codigo as socio,cf.multa,cf.total,cf.fecha FROM cabecera_facturas as cf
INNER JOIN usuarios as u ON(u.id=cf.usuario_ingreso) INNER JOIN socios as s ON(s.id=cf.socio)
WHERE s.codigo LIKE '${socio}%'
ORDER BY cf.id DESC`);
  res.json(result);
}
```

Elaborado por: El investigador

- **Buscar lecturas para cobro**

Este método permite buscar las lecturas según el socio y que se encuentren pendientes de pago.

Ilustración 85. Buscar lecturas para cobro

```
public async getLecturasRecaudacion(req: Request, res: Response): Promise<void> {
  const codigo = req.body.codigo;
  const estado = 'Pendiente';
  const result =
    await pool.query(`SELECT f.id,s.codigo as socio, CONCAT(s.nombres, ' ', s.apellidos) nombres,CONCAT(u.nombres, ' ', u.apellidos) usuario_ingreso,u.usuario,
f.mes_cobro, f.lectura_anterior, f.lectura_actual, f.total_metros_cubicos,v.valor,
f.materiales,f.valor_material,f.observacion,f.total,e.estado,f.fecha_creacion,f.fecha_modificacion
FROM facturas as f INNER JOIN socios as s on(f.socio=s.id)
INNER join usuarios as u on(f.usuario_ingreso=u.id)
INNER JOIN valor_m3 as v on(f.valor_m3 = v.id)
INNER JOIN estado_factura as e on(f.estado=e.id)
WHERE s.codigo = ${codigo} AND e.estado = ${estado}
ORDER BY f.id DESC`, [codigo,estado]
);
  res.json(result);
}
```

Elaborado por: El investigador

- **Buscar lecturas por socio**

Este método permite buscar las lecturas según el socio al que se encuentren ligadas.

## Ilustración 86. Buscar lecturas por socio

```
public async getLecturas(req: Request, res: Response): Promise<void> {
  const codigo = req.body.codigo;
  const result =
    await pool.query(`SELECT f.id,s.codigo as socio, CONCAT(s.nombres, ' ', s.apellidos) nombres,CONCAT(u.nombres, ' ', u.apellidos) usuario_ingreso, u.usuario,
    f.mes_cobro, f.lectura_anterior, f.lectura_actual, f.total_metros_cubicos,v.valor,
    f.materiales, f.valor_material, f.observacion, f.total_e_estado, f.fecha_creacion, f.fecha_modificacion
    FROM facturas as f INNER JOIN socios as s on(f.socio=s.id)
    INNER JOIN usuarios as u on(f.usuario_ingreso=u.id)
    INNER JOIN valor_m3 as v on(f.valor_m3 = v.id)
    INNER JOIN estado_factura as e on(f.estado=e.id)
    WHERE concat(s.codigo,e.estado) LIKE '${codigo}%'
    ORDER BY f.id DESC
    `);
  res.json(result);
}
```

Elaborado por: El investigador

- **Listar mortuoria**

Este método permite listar mortuoria.

## Ilustración 87. Listar mortuoria

```
public async list(req: Request, res: Response): Promise<void> {
  const result = await pool.query(
    `Select m.id,u.cedula as usuario_ingreso,s.codigo as socio,CONCAT(s.nombres, ' ', s.apellidos) as nombres ,
    u.usuario, m.fecha from mortuoria as m INNER JOIN usuarios as u ON (m.usuario_ingreso=u.id) INNER JOIN socios as s
    ON (m.socio=s.id) ORDER BY m.id DESC`
  );
  res.json(result);
}
```

Elaborado por: El investigador

- **Agregar mortuoria**

Este método permite agregar la mortuoria, valida que el socio no tenga registrado otro aporte la misma fecha.

## Ilustración 88. Agregar mortuoria

```
public async create(req: Request, res: Response): Promise<void> {
  const query = await pool.query(
    `Select * from mortuoria where fecha =? and socio=?`,
    [req.body.fecha, req.body.socio]
  );
  if (query.length > 0) {
    res
      .status(400)
      .json({
        message: "El socio ya cuenta con aporte a mortuoria en esta fecha",
      });
    return;
  }
  try {
    const result = await pool.query(`INSERT INTO mortuoria set ?`, [
      req.body,
    ]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(400).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Eliminar mortuoria**

Este método permite eliminar la mortuoria

## Ilustración 89. Eliminar mortuoria

```
public async delete(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  try {
    await pool.query(`DELETE FROM mortuoria WHERE id = ?`, [id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(400).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Buscar mortuoria**

Este método permite buscar la mortuoria según el código del socio, nombres o apellidos.

### Ilustración 90. Buscar mortuoria

```
public async buscarMortuoria(req: Request, res: Response): Promise<void> {  
    const {fecha} = req.body;  
    const query = await pool.query(`Select * from mortuoria where fecha =?`, [  
        req.body.fecha,  
    ]);  
    res.json(query);  
}
```

Elaborado por: El investigador

- **Listar cuadros de caja**

Este método permite listar los cuadros de caja.

### Ilustración 91. Listar cuadros de caja

```
public async list(req: Request, res: Response): Promise<void> {  
    const result = await pool.query(  
        `SELECT c.id, u.usuario as usuario_ingreso,u.usuario,CONCAT(u.nombres, ' ', u.apellidos) nombres,c.mortuoria,  
        c.fecha, c.total_recaudado, c.total_cuadre, c.fondo, c.observacion, c.observacion,c.fecha_modificacion FROM cuadro as c  
        INNER JOIN usuarios as u ON(c.usuario_ingreso=u.id) ORDER BY C.ID DESC`  
    );  
    res.json(result);  
}
```

Elaborado por: El investigador

- **Agregar cuadro de caja**

Este método permite agregar un cuadro de caja.

### Ilustración 92. Agregar cuadro de caja

```
public async create(req: Request, res: Response): Promise<void> {  
    const fecha = new Date();  
    req.body.fecha = fecha;  
    req.body.fecha_modificacion = fecha;  
    try {  
        const result = await pool.query(`INSERT INTO cuadro set ?`, [req.body]);  
        res.json({ message: "Ok" });  
    } catch (error) {  
        res.status(404).json({ message: "Error" });  
    }  
}
```

Elaborado por: El investigador

- **Listado compras**

Este método permite listar las compras.

Ilustración 93. Listado de compras

```
public async list(req: Request, res: Response): Promise<void> {
  const result = await pool.query(
    `SELECT c.id, u.usuario as usuario_ingreso,CONCAT(u.nombres, ' ', u.apellidos) nombres,u.usuario, c.ruc,
    c.n,c.proveedor,c.iva0,c.iva12,c.iva,c.total,c.iva,c.total,c.fecha_modificacion,c.fecha FROM compras as c
    INNER JOIN usuarios as u ON(u.id=c.usuario_ingreso) `
  );
  res.json(result);
}
```

Elaborado por: El investigador

- **Agregar compra**

Este método permite agregar compras.

Ilustración 94. Agregar compra

```
public async create(req: Request, res: Response): Promise<void> {
  const fecha = new Date();
  req.body.fecha_modificacion = fecha;
  try {
    const result = await pool.query(`INSERT INTO compras set ?`, [req.body]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Editar compra**

Este método permite editar las compras

### Ilustración 95. Editar compra

```
public async update(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  const fecha = new Date();
  req.body.fecha_modificacion = fecha;
  try {
    await pool.query(`UPDATE compras set ? WHERE id = ?`, [req.body, id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

- **Eliminar compra**

Este método permite eliminar las compras.

### Ilustración 96. Eliminar compra

```
public async delete(req: Request, res: Response): Promise<void> {
  const { id } = req.params;
  try {
    await pool.query(`DELETE FROM compras WHERE id = ?`, [id]);
    res.json({ message: "Ok" });
  } catch (error) {
    res.status(404).json({ message: "Error" });
  }
}
```

Elaborado por: El investigador

#### 3.2.3.2 Métodos del Frontend (Angular - Servicios)

- **Login**

Este método envía las variables usuario y contraseña al back, una vez se valida guarda las credenciales para permitir el acceso al sistema.

Ilustración 97. Login

```
login(authData: User): Observable<UserResponse | void> {
  return this.http.post<UserResponse>(`${environment.api_URL}/auth`, authData)
    .pipe(
      map((res: UserResponse) => {
        this.guardarCredenciales(res);
        this.loggedId.next(true);
        return res;
      }),
      catchError((err) => this.error(err))
    );
}
```

Elaborado por: El investigador

- **Verificación del token**

Este método valida que el token actual no se encuentre expirado.

Ilustración 98. Verificación del token

```
private checkToken(): void {
  const userToken = localStorage.getItem('token');
  const isExpired = helper.isTokenExpired(userToken);
  isExpired ? this.logout() : this.loggedId.next(true);
}
```

Elaborado por: El investigador

- **Guard rutas aplicación.**

Este método protege las rutas en caso de que el usuario no haya iniciado sesión, en caso de que tenga alguna ruta del sistema este método lo redirigirá inmediatamente al Login.

Ilustración 99. Guard rutas aplicación

```
canActivate(): Observable<boolean> {
  return this.authService.isLogget.pipe(
    take(1),
    map((isLogget:boolean) => {
      if(isLogget) {
        return isLogget
      }else{
        this.router.navigate(['/login'])
        return isLogget;
      }
    })
  )
}
```

Elaborado por: El investigador

- **Generar cabecera para las consultas**

Este método obtiene el token y lo agrega a una cabecera http, este token se verifica al momento de utilizar el api para verificar que el usuario tenga los permisos para realizar dicha acción.

Ilustración 100. Generar cabecera para las consultas

```
private createRequestOptions() {
  let token = localStorage.getItem("token");
  let headers = new HttpHeaders({
    "Content-Type": "application/json",
    auth: token,
  });
  return headers;
}
```

Elaborado por: El investigador

- **Listar usuarios**

Este método obtiene el listado de usuarios llamando al api.



Ilustración 101. Listar usuarios

```
getUsuarios(): Observable<Usuario[]> {  
  let options = this.createRequestOptions();  
  return this.http.get<Usuario[]>(`${environment.api_URL}/usuarios`, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigador

- **Agregar usuario**

Este método envía el objeto usuario al api para poder ser guardado.

