



**UNIVERSIDAD TÉCNICA DE AMBATO**

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

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

**Tema:**

---

**APLICACIÓN WEB PROGRESIVA (PWA) PARA LA AUTOMATIZACIÓN  
DE LOS PROCESOS DE GESTIÓN E INFORMACIÓN EN LIGA  
DEPORTIVA PARROQUIAL TOTORAS”**

---

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

**ÁREA: Software**

**LÍNEA DE INVESTIGACIÓN: Desarrollo de Software**

**AUTOR: Santiago Damián Yaguargos Castro**

**TUTOR: Ing. Edison Homero Álvarez Mayorga**

**Ambato - Ecuador**

**Agosto - 2020**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: “APLICACIÓN WEB PROGRESIVA (PWA) PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE GESTIÓN E INFORMACIÓN EN LIGA DEPORTIVA PARROQUIAL TOTORAS.”, desarrollado bajo la modalidad Proyecto de Investigación por el señor Santiago Damián Yaguargos Castro, 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, agosto 2020.



Firmado electrónicamente por:  
**EDISON HOMERO  
ALVAREZ MAYORGA**

---

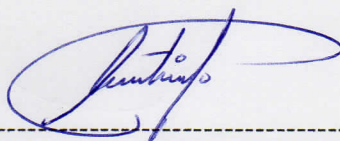
Ing. Edison Álvarez, Mg.

**TUTOR**

## AUTORÍA

El presente Proyecto de Investigación titulado: “APLICACIÓN WEB PROGRESIVA (PWA) PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE GESTIÓN E INFORMACIÓN EN LIGA DEPORTIVA PARROQUIAL TOTORAS.” es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2020.



-----  
Santiago Damián Yaguargos Castro

C.I.: 180434800-9

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Santiago Damián Yaguargos Castro, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado “APLICACIÓN WEB PROGRESIVA (PWA) PARA LA AUTOMATIZACIÓN DE LOS PROCESOS DE GESTIÓN E INFORMACIÓN EN LIGA DEPORTIVA PARROQUIAL TOTORAS.”, 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, agosto 2020.



Firmado electrónicamente por:  
**ELSA PILAR  
URRUTIA**

-----  
Ing. Pilar Urrutia, Mg.  
PRESIDENTA DEL TRIBUNAL

-----  
Ing. Hernán Naranjo, Mg.  
PROFESOR CALIFICADOR

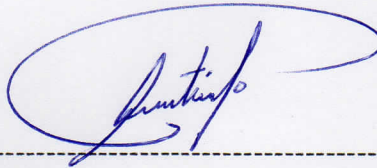
-----  
Ing. Carlos Núñez, Mg.  
PROFESOR CALIFICADOR

## 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, agosto 2020.



-----  
Santiago Damián Yaguargos Castro

C.I.: 180434800-9

AUTOR

## **DEDICATORIA**

*A Dios, con amor infinito, por darme la existencia y permitirme cumplir mis metas con su bendición. A su hijo Jesús, mi mejor amigo.*

*A mis padres Juan e Inés a quienes más amo en la vida, por ser quienes me dieron la vida y con su amor, esfuerzo y comprensión día a día, me han forjado como una persona de bien.*

*A mi hermano Lenin, su esposa Verónica y mi sobrino Juan Fernando, a quienes admiro por brindarme su confianza, enseñándome que con humildad, siempre se puede superar cualquier obtáculo en la vida.*

***Santiago Damián Yaguargos Castro***

## **AGRADECIMIENTO**

*A Dios por todas sus bendiciones derramadas durante mi carrera, especialmente en los momentos más difíciles.*

*A mis padres Juan e Inés, por su amor, su esfuerzo, sus consejos, sus enseñanzas que me motivaron a luchar por mis sueños, a mi hermano Lenin, su esposa Verónica y a mi sobrino Juan Fernando, por brindarme la confianza y el apoyo desde el calor de hogar.*

*A docentes y administrativos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial que me impartieron todo su conocimiento, especialmente a los Ingenieros Edison Alvarez, Clay Aldás, Hernán Naranjo y Carlos Nuñez, quienes me motivaron a ser un excelente profesional en la etapa final de mi carrera.*

*A mis compañeros y amigos que estuvieron en los momentos buenos y malos durante mi carrera universitaria. A Dani, con quien comparto mis sentimientos, por su apoyo y motivación durante la etapa final del desarrollo de mi investigación.*