Ilustración 102. Agregar usuario

```
saveUsuario(usuario: Usuario): any {  
  let options = this.createRequestOptions();  
  return this.http.post<Usuario[]>( ` ${environment.api_URL}/usuarios`,  
    usuario,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigador

- **Editar usuario**

Este método envía el objeto usuario al api para poder ser editado.

Ilustración 103. Editar usuario

```
editUsuario(id: number, usuario: any): any {  
  let options = this.createRequestOptions();  
  return this.http.put(` ${environment.api_URL}/usuarios/${id}`, usuario, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigador

- **Cambiar contraseña**

Este método envía la contraseña anterior y la nueva contraseña al api para poder ser validadas y actualizar la contraseña.

Ilustración 104. Cambiar contraseña

```
cambiarContraseña(cambio: any): any {
  let options = this.createRequestOptions();
  return this.http.post<Usuario[]>(
    `${environment.api_URL}/auth/cambioClave`,
    cambio,
    { headers: options }
  );
}
```

Elaborado por: El investigador

- **Buscar usuario**

Este método envía una variable al api para poder buscar en la lista de usuarios.

Ilustración 105. Buscar usuario

```
buscarUsuario(busqueda: string): any {
  let options = this.createRequestOptions();
  const enviar = {
    busqueda: busqueda,
  };
  return this.http.post<Usuario[]>(
    `${environment.api_URL}/usuarios/buscarUsuario`,
    enviar,
    { headers: options }
  );
}
```

Elaborado por: El investigador

- **Listar socios**

Este método obtiene el listado de socios llamando al api.

Ilustración 106. Listar socios

```
getSocios(): Observable<Socio[]> {
  let options = this.createRequestOptions();
  return this.http.get<Socio[]>(`${environment.api_URL}/socios`, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Agregar socio**

Este método envía el objeto socio al api para poder ser guardado.

Ilustración 107. Agregar socio

```
saveSocios(socio: Socio): any {
  let options = this.createRequestOptions();
  return this.http.post<Socio>(`${environment.api_URL}/socios`, socio, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Editar socio**

Este método envía el objeto socio al api para poder ser editado.

Ilustración 108. Editar socio

```
editSocio(id: number, socio: any): any {
  let options = this.createRequestOptions();
  return this.http.put(`${environment.api_URL}/socios/${id}`, socio, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Buscar socios**

Este método envía una variable al api para poder buscar en la lista de socios.

Ilustración 109. Buscar socios

```
buscarSocios(busqueda: string): any {
  let options = this.createRequestOptions();
  const enviar = {
    busqueda: busqueda,
  };
  console.log(enviar);
  return this.http.post<Socio[]>(
    `${environment.api_URL}/socios/buscarSocios`,
    enviar,
    { headers: options }
  );
}
```

Elaborado por: El investigador

- **Listar lecturas**

Este método obtiene el listado de lecturas llamando al api.

Ilustración 110. Listar lecturas

```
getLecturas(): Observable<Factura[]> {
  let options = this.createRequestOptions();
  return this.http.get<Factura[]>(`${environment.api_URL}/lecturas`, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Agregar lectura**

Este método envía el objeto lectura al api para poder ser guardado.

### Ilustración 111. Agregar lectura

```
saveLectura(lectura: any): any {
  let options = this.createRequestOptions();
  return this.http.post<Factura>(`${environment.api_URL}/lecturas`, lectura, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Editar lectura**

Este método envía el objeto lectura al api para poder ser editado.

### Ilustración 112. Editar lectura

```
editLectura(id: number, lectua: any): any {
  let options = this.createRequestOptions();
  return this.http.put(`${environment.api_URL}/lecturas/${id}`, lectua, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Eliminar lectura**

Este método envía el id de la factura a eliminar al api.

### Ilustración 113. Eliminar lectura

```
deleteLectura(id: number): any {
  let options = this.createRequestOptions();
  return this.http.delete<Factura>(`${environment.api_URL}/lecturas/${id}`, {
    headers: options,
  });
}
```

Elaborado por: El investigador

- **Buscar lecturas para cobro**

Este método obtiene el listado de lecturas pendientes por cada socio.

Ilustración 114. Buscar lectura para cobro

```
getLecturasRecaudacion(codigo: any): Observable<Factura[]> {
  let options = this.createRequestOptions();
  const enviar = {
    codigo: codigo,
  };
  return this.http.post<Factura[]>(
    `${environment.api_URL}/buesquedaFactura/recaudacion`,
    enviar,
    { headers: options }
  );
}
```

Elaborado por: El investigado

- **Buscar lecturas por fechas**

Este método obtiene las lecturas según una fecha indicada.

Ilustración 115. Buscar lectura por fechas

```
getLecturasbyFechas(fecha: string): Observable<Factura[]> {
  let options = this.createRequestOptions();
  const enviar = {
    fecha: fecha,
  };
  return this.http.post<Factura[]>(
    `${environment.api_URL}/buesquedaFactura/fechas`,
    enviar,
    { headers: options }
  );
}
```

Elaborado por: El investigado

- **Listar facturas**

Este método obtiene el listado de facturas llamando al api.

### Ilustración 116. Listar facturas

```
getCabeceraFacturas(): Observable<CabeceraFacturas[]> {  
  let options = this.createRequestOptions();  
  return this.http.get<CabeceraFacturas[]>(  
    `${environment.api_URL}/facturas_cabecera`,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigador

- **Agregar factura**

Este método envía el objeto factura al api para poder ser guardado.

### Ilustración 117. Agregar factura

```
saveCabeceraFactura(cabecera: any): any {  
  let options = this.createRequestOptions();  
  return this.http.post<CabeceraFacturas>(  
    `${environment.api_URL}/facturas_cabecera`,  
    cabecera,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigador

- **Buscar facturas por fechas**

Este método permite buscar facturas por fecha.

### Ilustración 118. Buscar facturas por fechas

```
getFacturasbyFechas(fecha: string): Observable<CabeceraFacturas[]> {  
  let options = this.createRequestOptions();  
  const enviar = {  
    fecha: fecha,  
  };  
  return this.http.post<CabeceraFacturas[]>(  
    `${environment.api_URL}/facturas_cabecera/buscarCabeceraFactura`,  
    enviar,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigador

- **Buscar lecturas por factura**

Este método permite buscar lecturas según la factura a la que se encuentran ligadas.

Ilustración 119. Buscar lecturas por facturas

```
getLecturasbyCabecera(id: number): Observable<Factura[]> {  
  let options = this.createRequestOptions();  
  const enviar = {  
    id: id,  
  };  
  return this.http.post<Factura[]>(  
    `${environment.api_URL}/facturas_cabecera/buscarLecturasbyCabeceras`,  
    enviar,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigado

- **Buscar factura por socio**

Este método permite buscar lecturas según el socio al que se encuentran ligadas.

Ilustración 120. Buscar lecturas por socio

```
getFacturabySocio(socio: string): Observable<CabeceraFacturas[]> {  
  let options = this.createRequestOptions();  
  const enviar = {  
    socio: socio,  
  };  
  return this.http.post<CabeceraFacturas[]>(  
    `${environment.api_URL}/facturas_cabecera/buscarFacturasbySocio`,  
    enviar,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigado

- **Listar mortuoria**

Este método obtiene el listado de mortuoria llamando al api.



### Ilustración 121. Listar mortuoria

```
getMortuorias(): Observable<Mortuoria[]> {  
  let options = this.createRequestOptions();  
  return this.http.get<Mortuoria[]>(`${environment.api_URL}/mortuoria`, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigado

- **Agregar mortuoria**

Este método envía el objeto mortuorio al api para poder ser guardado.

### Ilustración 122. Agregar mortuoria

```
saveMortuoria(mortuoria: any): any {  
  let options = this.createRequestOptions();  
  return this.http.post<Mortuoria>(  
    `${environment.api_URL}/mortuoria`,  
    mortuoria,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigado

- **Eliminar mortuoria**

Este método envía el id de la mortuoria al api para ser eliminado.

### Ilustración 123. Agregar mortuoria

```
deleteMortuoria(id: number): any {  
  let options = this.createRequestOptions();  
  return this.http.delete<Mortuoria>(  
    `${environment.api_URL}/mortuoria/${id}`,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigado

- **Listar cuadro de caja**

Este método obtiene el listado de cuadro de caja llamando al api.

Ilustración 124. Listar cuadro de caja

```
getCuadros(): Observable<Cuadre[]> {  
  let options = this.createRequestOptions();  
  return this.http.get<Cuadre[]>(`${environment.api_URL}/cuadreDeCaja`, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigado

- **Agregar cuadro de caja**

Este método envía el objeto cuadro de caja al api para poder ser guardado.

Ilustración 125. Agregar cuadro de caja

```
saveCuadre(cuadre: any): any {  
  let options = this.createRequestOptions();  
  return this.http.post<Cuadre>(  
    `${environment.api_URL}/cuadreDeCaja`,  
    cuadro,  
    { headers: options }  
  );  
}
```

Elaborado por: El investigado

- **Listar compras**

Este método obtiene el listado de compras de caja llamando al api.

Ilustración 126. Listar compras

```
getCompras(): Observable<Compras[]> {  
  let options = this.createRequestOptions();  
  return this.http.get<Compras[]>(`${environment.api_URL}/compras`, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigado

- **Agregar compra**

Este método envía el objeto compra al api para poder ser guardado.

Ilustración 127. Agregar compra

```
saveCompra(compra: any): any {  
  let options = this.createRequestOptions();  
  return this.http.post<Compras>(`${environment.api_URL}/compras`, compra, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigado

- **Editar compra**

Este método enviar el objeto compra al api para poder ser editado.

Ilustración 128. Editar compra

```
editCompra(id: number, compra: any): any {  
  let options = this.createRequestOptions();  
  return this.http.put(`${environment.api_URL}/compras/${id}`, compra, {  
    headers: options,  
  });  
}
```

Elaborado por: El investigado

- **Eliminar compra**

Este método envía el id de la compra al api para poder ser eliminado.

Ilustración 129. Eliminar compra

```
deleteCompra(id: number): any {
  let options = this.createRequestOptions();
  return this.http.delete<Compras>(`${environment.api_URL}/compras/${id}`, {
    headers: options,
  });
}
```

Elaborado por: El investigado

### 3.2.3.3 Métodos del Frontend (Angular - Interfaces)

- **Abrir modales**

Este método permite abrir las pestañas modales en toda la aplicación.

Ilustración 130. Abrir modales

```
open(content, size): void {
  this.modal = this._modalService.open(content, {
    ariaLabelledBy: "modal-basic-title",
    size: size,
    centered: true,
  });
  this.modal.result.then((result) => {});
}
```

Elaborado por: El investigado

- **Formulario reactivo usuarios**

En este formulario se define los campos que tiene el usuario, además de las validaciones que tiene cada campo. Para que el formulario pueda ser guardado tiene que cumplir con todas las validaciones.

Ilustración 131. Formularios reactivos usuarios

```
usuarioForm = this.fb.group(  
  {  
    cedula: [  
      "",  
      [  
        Validators.required,  
        Validators.minLength(10),  
        Validators.maxLength(10),  
      ],  
    ],  
    nombres: ["", [Validators.required]],  
    apellidos: ["", [Validators.required]],  
    direccion: ["", [Validators.required]],  
    telefono: [  
      "",  
      [  
        Validators.required,  
        Validators.minLength(10),  
        Validators.maxLength(10),  
      ],  
    ],  
    usuario: ["", [Validators.required, Validators.minLength(6)]],  
    clave: ["", [Validators.required, Validators.minLength(6)]],  
    rol: ["", [Validators.required]],  
    estado: ["", [Validators.required]],  
  },  
  { validators: this.validarCedula }  
);
```

Elaborado por: El investigador

- **Cambiar estado del usuario**

Este método permite cambiar el estado del usuario de activo a inactivo.

Ilustración 132. Cambiar estado del usuario

```
async cambiarEstado(id: number, estado: boolean) {
  estado = !estado;
  const modificar = {
    estado: estado,
  };
  Swal.fire({
    title: estado ? "Habilitar Usuario?" : "Deshabilitar Usuario?",
    showCancelButton: true,
    confirmButtonText: "Aceptar",
  }).then((result) => {
    if (result.isConfirmed) {
      this._usuariosService.editUsuario(id, modificar).subscribe(
        (result) => {
          this.getUsuarios();
          //Mostrando alerta de eliminación
          this._notificacion.showNotification(
            "Usuario modificado",
            "success"
          );
          this.cerrarModal();
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
    }
  });
}
```

Elaborado por: El investigado

- **Guardar y modificar usuario**

Este método permite guardar o editar usuarios, si existe un usuario seleccionado se procede a actualizarlo caso contrario se guardará un nuevo usuario.

Ilustración 133. Guardar y modificar usuario

```
saveUsuario() {
  if (this.usuarioSeleccionado === undefined) {
    this._usuariosService.saveUsuario(this.usuarioForm.value).subscribe(
      (result) => {
        this.getUsuarios();
        if (result.message === "Ok") {
          this._notificacion.showNotification(
            "El usuario a sido agregado correctamente",
            "success"
          );
          this.cerrarModal();
          return;
        }
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
  } else {
    this._usuariosService
      .editUsuario(this.usuarioSeleccionado.id, this.usuarioForm.value)
      .subscribe(
        (result) => {
          this.getUsuarios();
          if (result.message === "Ok") {
            this._notificacion.showNotification(
              "El usuario a sido actualizado correctamente",
              "success"
            );
            this.cerrarModal();
            return;
          }
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
  }
}
```

Elaborado por: El investigado

- **Validar cedula**

Este método valida que el número de cedula sea ingresado correctamente.

### Ilustración 134. Validar cedula

```
validarCedula(group: FormGroup): any {
  const cedula = group.controls.cedula?.value;
  if (cedula !== null) {
    if (cedula.length == 10) {
      var digito_region = cedula.substring(0, 2);
      if (digito_region >= 1 && digito_region <= 24) {
        var ultimo_digito = cedula.substring(9, 10);
        var pares =
          parseInt(cedula.substring(1, 2)) +
          parseInt(cedula.substring(3, 4)) +
          parseInt(cedula.substring(5, 6)) +
          parseInt(cedula.substring(7, 8));
        var numero1 = cedula.substring(0, 1);
        numero1 = numero1 * 2;
        if (numero1 > 9) {
          numero1 = numero1 - 9;
        }
        var numero3 = cedula.substring(2, 3);
        numero3 = numero3 * 2;
        if (numero3 > 9) {
          numero3 = numero3 - 9;
        }
        var numero5 = cedula.substring(4, 5);
        numero5 = numero5 * 2;
        if (numero5 > 9) {
          numero5 = numero5 - 9;
        }
        var numero7 = cedula.substring(6, 7);
        numero7 = numero7 * 2;
        if (numero7 > 9) {
          numero7 = numero7 - 9;
        }
        var numero9 = cedula.substring(8, 9);
        numero9 = numero9 * 2;
        if (numero9 > 9) {
          numero9 = numero9 - 9;
        }
        var impares = numero1 + numero3 + numero5 + numero7 + numero9;
        var suma_total = pares + impares;
        var primer_digito_suma = String(suma_total).substring(0, 1);
        var decena = (parseInt(primer_digito_suma) + 1) * 10;
        var digito_validador = decena - suma_total;
        if (digito_validador == 10) var digito_validador = 0;
        if (digito_validador == ultimo_digito) {
          return null;
        } else {
          return { invalido: true };
        }
      } else {
        return { invalido: true };
      }
    } else {
      return { invalido: true };
    }
  }
}
```

Elaborado por: El investigado

- **Buscar usuario**

Este método llama al servicio buscar usuario y envía la variable buscar para filtrar los usuarios.



Ilustración 135. Buscar usuario

```
buscarUsuarios() {
  if (this.buscar === "") {
    this.usuariosFiltrados = this.usuarios;
    return;
  }
  this._usuariosService.buscarUsuario(this.buscar).subscribe((result) => {
    this.usuariosFiltrados = result;
  });
}
```

Elaborado por: El investigador

- **Formulario reactivo socios**

En este formulario se define los campos que tiene el socio, además de las validaciones que tiene cada campo. Para que el formulario pueda ser guardado tiene que cumplir con todas las validaciones.

Ilustración 136. Formulario reactivo socios

```
socioForm = this.fb.group({
  codigo: [
    "",
    [Validators.required, Validators.minLength(5), Validators.maxLength(5)],
  ],
  cedula: [
    "",
    [
      Validators.required,
      Validators.minLength(10),
      Validators.maxLength(10),
    ],
  ],
  nombres: ["", [Validators.required]],
  apellidos: ["", [Validators.required]],
  direccion: ["", [Validators.required]],
  telefono: [
    "",
    [
      Validators.required,
      Validators.minLength(10),
      Validators.maxLength(10),
    ],
  ],
  alcantarillado: ["", [Validators.required]],
  tipo_medidor: ["", [Validators.required]],
  lectura_actual: ["", [Validators.required]],
  estado: ["", [Validators.required]],
}, { validators: this.validarCedula });
```

Elaborado por: El investigador

- **Guardar y editar socio**

Este método permite guardar o editar socios, si existe un socio seleccionado se procede a actualizarlo caso contrario se guardará un nuevo socio.

Ilustración 137. Guardar y editar socios

```
saveSocio() {
  if (this.socioSeleccionado === undefined) {
    this._sociosService.saveSocios(this.socioForm.value).subscribe(
      (result) => {
        this.getSocios();
        if (result.message === "Ok") {
          this._notificacion.showNotification(
            "El socio a sido agregado correctamente",
            "success"
          );
          this.cerrarModal();
          return;
        }
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
  } else {
    this._sociosService
      .editSocio(this.socioSeleccionado.id, this.socioForm.value)
      .subscribe(
        (result) => {
          this.getSocios();
          if (result.message === "Ok") {
            this._notificacion.showNotification(
              "El socio a sido actualizado correctamente",
              "success"
            );
            this.cerrarModal();
            return;
          }
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
  }
}
```

Elaborado por: El investigador

- **Cambiar estado del socio**

Este método permite cambiar el estado del socio de activo a inactivo.

Ilustración 138. Cambiar estado del socio

```
async cambiarEstado(id: number, estado: boolean) {
  estado = !estado;
  const modificar = {
    estado: estado,
  };

  Swal.fire({
    title: estado ? "Habilitar Socio?" : "Deshabilitar Socio?",
    showCancelButton: true,
    confirmButtonText: "Aceptar",
  }).then((result) => {
    if (result.isConfirmed) {
      this._sociosService.editSocio(id, modificar).subscribe(
        (result) => {
          this.getSocios();
          this._notificacion.showNotification("Socio modificado", "success");
          this.cerrarModal();
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
    }
  });
}
```

Elaborado por: El investigador

- **Buscar socio**

Este método llama al servicio buscar socios y envía la variable buscar para filtrar los socios.

Ilustración 139. Buscar socio

```
buscarSocios() {
  if (this.buscar === "") {
    this.soscioFiltrados = this.socios;
    return;
  }
  this._sociosService.buscarSocios(this.buscar).subscribe((result) => {
    this.soscioFiltrados = result;
  });
}
```

Elaborado por: El investigador

- **Formulario reactivo lecturas**

En este formulario se define los campos que tiene el socio, además de las validaciones que tiene cada campo. Para que el formulario pueda ser guardado tiene que cumplir con todas las validaciones.

Ilustración 140. Formulario reactivo lecturas