**Santiago Damián Yaguargos Castro**

## ÍNDICE GENERAL DE CONTENIDOS

|  |     |
|--|-----|
| PORTADA.....   | i   |
| APROBACIÓN DEL TUTOR.....                                  | ii  |
| AUTORÍA.....   | iii |
| APROBACIÓN TRIBUNAL DE GRADO.....                          | iv  |
| DERECHOS DE AUTOR.....                                     | v   |
| DEDICATORIA.....   | vi  |
| AGRADECIMIENTO.....  | vii |
| RESUMEN EJECUTIVO.....                                     | xv  |
| ABSTRACT (SUMMARY).....                                    | xvi |
| CAPÍTULO I.....  | 1   |
| MARCO TEÓRICO.....   | 1   |
| 1.1    Antecedentes investigativos.....                    | 1   |
| 1.2    Objetivos.....                                      | 2   |
| 1.2.1    General.....                                      | 2   |
| 1.2.2    Específicos.....                                  | 2   |
| 1.3    Fundamentación teórica.....                         | 3   |
| 1.3.1    Liga Deportiva Parroquial.....                    | 3   |
| 1.3.2    Automatización de procesos.....                   | 3   |
| 1.3.3    Gestión (informática).....                        | 3   |
| 1.3.4    Información (informática).....                    | 4   |
| 1.3.5    JavaScript.....                                   | 4   |
| 1.3.6    NodeJS.....                                       | 4   |
| 1.3.7    Framework.....                                    | 5   |
| 1.3.8    VueJS.....  | 5   |
| 1.3.8.1    Vuex.....                                       | 7   |
| 1.3.8.2    VueJS CLI.....                                  | 9   |
| 1.3.8.3    Vuetify.....                                    | 10  |
| 1.3.8.4    Vuelidate.....                                  | 11  |
| 1.3.9    Progressive Web App (PWA).....                    | 12  |
| •    Características de una Aplicación Web Progresiva..... | 13  |
| •    Ventajas de una Aplicación Web Progresiva.....        | 13  |
| •    Service Worker.....                                   | 14  |



|                                     |  |           |
|-------------------------------------|--|-----------|
| •                                   | <b>Manifiesto .....</b>                                      | <b>16</b> |
| •                                   | <b>Estrategias de caché en el Service Worker .....</b>       | <b>18</b> |
| 1.3.10                              | <b>Base de Datos No SQL .....</b>                            | <b>19</b> |
| 1.3.11                              | <b>Firestore .....</b>                                       | <b>20</b> |
| •                                   | <b>Firestore Realtime Database .....</b>                     | <b>20</b> |
| •                                   | <b>Firestore Authentication .....</b>                        | <b>21</b> |
| •                                   | <b>Firestore Cloud Storage .....</b>                         | <b>21</b> |
| •                                   | <b>Firestore Hosting.....</b>                                | <b>21</b> |
| •                                   | <b>Firestore Cloud Messaging.....</b>                        | <b>22</b> |
| 1.3.12                              | <b>PostgreSQL.....</b>                                       | <b>22</b> |
| 1.3.13                              | <b>Spring Framework.....</b>                                 | <b>22</b> |
| 1.3.13.1                            | <b>Spring Boot .....</b>                                     | <b>23</b> |
| 1.3.14                              | <b>Metodología ágil .....</b>                                | <b>23</b> |
| 1.3.15                              | <b>Pruebas de caja negra.....</b>                            | <b>23</b> |
| <b>CAPÍTULO II .....</b>            |  | <b>25</b> |
| <b>METODOLOGÍA .....</b>            |  | <b>25</b> |
| 2.1.                                | <b>Materiales.....</b>                                       | <b>25</b> |
| 2.2.                                | <b>Métodos .....</b>   | <b>25</b> |
| 2.2.1.                              | <b>Modalidad de la investigación.....</b>                    | <b>25</b> |
| 2.2.2.                              | <b>Población.....</b>  | <b>26</b> |
| 2.2.3.                              | <b>Recolección de Información .....</b>                      | <b>27</b> |
| 2.2.4.                              | <b>Procesamiento y análisis de datos.....</b>                | <b>27</b> |
| 2.2.5.                              | <b>Metodología de Desarrollo .....</b>                       | <b>28</b> |
| 2.2.6.                              | <b>Desarrollo del Proyecto.....</b>                          | <b>29</b> |
| <b>CAPÍTULO III.....</b>            |  | <b>31</b> |
| <b>RESULTADOS Y DISCUSIÓN .....</b> |  | <b>31</b> |
| 3.1.                                | <b>Análisis y discusión de resultados.....</b>               | <b>31</b> |
| 3.1.1.                              | <b>Recolección de la información.....</b>                    | <b>31</b> |
| 3.1.2.                              | <b>Análisis de la información .....</b>                      | <b>31</b> |
| 3.2.                                | <b>Desarrollo de la metodología.....</b>                     | <b>35</b> |
| 3.2.1.                              | <b>Análisis de las herramientas para el desarrollo .....</b> | <b>35</b> |
| •                                   | <b>Base de Datos .....</b>                                   | <b>35</b> |
| •                                   | <b>Framework de desarrollo Frontend .....</b>                | <b>36</b> |
| •                                   | <b>Framework de desarrollo Backend .....</b>                 | <b>38</b> |
| 3.2.2.                              | <b>Metodología ágil .....</b>                                | <b>38</b> |