```
lecturasForm = this.fb.group({
  socio: [
    "",
    [Validators.required, Validators.minLength(5), Validators.maxLength(5)],
  ],
  nombres: ["", Validators.required],
  apellidos: ["", Validators.required],
  mes_cobro: ["", [Validators.required]],
  lectura_anterior: ["", [Validators.required]],
  lectura_actual: ["", [Validators.required]],
  total_metros_cubicos: ["", Validators.required],
  valor_m3: ["", [Validators.required]],
  alcantarillado: ["", [Validators.required]],
  tipo_medidor: ["", [Validators.required]],
  materiales: ["", [Validators.required]],
  valor_material: [""],
  observacion: [""],
  total: ["", [Validators.required]],
  estado: ["", ],
});
```

Elaborado por: El investigado

- **Guardar y editar lecturas**

Este método permite guardar o editar lecturas, si existe una lectura seleccionada se procede a actualizarla caso contrario se guardará una nueva lectura.

## Ilustración 141. Guardar y editar lectura

```
guardarLectura() {
  if (this.lecturaSeleccionada === undefined) {
    const lecturaGuardar = {
      socio: this.idSocio,
      usuario_ingreso: localStorage.getItem("userId"),
      mes_cobro: this.lecturasForm.get("mes_cobro").value,
      lectura_anterior: this.lecturasForm.get("lectura_anterior").value,
      lectura_actual: this.lecturasForm.get("lectura_actual").value,
      total_metros_cubicos: this.lecturasForm.get("total_metros_cubicos").value,
      valor_m3: this.idValorM3,
      alcantarillado:
        this.lecturasForm.get("alcantarillado").value === "Si" ? 1 : 0,
      materiales: this.lecturasForm.get("materiales").value,
      valor_material: this.lecturasForm.get("valor_material").value,
      observacion: this.lecturasForm.get("observacion").value,
      total: this.lecturasForm.get("total").value,
      estado: this.lecturasForm.get("estado").value,
    };
    this._lecturaService.savetLectura(lecturaGuardar).subscribe(
      (result) => {
        this.getLecturas();
        if (result.message === "Ok") {
          this._notificacion.showNotification(
            "La factura a sido agregada correctamente",
            "success"
          );
          const actualizarMedicion = {
            lectura_actual: this.lecturasForm.get("lectura_actual").value,
          };
          this._sociosService
            .editSocio(this.idSocio, actualizarMedicion)
            .subscribe(() => {});
          this.siguiente();
          return;
        }
      },
      (err) => {
        if (
          err.error.message ===
          "El socio ya tiene una factura registrada en este mes"
        ) {
          Swal.fire({
            title: err.error.message,
            icon: "error",
            confirmButtonText: "Aceptar",
          });
        } else {
          this._notificacion.mensajeError(err);
        }
      }
    );
  } else {
    const lecturaGuardar = {
      observacion: this.lecturasForm.get("observacion").value
    };
    this._lecturaService
      .editLectura(this.lecturaSeleccionada.id, lecturaGuardar)
      .subscribe(
        (result) => {
          this.getLecturas();
          if (result.message === "Ok") {
            this._notificacion.showNotification(
              "La factura a sido actualizada correctamente",
              "success"
            );
            const actualizarMedicion = {
              lectura_actual: this.lecturasForm.get("lectura_actual").value,
            };
          }
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
  }
}
```

Elaborado por: El investigador

- **Siguiente lectura**

Este método permite cambiar al siguiente socio en la lista.

Ilustración 142. Siguiente lectura

```
siguiente() {
  const mes = this.lecturasForm.get("mes_cobro").value;
  const estado = this.lecturasForm.get("estado").value;
  this.lecturasForm.reset();
  this.idValorm3 = null;
  this.lecturasForm.controls.mes_cobro.setValue(mes);
  this.lecturasForm.controls.estado.setValue(estado);
  this.lecturasForm.controls.materiales.setValue(false);

  this._sociosService
    .getSocio(String(Number(this.idSocio) + Number(1)))
    .subscribe(
      (result) => {
        if (!result.estado) {
          this.siguiente();
        } else {
          this.idSocio = result.id;
          let idMedidor;
          this.lecturasForm.controls.socio.setValue(result.codigo);
          this.lecturasForm.controls.nombres.setValue(result.nombres);
          this.lecturasForm.controls.apellidos.setValue(result.apellidos);
          this.lecturasForm.controls.alcantarillado.setValue(
            result.alcantarillado ? "Si" : "No"
          );
          this.lecturasForm.controls.lectura_anterior.setValue(
            result.lectura_actual
          );
          this.lecturasForm.controls.tipo_medidor.setValue(
            result.tipo_medidor
          );
          this.tipo_medidor.forEach((element) => {
            if (element.tipo === result.tipo_medidor) {
              idMedidor = element.id;
            }
          });

          for (let index = 0; index < this.valorm3.length; index++) {
            if (this.valorm3[index].tipo_medidor === idMedidor) {
              this.lecturasForm.controls.valor_m3.setValue(
                this.valorm3[index].valor,
                (this.idValorm3 = this.valorm3[index].id)
              );
            }
          }
        }
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
}
```

Elaborado por: El investigado

- **Lectura anterior**

Este método permite cambiar al anterior socio en la lista.

Ilustración 143. Lectura anterior

```
anterior() {
  const mes = this.lecturasForm.get("mes_cobro").value;
  const estado = this.lecturasForm.get("estado").value;
  this.lecturasForm.reset();
  this.idValorm3 = null;
  this.lecturasForm.controls.mes_cobro.setValue(mes);
  this.lecturasForm.controls.estado.setValue(estado);
  this.lecturasForm.controls.materiales.setValue(false);

  this._sociosService
    .getSocio(String(Number(this.idSocio) - Number(1)))
    .subscribe(
      (result) => {
        if (!result.estado) {
          this.siguiete();
        } else {
          this.idSocio = result.id;
          let idMedidor;
          this.lecturasForm.controls.socio.setValue(result.codigo);
          this.lecturasForm.controls.nombres.setValue(result.nombres);
          this.lecturasForm.controls.apellidos.setValue(result.apellidos);
          this.lecturasForm.controls.alcantarillado.setValue(
            result.alcantarillado ? "Si" : "No"
          );
          this.lecturasForm.controls.lectura_anterior.setValue(
            result.lectura_actual
          );
          this.lecturasForm.controls.tipo_medidor.setValue(
            result.tipo_medidor
          );
          this.tipo_medidor.forEach((element) => {
            if (element.tipo === result.tipo_medidor) {
              idMedidor = element.id;
            }
          });

          for (let index = 0; index < this.valorm3.length; index++) {
            if (this.valorm3[index].tipo_medidor === idMedidor) {
              this.lecturasForm.controls.valor_m3.setValue(
                this.valorm3[index].valor,
                (this.idValorm3 = this.valorm3[index].id)
              );
            }
          }
        }
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
}
```

Elaborado por: El investigado

- **Buscar socio**

Este método permite buscar el socio para agregar una nueva lectura.

Ilustración 144. Buscar socio

```
buscarSocio() {
  const socio = this.lecturasForm.get("socio").value;
  if (socio.length === 5) {
    this._sociosService.getSocio(socio).subscribe(
      (result) => {
        if (!result.estado) {
          this.UsuarioDeshabilitado();
          return;
        }
        this.idSocio = result.id;
        let idMedidor;
        this.lecturasForm.controls.nombres.setValue(result.nombres);
        this.lecturasForm.controls.apellidos.setValue(result.apellidos);

        this.lecturasForm.controls.lectura_anterior.setValue(
          result.lectura_actual
        );
        this.lecturasForm.controls.tipo_medidor.setValue(result.tipo_m
        this.tipo_medidor.forEach((element) => {
          if (element.tipo === result.tipo_medidor) {
            idMedidor = element.id;
          }
        });
        this.lecturasForm.controls.alcantarillado.setValue(
          result.alcantarillado ? "Si" : "No"
        );
        for (let index = 0; index < this.valorm3.length; index++) {
          if (this.valorm3[index].tipo_medidor === idMedidor) {
            this.lecturasForm.controls.valor_m3.setValue(
              this.valorm3[index].valor,
              (this.idValorm3 = this.valorm3[index].id)
            );
          }
        }
        result.alcantarillado
          ? this.lecturasForm.controls.total.setValue(1)
          : "";
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
  } else {
    const mes = this.lecturasForm.get("mes_cobro").value;
    const estado = this.lecturasForm.get("estado").value;
    const socio = this.lecturasForm.get("socio").value;
    this.lecturasForm.reset();
    this.lecturasForm.controls.socio.setValue(socio);
    this.lecturasForm.controls.materiales.setValue(false);
    this.lecturasForm.controls.mes_cobro.setValue(mes);
    this.lecturasForm.controls.estado.setValue(estado);
  }
}
```

Elaborado por: El investigado



- **Calcular totales**

Este método permite calcular el total que debe pagar el socio por el mes de consumo.

Ilustración 145. Calcular totales

```
calcularTotales() {
  const actual = this.lecturasForm.get("lectura_actual").value;
  const anterior = this.lecturasForm.get("lectura_anterior").value;
  const alcantarillado = this.lecturasForm.get("alcantarillado").value;
  const valorM3 = this.lecturasForm.get("valor_m3").value;
  const materiales = this.lecturasForm.get("materiales").value;
  let totalFactura = 0;
  let totalM3 = 0;

  alcantarillado ? (totalFactura += 1) : "";

  if (actual != "" && anterior != "") {
    if (Number(actual) > Number(anterior)) {
      totalM3 = actual - anterior;
      this.lecturasForm.controls.total_metros_cubicos.setValue(totalM3);
      if (totalM3 <= 10) {
        totalFactura += 3;
      } else {
        totalM3 = totalM3 - 10;
        totalFactura += totalM3 * valorM3 + 3;
      }
    } else {
      this.lecturasForm.controls.total_metros_cubicos.setValue("");
    }
  }
  materiales
  ? (totalFactura += Number(this.lecturasForm.get("valor_material").value))
  : "";
  this.lecturasForm.controls.total.setValue(totalFactura.toFixed(2));
}
```

Elaborado por: El investigado

- **Eliminar lectura**

Este método permite eliminar la lectura.

Ilustración 146. Eliminar lectura

```
eliminarFactura(id: number) {
  Swal.fire({
    title: "Eliminar lectura",
    text: "La lectura se eliminara permanentemente de la base de datos",
    showCancelButton: true,
    confirmButtonText: "Aceptar",
  }).then((result) => {
    if (result.isConfirmed) {
      this._lecturaService.deleteLectura(id).subscribe(
        (result) => {
          this._notificacion.showNotificacion("Lecturas eliminada", "success");
          this.getLecturas();
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
    }
  });
}
```

Elaborado por: El investigado

- **Buscar lecturas**

Este método permite filtrar las lecturas por código de socio y fecha.

Ilustración 147. Buscar lecturas

```
buscarLecturas() {  
  if (this.buscar == "") {  
    this.lecturasFiltradas = this.lecturas;  
    this.buscarPorFechas();  
    return;  
  }  
  const enviar = {  
    codigo: this.buscar,  
  };  
  this._buscarLecturasService.getLecturas(enviar).subscribe((result) => {  
    this.lecturasFiltradas = result;  
    this.buscarPorFechas();  
  });  
}
```

Elaborado por: El investigado

- **Generar PDF**

Este método permite generar los pdf del sistema.

Ilustración 148. Generar PDF

```
getPDF() {  
  var HTML_Width = $(".canvas_div_pdf").width();  
  var HTML_Height = $(".canvas_div_pdf").height();  
  var top_left_margin = 0;  
  var PDF_Width = HTML_Width + top_left_margin * 2;  
  var PDF_Height = PDF_Width * 1.5 + top_left_margin * 2;  
  var canvas_image_width = HTML_Width;  
  var canvas_image_height = HTML_Height;  
  
  var totalPDFPages = Math.ceil(HTML_Height / PDF_Height) - 1;  
  
  html2canvas($(".canvas_div_pdf")[0], { allowTaint: true }).then(function ( canvas )  
  ) {  
    canvas.getContext("2d");  
  
    var imgData = canvas.toDataURL("image/jpeg", 1.0);  
    var pdf = new jsPDF("p", "pt", [PDF_Width, PDF_Height]);  
    pdf.addImage(  
      imgData,  
      "JPG",  
      top_left_margin,  
      top_left_margin,  
      canvas_image_width,  
      canvas_image_height  
    );  
  
    for (var i = 1; i <= totalPDFPages; i++) {  
      pdf.addPage(PDF_Width, PDF_Height);  
      pdf.addImage(  
        imgData,  
        "JPG",  
        top_left_margin,  
        -(PDF_Height * i) + top_left_margin * 4,  
        canvas_image_width,  
        canvas_image_height  
      );  
    }  
    const fecha = new Date().toLocaleDateString();  
    const nombre = "Compra - " + fecha;  
    pdf.save(nombre);  
  });  
}
```

Elaborado por: El investigado

- **Buscar lecturas por fechas**

Este método permite filtrar las lecturas por fechas.

Ilustración 149. Buscar lecturas por fechas

```
buscarPorFechas() {  
  if (this.fechaInicio != undefined && this.fechaFin != undefined) {  
    this.lecturasFiltradas = this.lecturasFiltradas.filter(  
      (element) =>  
        new Date(element.mes_cobro) >= new Date(this.fechaInicio) &&  
        new Date(element.mes_cobro) <= new Date(this.fechaFin)  
    );  
  } else if (this.fechaInicio != undefined) {  
    this.lecturasFiltradas = this.lecturasFiltradas.filter(  
      (element) => new Date(element.mes_cobro) >= new Date(this.fechaInicio)  
    );  
  } else if (this.fechaFin != undefined) {  
    this.lecturasFiltradas = this.lecturasFiltradas.filter(  
      (element) => new Date(element.mes_cobro) <= new Date(this.fechaFin)  
    );  
  }  
}
```

Elaborado por: El investigado

- **Buscar mortuoria por socio**

Este método permite filtrar las mortuorias por socios.

Ilustración 150. Buscar mortuoria por socios

```
buscarM(){  
  if (this.buscar == "") {  
    this.mortuoriaFiltrada = this.mortuoria;  
    this.buscarMortuoriaFecha();  
    return;  
  }  
  
  this._mortuoriaService.buscarMortuoriaPorSocio(this.buscar).subscribe((result) => {  
    this.mortuoriaFiltrada = result;  
    this.buscarMortuoriaFecha();  
  });  
}
```

Elaborado por: El investigado

- **Buscar mortuoria por fechas**

Este método permite filtrar las mortuorias por fechas.

Ilustración 151. Buscar mortuoria por fechas

```
buscarMortuoriaFecha() {
  if (this.fechaInicio !== undefined && this.fechaFin !== undefined) {
    this.mortuoriaFiltrada = this.mortuoria.filter(
      (element) =>
        new Date(element.fecha.toString().substring(0,10)) >= new Date(this.fechaInicio) &&
        new Date(element.fecha.toString().substring(0,10)) <= new Date(this.fechaFin)
    );
  } else if (this.fechaInicio !== undefined) {
    this.mortuoriaFiltrada = this.mortuoria.filter(
      (element) => new Date(element.fecha.toString().substring(0,10)) >= new Date(this.fechaInicio)
    );
  } else if (this.fechaFin !== undefined) {
    this.mortuoriaFiltrada = this.mortuoria.filter(
      (element) => new Date(element.fecha.toString().substring(0,10)) <= new Date(this.fechaFin)
    );
  }
}
```

Elaborado por: El investigado

- **Guardar mortuoria**

Este método permite guardar el registro de mortuoria.

Ilustración 152. Guardar mortuoria

```
guardarMortuoria() {
  const mortuoria: Mortuoria = {
    usuario_ingreso: localStorage.getItem("userId"),
    socio: this.idSocio,
    fecha: this.mortuoriaForm.get("fecha").value,
  };
  this._mortuoriaService.saveMortuoria(mortuoria).subscribe(
    (result) => {
      this.getMortuoria();
      this.cerrarModal();
    },
    (err) => {
      this._notificacion.mensajeError(err);
    }
  );
}
```

Elaborado por: El investigado

- **Buscar socio mortuoria**

Este método permite buscar el socio para el registro de mortuoria.

Ilustración 153. Buscar socio mortuoria

```
buscarSocio() {
  const socio = this.mortuoriaForm.get("socio").value;
  if (socio.length === 5) {
    this._sociosService.getSocio(socio).subscribe(
      (result) => {
        if (!result.estado) {
          this.UsuarioDeshabilitado();
          return;
        }
        this.idSocio = result.id;
        this.mortuoriaForm.controls.nombres.setValue(result.nombre);
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
  } else {
    this.mortuoriaForm.controls.nombres.setValue("");
  }
}
```

Elaborado por: El investigado

- **Eliminar mortuoria**

Este método permite eliminar el registro de mortuoria.

Ilustración 154. Eliminar mortuoria

```
eliminarMortuoria(id: number) {
  Swal.fire({
    title: "Eliminar mortuoria",
    text: "El registro se eliminara permanentemente de la base de datos",
    showCancelButton: true,
    confirmButtonText: "Aceptar",
  }).then((result) => {
    if (result.isConfirmed) {
      this._mortuoriaService.deleteMortuoria(id).subscribe(
        (result) => {
          this._notificacion.showNotification(
            "Registro Eliminado",
            "success"
          );
          this.getMortuoria();
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
    }
  });
}
```

Elaborado por: El investigado

- **Formulario reactivo cuadro de caja**

En este formulario se define los campos que tiene el cuadro de caja, además de las validaciones que tiene cada campo. Para que el formulario pueda ser guardado tiene que cumplir con todas las validaciones.

Ilustración 155. Formulario reactivo cuadro de caja

```
cuadre = this.fb.group(  
{  
  fecha: [""],  
  ciend: [""],  
  totalciend: [""],  
  cincuentad: [""],  
  totalcincuentad: [""],  
  veinted: [""],  
  totalveinted: [""],  
  diesd: [""],  
  totaldiesd: [""],  
  cincod: [""],  
  totalcincod: [""],  
  und: [""],  
  totalund: [""],  
  unm: [""],  
  totalunm: [""],  
  cincuentam: [""],  
  totalcincuentam: [""],  
  venteycinco: [""],  
  totalventeycinco: [""],  
  diesm: [""],  
  totaldiesm: [""],  
  cincom: [""],  
  totalcincom: [""],  
  unom: [""],  
  totalunom: [""],  
  totaldia: [""],  
  totalcuadre: [""],  
  observacion: [""],  
  fondoCaja: [""],  
  mortuoria: [],  
},  
{ validators: this.validarCuadre }  
);
```

Elaborado por: El investigado

- **Verificar cuadro de caja**

Este método verifica que el total recaudado en el día sea igual al total registrado en sistema.

Ilustración 156. Verificar cuadro de caja

```
verificar(): boolean {
  const totaldia: number = this.cuadre.get("totaldia").value;
  const totalcuadre: number = this.cuadre.get("totalcuadre").value;
  if (totaldia !== totalcuadre) {
    return false;
  } else {
    return true;
  }
}
```

Elaborado por: El investigado

- **Obtener total recaudado en el día**

Este método obtiene el total que se recaudó en el día.

Ilustración 157. Verificar cuadro de caja

```
getTotalRecaudacion() {
  let acumulador = 0;
  const fecha: string = this.cuadre.get("fecha").value;

  this._mortuoriaService.buscarMortuoria(fecha).subscribe((result) => {
    this.cuadre.controls.mortuoria.setValue(result.length);
  });

  this._cabeceraFacturaService
    .getFacturasbyFechas(fecha)
    .subscribe((result) => {
      result.forEach((element) => {
        acumulador += Number(element.total);
      });
      this.cuadre.controls.totaldia.setValue(acumulador);
    });
}
```

Elaborado por: El investigado

- **Guardar cuadro de caja**

Este método permite guardar el nuevo registro de cuadro de caja.

Ilustración 158. Guardar cuadro de caja

```
guardar() {
  const totalRecaudado = this.cuadre.get("totaldia").value;
  if (totalRecaudado === 0 || totalRecaudado === "") {
    return Swal.fire({
      title: "No se encuentran registros de esta fecha",
      icon: "error",
      confirmButtonText: "Aceptar",
    });
  }
  const usuario = localStorage.getItem("userId");

  const totalCuadre = this.cuadre.get("totalcuadre").value;
  const fondo = this.cuadre.get("fondoCaja").value;
  const observacion = this.cuadre.get("observacion").value;
  const mortuoria = this.cuadre.get("mortuoria").value;
  const cuadro = {
    usuario_ingreso: usuario,
    total_recaudado: totalRecaudado,
    total_cuadre: totalCuadre,
    fondo: fondo,
    mortuoria: mortuoria,
    observacion: observacion,
  };
  if (!this.verificar()) {
    Swal.fire({
      title: "La caja no se encuentra cuadrada decaea continuar ? ",
      showCancelButton: true,
      confirmButtonText: "Aceptar",
    }).then((result) => {
      if (result.isConfirmed) {
        this._cuadreService.saveCuadre(cuadre).subscribe((result) => {
          this.cerrarModal();
        });
      }
    });
  } else {
    this._cuadreService.saveCuadre(cuadre).subscribe((result) => {
      this.cerrarModal();
    });
  }
}
```

Elaborado por: El investigado



- **Buscar cuadro de caja**

Este método permite filtrar los cuadros de caja por fechas.

Ilustración 159. Buscar cuadro de caja

```
buscarCuadroDeCaja() {
  if (this.fechaInicio != undefined && this.fechaFin != undefined) {
    this.cuadrosFiltrados = this.cuadros.filter(
      (element) =>
        new Date(element.fecha.toString().substring(0,10)) >= new Date(this.fechaInicio) &&
        new Date(element.fecha.toString().substring(0,10)) <= new Date(this.fechaFin)
    );
  } else if (this.fechaInicio != undefined) {
    this.cuadrosFiltrados = this.cuadros.filter(
      (element) => new Date(element.fecha.toString().substring(0,10)) >= new Date(this.fechaInicio)
    );
  } else if (this.fechaFin != undefined) {
    this.cuadrosFiltrados = this.cuadros.filter(
      (element) => new Date(element.fecha.toString().substring(0,10)) <= new Date(this.fechaFin)
    );
  }
}
```

Elaborado por: El investigado

- **Formulario reactivo compras**

En este formulario se define los campos que tiene una compra, además de las validaciones que tiene cada campo. Para que el formulario pueda ser guardado tiene que cumplir con todas las validaciones.

Ilustración 160. Formulario reactivo compras

```
comprasForm = this.fb.group({
  fecha: [""], [Validators.required],
  usuario_ingreso: [""],
  n: [""], [Validators.required],
  ruc: [""], [Validators.required],
  proveedor: [""], [Validators.required],
  iva0: [""],
  iva12: [""],
  iva: [""], [Validators.required],
  total: [""], [Validators.required],
});
```

Elaborado por: El investigado

- **Guardar y editar compras**

Este método permite guardar o editar las compras, si existe una compra seleccionada se procede a actualizarla caso contrario se guardará una nueva compra.

Ilustración 161. Guardar y editar compras

```
guardarCompras() {
  this.comprasForm.controls.usuario_ingreso.setValue(
    localStorage.getItem("userId")
  );
  if (this.compraSeleccionada === undefined) {
    this._comprasService.saveCompra(this.comprasForm.value).subscribe(
      (result) => {
        this._notificacion.showNotification("Compra guardada", "success");
        this.getCompras();
        this.cerrarModal();
      },
      (err) => {
        this._notificacion.mensajeError(err);
      }
    );
  } else {
    this._comprasService
      .editCompra(this.compraSeleccionada.id, this.comprasForm.value)
      .subscribe((result) => {
        this._notificacion.showNotification("Compra editada", "success");
        this.getCompras();
        this.cerrarModal();
      });
  }
}
```

Elaborado por: El investigador

- **Eliminar compra**

Este método permite eliminar las compras registradas.

Ilustración 162. Eliminar compra

```
eliminar(id: number) {
  Swal.fire({
    title: "Eliminar Compra",
    text: "La compra se eliminara permanentemente",
    showCancelButton: true,
    confirmButtonText: "Aceptar",
  }).then((result) => {
    if (result.isConfirmed) {
      this._comprasService.deleteCompra(id).subscribe(
        (result) => {
          this._notificacion.showNotification("Compra eliminada", "success");
          this.getCompras();
          this.cerrarModal();
        },
        (err) => {
          this._notificacion.mensajeError(err);
        }
      );
    }
  });
}
```

Elaborado por: El investigado

- **Calcular total compra**

Este método permite calcular el total de la compra.

Ilustración 163. Calcular total compra

```
calculoTotales() {
  let acumulador: number = 0;
  const iva0: string = this.comprasForm.get("iva0").value;
  const iva12: string = this.comprasForm.get("iva12").value;
  iva0 != "" ? (this.disable12 = true) : (this.disable12 = false);
  iva12 != "" ? (this.disable0 = true) : (this.disable0 = false);
  iva0 != "" ? (acumulador += Number(iva0)) : "";
  const iva = Number(iva12) * 0.12;
  iva12 != "" ? (acumulador += iva + Number(iva12)) : "";
  this.comprasForm.controls.iva.setValue(iva.toFixed(2));
  this.comprasForm.controls.total.setValue(acumulador.toFixed(2));
  console.log(this.disable0, this.disable12);
}
```

Elaborado por: El investigado

- **Buscar compras**

Este método permite filtrar las compras por fechas.

Ilustración 164. Buscar compras

```
buscarCompras() {
  if (this.fechaInicio != undefined && this.fechaFin != undefined) {
    this.comprasFiltradas = this.compras.filter(
      (element) =>
        new Date(element.fecha.toString().substring(0,10)) >= new Date(this.fechaInicio) &&
        new Date(element.fecha.toString().substring(0,10)) <= new Date(this.fechaFin)
    );
  } else if (this.fechaInicio != undefined) {
    this.comprasFiltradas = this.compras.filter(
      (element) => new Date(element.fecha.toString().substring(0,10)) >= new Date(this.fechaInicio)
    );
  } else if (this.fechaFin != undefined) {
    this.comprasFiltradas = this.compras.filter(
      (element) => new Date(element.fecha.toString().substring(0,10)) <= new Date(this.fechaFin)
    );
  }
}
```

Elaborado por: El investigado

### 3.2.4 Fase IV: Pruebas

Las pruebas son una herramienta necesaria en el desarrollo de software debido a que indica si el software desarrollado se encuentra en óptimas condiciones para ser introducido a un ambiente de producción. Estas pruebas son necesarias para que tanto los desarrolladores como los usuarios finales estén conformes con el trabajo realizado y saber si el software cumple con las indicaciones descritas en las historias de usuario.

Tabla 98. Prueba de aceptación 001

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 001
<b>Identificador de Historia:</b> H012
<b>Denominación:</b> Inicio de sesión
<b>Descripción:</b> El usuario del sistema podrá acceder a la aplicación mediante autenticación de usuario y contraseña.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una pantalla de inicio de sesión con los respectivos campos necesarios como usuario y contraseña.
<b>Interfaz:</b> La pantalla validara los campos necesarios para el inicio de sesión los cuales con usuario y contraseña.
<b>Resultado esperado:</b> En caso de ser correctos permitirá el acceso a la aplicación caso contrario mostrará un mensaje en pantalla de error de contraseña o usuario
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 99. Prueba de aceptación 002

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 002
<b>Identificador de Historia:</b> H013
<b>Denominación:</b> Listado de usuarios
<b>Descripción:</b> El usuario del sistema podrá listar los usuarios que se encuentren registrados en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una lista de usuarios registrados.
<b>Interfaz:</b> La aplicación mostrara el listado en una tabla.
<b>Resultado esperado:</b> Listar los usuarios del sistema
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 100. Prueba de aceptación 003

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 003
<b>Identificador de Historia:</b> H014
<b>Denominación:</b> Creación de usuarios
<b>Descripción:</b> El usuario del sistema podrá crear nuevos.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá agregar nuevos usuarios.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la creación del nuevo usuario.
<b>Resultado esperado:</b> Agregar nuevos usuarios.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 101. Prueba de aceptación 004

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 004
<b>Identificador de Historia:</b> H015
<b>Denominación:</b> Edición de usuarios
<b>Descripción:</b> El usuario del sistema podrá editar los usuarios existentes.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá editar usuarios.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la edición del usuario.
<b>Resultado esperado:</b> Editar usuarios.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 102. Prueba de aceptación 005

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 005
<b>Identificador de Historia:</b> H016
<b>Denominación:</b> Habilitar y deshabilitar usuarios
<b>Descripción:</b> El usuario del sistema podrá habilitar y deshabilitar los usuarios existentes.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara la confirmación para habilitar o deshabilitar el usuario.
<b>Resultado esperado:</b> Habilitar y deshabilitar usuarios.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 103. Prueba de aceptación 006

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 006
<b>Identificador de Historia:</b> H017
<b>Denominación:</b> Listar socios
<b>Descripción:</b> El usuario del sistema podrá listar los socios que se encuentren registrados en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una lista de socios registrados.
<b>Interfaz:</b> La aplicación mostrara el listado de socios en una tabla.
<b>Resultado esperado:</b> Listar socios del sistema.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 104. Prueba de aceptación 007

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 007
<b>Identificador de Historia:</b> H018
<b>Denominación:</b> Crear socios.
<b>Descripción:</b> El usuario del sistema podrá agregar nuevos socios al sistema.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá agregar nuevos socios.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la creación del nuevo socio.
<b>Resultado esperado:</b> Agregar socios al sistema.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 105. Prueba de aceptación 008

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 008
<b>Identificador de Historia:</b> H019
<b>Denominación:</b> Editar socios.
<b>Descripción:</b> El usuario del sistema podrá editar los socios del sistema.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá editar los socios.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la edición del socio.
<b>Resultado esperado:</b> Editar socios del sistema.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador



Tabla 106. Prueba de aceptación 009

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 009
<b>Identificador de Historia:</b> H020
<b>Denominación:</b> Habilitar y deshabilitar socios.
<b>Descripción:</b> El usuario del sistema podrá habilitar y deshabilitar los socios existentes.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá editar los socios.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara la confirmación para habilitar o deshabilitar el socio.
<b>Resultado esperado:</b> Habilitar y deshabilitar socios.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador  
 Tabla 107. Prueba de aceptación 010

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 010
<b>Identificador de Historia:</b> H021
<b>Denominación:</b> Listar lecturas.
<b>Descripción:</b> El usuario del sistema podrá listar las lecturas que se encuentren registrados en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una lista de lecturas registradas.
<b>Interfaz:</b> La aplicación mostrara un listado de lecturas en una tabla.
<b>Resultado esperado:</b> Listar lecturas.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 108. Prueba de aceptación 011

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 011
<b>Identificador de Historia:</b> H022
<b>Denominación:</b> Agregar lecturas.
<b>Descripción:</b> El usuario del sistema podrá agregar nuevas lecturas al sistema.
<b>Condición de ejecución:</b> Únicamente el usuario administrador y recaudador podrá agregar nuevas lecturas.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la creación de una nueva lectura.
<b>Resultado esperado:</b> Listar lecturas.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 109. Prueba de aceptación 012

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 012
<b>Identificador de Historia:</b> H023
<b>Denominación:</b> Editar lecturas.
<b>Descripción:</b> El usuario del sistema podrá editar las lecturas del sistema.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá editar las lecturas.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la edición de la lectura.
<b>Resultado esperado:</b> Listar lecturas.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 110. Prueba de aceptación 013

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 013
<b>Identificador de Historia:</b> H024
<b>Denominación:</b> Eliminar lecturas.
<b>Descripción:</b> El usuario del sistema podrá eliminar las lecturas del sistema.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá eliminar las lecturas.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara confirmar la eliminación de la lectura.
<b>Resultado esperado:</b> Eliminar lecturas.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 111. Prueba de aceptación 014

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 014
<b>Identificador de Historia:</b> H025
<b>Denominación:</b> Listar facturas.
<b>Descripción:</b> El usuario del sistema podrá listar las facturas que se encuentren registradas en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar un listado de facturas.
<b>Interfaz:</b> La aplicación mostrara un listado de facturas en una tabla.
<b>Resultado esperado:</b> Listar facturas.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 112. Prueba de aceptación 015

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 015
<b>Identificador de Historia:</b> H026
<b>Denominación:</b> Cobro de facturas.
<b>Descripción:</b> El usuario del sistema podrá realizar el cobro de las lecturas registradas en el sistema.
<b>Condición de ejecución:</b> Únicamente el usuario administrador y recaudador podrá realizar esta acción, las lecturas se agregarán a una cabecera de facturas.
<b>Interfaz:</b> La aplicación mostrara un listado de las lecturas correspondientes al socio indicado las cuales podrán ser cobradas.
<b>Resultado esperado:</b> Cobro de facturas.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 113. Prueba de aceptación 016

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 016
<b>Identificador de Historia:</b> H027
<b>Denominación:</b> Listar mortuoria.
<b>Descripción:</b> El usuario del sistema podrá listar el registro de0 mortuoria que se encuentren registrados en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una lista de la mortuoria registradas en el sistema.
<b>Interfaz:</b> La aplicación mostrara un listado de mortuoria registrado en el sistema.
<b>Resultado esperado:</b> Listar mortuoria.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 114. Prueba de aceptación 017

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 017
<b>Identificador de Historia:</b> H028
<b>Denominación:</b> Agregar mortuoria.
<b>Descripción:</b> El usuario del sistema podrá agregar nuevos registros de mortuoria.
<b>Condición de ejecución:</b> Únicamente el usuario administrador y recaudador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la creación del nuevo registro de mortuoria.
<b>Resultado esperado:</b> Agregar mortuoria.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 115. Prueba de aceptación 018

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 018
<b>Identificador de Historia:</b> H029
<b>Denominación:</b> Eliminar mortuoria.
<b>Descripción:</b> El usuario del sistema podrá eliminar los registros de mortuoria.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara confirmar la eliminación del registro de mortuoria.
<b>Resultado esperado:</b> Eliminar mortuoria.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 116. Prueba de aceptación 019

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 019
<b>Identificador de Historia:</b> H030
<b>Denominación:</b> Listar cuadro de caja.
<b>Descripción:</b> El usuario del sistema podrá listar los cuadros de caja que se encuentren registrados en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una lista de cuadros de cajas registrados.
<b>Interfaz:</b> La aplicación mostrara un listado de cuadros de caja registrado en el sistema.
<b>Resultado esperado:</b> Listar cuadro de caja.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 117. Prueba de aceptación 020

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 020
<b>Identificador de Historia:</b> H031
<b>Denominación:</b> Agregar cuadro de caja.
<b>Descripción:</b> El usuario del sistema podrá agregar nuevos registros de cuadro de caja.
<b>Condición de ejecución:</b> Únicamente el usuario administrador y recaudador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la creación del nuevo registro de cuadro de caja.
<b>Resultado esperado:</b> Agregar cuadro de caja.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 118. Prueba de aceptación 021

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 021
<b>Identificador de Historia:</b> H032
<b>Denominación:</b> Listar compra.
<b>Descripción:</b> El usuario del sistema podrá listar las compras que se encuentren registrados en el sistema.
<b>Condición de ejecución:</b> La aplicación deberá mostrar una lista de cuadros de caja registrados.
<b>Interfaz:</b> La aplicación mostrara un listado de compras registrado en el sistema.
<b>Resultado esperado:</b> Listar compras.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 119. Prueba de aceptación 022

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 022
<b>Identificador de Historia:</b> H033
<b>Denominación:</b> Agregar compra.
<b>Descripción:</b> El usuario del sistema podrá agregar nuevos registros de compras.
<b>Condición de ejecución:</b> Únicamente el usuario administrador y recaudador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la creación del nuevo registro de compra.
<b>Resultado esperado:</b> Agregar compra.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 120. Prueba de aceptación 023

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 023
<b>Identificador de Historia:</b> H034
<b>Denominación:</b> Editar compra.
<b>Descripción:</b> El usuario del sistema podrá editar las compras que estén registradas.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara los datos correspondientes para la edición de la compra.
<b>Resultado esperado:</b> Editar compra.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 121. Prueba de aceptación 024

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 024
<b>Identificador de Historia:</b> H035
<b>Denominación:</b> Eliminar compra.
<b>Descripción:</b> El usuario del sistema podrá eliminar los registros de compras.
<b>Condición de ejecución:</b> Únicamente el usuario administrador podrá realizar esta acción.
<b>Interfaz:</b> La aplicación mostrara una ventana modal en dónde solicitara confirmar la eliminación del registro de compra.
<b>Resultado esperado:</b> Eliminar compra.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador



Tabla 122. Prueba de aceptación 024

<b>PRUEBA DE ACEPTACIÓN</b>
<b>Número:</b> 025
<b>Identificador de Historia:</b> H036
<b>Denominación:</b> Generación de reportes.
<b>Descripción:</b> El usuario del sistema podrá generar reportes del sistema.
<b>Condición de ejecución:</b> El usuario podrá generar reportes con parámetros como fechas, socios, etc....
<b>Interfaz:</b> La aplicación mostrará una ventana modal en dónde se presentará su respectivo reporte y también permitirá el ingreso de parámetros.
<b>Resultado esperado:</b> Eliminar compra.
<b>Resultado de la prueba:</b> Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

## CAPÍTULO IV

### 4 Conclusiones y recomendaciones

#### 4.1 Conclusiones

El desarrollo de este proyecto permitió implementar una PWA que permite mantener un mejor control sobre los registros de la junta de agua potable Huambaló caserío La Merced, teniendo como conclusión lo siguiente:

- Al analizar los procesos de gestión de cobros de la junta de agua potable Huambaló caserío La Merced se pudo determinar, la forma de trabajo para realizar los cobros, con lo cual se pudo automatizar las tareas, así como también mejorar la seguridad de los registros que maneja la junta de agua potable.
- Se investigaron diversas tecnologías para el desarrollo de aplicaciones web progresivas de las cuales se seleccionó el framework Angular debido a que, al realizar el análisis comparativo, este cuenta con características superiores, las mismas que lo hacen un framework de desarrollo robusto, además es un framework compatible con el back-end desarrollado en Node.js.
- La implantación de la aplicación se cumplió de forma satisfactoria y fue bien recibida por los miembros de la junta de agua potable Huambaló caserío La Merced, esto puede ser evidenciado por medio de las pruebas de aceptación de las historias de usuario.

#### 4.2 Recomendaciones

- Se sugiere seguir evaluando la gestión de procesos y posteriormente ir adicionando nuevos módulos e incrementar las funcionalidades y gradualmente cubrir las necesidades que la organización vaya presentando con el paso del tiempo.
- Se recomienda realizar pruebas unitarias en el desarrollo del back-end dado que con los continuos cambios en el código se pueden producir errores.
- Adicionalmente, se recomienda el uso de una herramienta de control de versiones como GitHub con el fin de tener siempre el respaldo del código que se esté realizando.



## Bibliografía

- [1] E. V. Pico Gutiérrez and S. C. Núñez Neira, “El software contable como herramienta técnica en las microempresas de la provincia de Santa Elena, Ecuador,” *Kill. Soc.*, vol. 2, no. 1, pp. 21–26, 2018, doi: 10.26871/killkana\_social.v2i1.242.
- [2] E. A. Darko, S. Adarkwah, F. Donkor, and E. Kyei, “Management of Accounts Receivables in Utility Companies: A focus on Electricity Company of Ghana (E.C.G.),” *Int. J. Acad. Res. Bus. Soc. Sci.*, vol. 6, no. 11, pp. 486–518, 2016, doi: 10.6007/ijarbss/v6-i11/2422.
- [3] D. Acosta Chávez and G. Navarrete Navarrete, “Importancia del uso del software contable en pequeñas, medianas y grandes empresas del cantón Portoviejo,” *La Técnica Rev. las Agrociencias. ISSN 2477-8982*, no. 10, p. 62, 2013, doi: 10.33936/la\_tecnica.v0i10.572.
- [4] J. R. M. Quintanilla, “LA AUTOMATIZACIÓN DE LA INFORMACIÓN Y SU RELACIÓN CON LA GESTIÓN DE PROCESOS CONTABLES EN LA EMPRESA ‘CALZADO J7’ DE LA CIUDAD DE AMBATO, PROVINCIA DE TUNGURAHUA,,” 2016.
- [5] A. G. Vilcacundo Cordova, “La Innovación Tecnológica como Factor de Competitividad en las Pequeñas y Medianas Empresas Manufactureras del Cantón Ambato,” *Sist. Intel. para Domic. y Edif.*, p. 10, 2014, [Online]. Available: <http://repo.uta.edu.ec/bitstream/123456789/13279/1/FCHE-EBS-1519.pdf%0Ahttp://es.slideshare.net/Andysebas1/domotica-42887798>
- [6] J. Molina, M. Zea, M. J. Contenido, and F. García, “Comparación De Metodologías En Aplicaciones Web,” *3c Tecnol. glosas innovación Apl. a la pyme*, vol. 7, no. 1, pp. 1–19, 2018.
- [7] “Aplicaciones Web Progresivas: qué son, cómo funcionan y qué tienes que saber.” <https://www.muycomputerpro.com/2019/09/26/aplicaciones-web-progresivas-que-son-como-funcionan-y-que-tienes-que-saber> (accessed Oct. 06, 2021).
- [8] D. Malqui, “Diseño y desarrollo Web accesible utilizando HTML5 y CSS3 con nivel de conformidad A , respecto a las pautas WCAG,” *Rev. Científica sobre Accesibilidad Univers.*, pp. 41–104, 2015.
- [9] “¿Qué Es CSS?” <https://www.hostinger.es/tutoriales/que-es-css> (accessed Oct. 06, 2021).
- [10] “¿Qué es JavaScript y para qué sirve?” <https://soyrafamos.com/que-es-javascript-para-que-sirve/> (accessed Oct. 06, 2021).

- [11] V. Hernandez, “Usos Y Tipos De Aplicaciones Móviles.,” *Over Rim*, no. 1, pp. 191–199, 2017.
- [12] “¿Qué es el diseño responsivo? - Blog Trazada Marketing.” <https://trazada.com/que-es-el-diseno-responsivo/> (accessed Feb. 13, 2022).
- [13] “¿Por qué Vue.js está ganando tanta popularidad?” <https://adrianalonso.es/desarrollo-web/frontend/frameworks-javascript/por-que-vue-js-esta-ganando-tanta-popularidad/> (accessed Jul. 22, 2022).
- [14] “Vue.js - The Progressive JavaScript Framework | Vue.js.” <https://vuejs.org/> (accessed Mar. 13, 2022).
- [15] “Side Project: Desarrollo de aplicación web con React y Leaflet (I) - SIGdeletras :: Tecnologías de Información Geográfica.” <https://sigdeletras.com/2020/side-project-desarrollo-de-aplicacion-web-con-react-y-leaflet-i/> (accessed Jul. 22, 2022).
- [16] C. Gackenhaimer, “What Is React? Introduction to React,” pp. 1–20, 2015, doi: 10.1007/978-1-4842-1245-5\_1.
- [17] “Cómo actualizar Angular-CLI de forma rápida y eficiente - DEV Community.” <https://dev.to/arielmirra/como-actualizar-angular-cli-de-forma-rapida-y-eficiente-3kb9> (accessed Jul. 22, 2022).
- [18] M. Rivero, “AngularJs Paso a Paso,” *Leanpub*, vol. Segunda Ed, no. 1, p. 196, 2016.
- [19] P. Rospigliosi, “Estudio sobre Angular 2 y superior,” p. 51, 2020, [Online]. Available: [https://ebuah.uah.es/xmlui/bitstream/handle/10017/44794/TFM\\_Rospigliosi\\_Beltran\\_2020.pdf?sequence=1&isAllowed=y](https://ebuah.uah.es/xmlui/bitstream/handle/10017/44794/TFM_Rospigliosi_Beltran_2020.pdf?sequence=1&isAllowed=y)
- [20] M. M. E. Navarro *et al.*, “Selección de Metodologías Ágiles e Integración de Arquitecturas de Software en el Desarrollo de Sistemas de Información,” pp. 632–636, 2016.
- [21] O. Iv, *La evaluación de desempeño por competencias , percepción de justicia y satisfacción por parte de los colaboradores de una organización “ Gestión del Talento Humano : Enfoques y Modelos ,”* no. May. 2021.
- [22] C. Rodríguez and R. Dorado, “¿ Por qué implementar Scrum ? Why to implement Scrum ? Pour quelles raisons mettre en place Scrum ? Porque implementar Scrum ?,” *Rev. Ontare*, vol. 3, no. 1, pp. 125–144, 2015, [Online]. Available: <http://200.0.187.30/index.php/Revistao/article/view/1253/1218>
- [23] “Islavisual.com - Diferencias Entre Scrum Y Xp de Desarrollo Web.” [https://islavisual.com/articulos/desarrollo\\_web/diferencias-entre-scrum-y-xp.php](https://islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php)

(accessed Jul. 22, 2022).

- [24] M. Calero Solis, “Una explicación de la programación extrema (XP),” *V Encuentro usuarios xBase*, vol. 1, p. 11, 2003, [Online]. Available: <ftp://200.0.25.51/Documentacion/ExplicaXP.pdf>
- [25] J. Ferrer, “Programación eXtrema y Software Libre Tabla de contenidos,” pp. 1–23, 2002, [Online]. Available: <http://es.tldp.org/Presentaciones/200211hispalinux/ferrer/robles-ferrer-ponencia-hispalinux-2002.pdf>
- [26] “Conoce las 3 metodologías ágiles más usadas | OpenWebinars.” <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/> (accessed Jul. 22, 2022).
- [27] “Instalación de Node.js en Ubuntu.” <https://devcode.la/tutoriales/instalacion-de-nodejs-en-ubuntu/> (accessed Jul. 22, 2022).
- [28] “Acerca | Node.js.” <https://nodejs.org/es/about/> (accessed Mar. 03, 2022).
- [29] “¿Qué es Node.js, y para qué sirve?” <https://www.itdo.com/blog/que-es-node-js-y-para-que-sirve/> (accessed Mar. 03, 2022).
- [30] “👉 Querys y comandos útiles de MySQL Server - Tutorial.” <https://espaciotecnologico.co/comandos-utiles-de-mysql-server/> (accessed Jul. 22, 2022).
- [31] P. López Herrera, “Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL.,” *Univ. Autónoma del Estado México*, p. 72, 2016, [Online]. Available: <http://ri.uaemex.mx/bitstream/handle/20.500.11799/62548/TesisPatriciaLopezHerrera.pdf?sequence=3>
- [32] L. A. Casillas Santillán, M. G. Ginestà, and Ó. Perez Mora, “Bases de datos en MySQL,” *Fuoc*, vol. 1, pp. 1–238, 2016, [Online]. Available: [https://d1wqtxts1xzle7.cloudfront.net/54167148/MYSQL\\_ManualL.pdf?1502991767=&response-content-disposition=inline%3B+filename%3DBases\\_de\\_datos\\_en\\_MySQL.pdf&Expires=1622769346&Signature=Gd6hKxHrsk0Ah-FpuFKekr1iK69VOe2YaJz67SnSyxPQHfPPTHW5ZZK~YftoNSO-NvoEKz7](https://d1wqtxts1xzle7.cloudfront.net/54167148/MYSQL_ManualL.pdf?1502991767=&response-content-disposition=inline%3B+filename%3DBases_de_datos_en_MySQL.pdf&Expires=1622769346&Signature=Gd6hKxHrsk0Ah-FpuFKekr1iK69VOe2YaJz67SnSyxPQHfPPTHW5ZZK~YftoNSO-NvoEKz7)
- [33] “Visual Studio Code ya está disponible en su versión online para todos.” <https://www.mundodeportivo.com/urbantecno/windows/microsoft-visual-studio-code-online> (accessed Jul. 22, 2022).

- [34] “Documentation for Visual Studio Code.” <https://code.visualstudio.com/docs> (accessed Mar. 03, 2022).
- [35] “Qué es GitHub y por qué es útil al aprender programación | HACK A BOSS.” <https://www.hackaboss.com/blog/que-es-github-y-por-que-es-util-al-aprender-programacion> (accessed Jul. 22, 2022).
- [36] “Qué es Github y qué es lo que le ofrece a los desarrolladores.” <https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores> (accessed Mar. 03, 2022).