|   |            |
|---|------------|
| • Fase 1. Exploración .....   | 38         |
| • Fase 2. Planificación del Proyecto.....   | 40         |
| • Fase 3. Iteraciones .....   | 62         |
| • Fase 4. Producción .....  | 64         |
| • Fase 5. Pruebas .....   | 136        |
| <b>CAPÍTULO IV .....</b>  | <b>158</b> |
| <b>CONCLUSIONES Y RECOMENDACIONES .....</b>   | <b>158</b> |
| <b>1.2.1. Conclusiones .....</b>  | <b>158</b> |
| <b>1.2.2. Recomendaciones .....</b>   | <b>159</b> |
| <b>BIBLIOGRAFÍA.....</b>  | <b>160</b> |
| <b>ANEXOS .....</b>   | <b>162</b> |
| <b>Anexo 1. Encuesta para determinar requerimientos y la factibilidad para el desarrollo una PWA para Liga Deportiva Parroquial Totoras. ....</b> | <b>162</b> |
| <b>Anexo 2. Manual de instalación de la PWA .....</b>   | <b>164</b> |
| <b>Anexo 3. Manual de uso del sistema PWA de Liga Deportiva Parroquial Totoras Versión 1.0.2. para el Administrador. ....</b>                     | <b>169</b> |

## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| Figura 1. Logo NodeJS .....  | 5  |
| Figura 2. Logo VueJs .....   | 6  |
| Figura 3. Flujo de datos unidireccional de Vue .....                           | 7  |
| Figura 4. Flujo de datos en una aplicación con Vuex .....                      | 8  |
| Figura 5. Interfaz Grafica (UI) Administrador de proyectos de Vue .....        | 9  |
| Figura 6. Logo de Vuetify .....  | 10 |
| Figura 7. Tabla de comparaciones entre librerías de diseño para Vue.....       | 11 |
| Figura 8. Logo de Vuelidate .....  | 11 |
| Figura 9. Instalación de un Service Worker.....                                | 14 |
| Figura 10. Proceso de inicialización de un Service Worker .....                | 15 |
| Figura 11. Service Worker instalado en una aplicación web .....                | 16 |
| Figura 12. Elementos del manifiesto.....                                       | 17 |
| Figura 13. Logo de Firebase .....  | 20 |
| Figura 14. Logo de PostgreSQL .....  | 22 |
| Figura 15. Logo de SpringBoot Java.....  | 23 |
| Figura 16. Estructura de la prueba de caja negra .....                         | 24 |
| Figura 17. Resultado de Encuesta - Pregunta 1.....                             | 31 |
| Figura 18. Resultado de Encuesta - Pregunta 2.....                             | 32 |
| Figura 19. Resultado de Encuesta - Pregunta.....                               | 33 |
| Figura 20. Resultado de Encuesta - Pregunta 4.....                             | 33 |
| Figura 21. Resultado de Encuesta - Pregunta 5.....                             | 34 |
| Figura 22. Resultado de Encuesta - Pregunta 6.....                             | 35 |
| Figura 23. Modelo Base de Datos relacional .....                               | 64 |
| Figura 24. Modelo Base de Datos no relacional .....                            | 65 |
| Figura 25. GUI de Vue.....   | 69 |
| Figura 26. Creación de un proyecto de Vue.....                                 | 69 |
| Figura 27. Presets en la creación de un proyecto en Vue.....                   | 70 |
| Figura 28. Panel del proyecto creado en Vue .....                              | 71 |
| Figura 29. Tareas del proyecto creado en Vue.....                              | 71 |
| Figura 30. Pantalla de inicio de la aplicación creada en Vue.....              | 72 |
| Figura 31. Estructura del proyecto con Vue.....                                | 73 |
| Figura 32. Complementos del proyecto en Vue.....                               | 74 |
| Figura 33. Vuetify Framework Plugin para Vue.....                              | 74 |
| Figura 34. Instalación de Vuetify Framework Plugin para Vue .....              | 75 |
| Figura 35. Pantalla de inicio de la aplicación con Vuetify .....               | 75 |
| Figura 36. Página inicial de Firebase .....                                    | 76 |
| Figura 37. Crear un proyecto con Firebase .....                                | 76 |
| Figura 38. Nombre del proyecto con Firebase .....                              | 77 |
| Figura 39. Google Analytics en el proyecto a crear de Firebase .....           | 77 |
| Figura 40. Mensaje exitoso del Proyecto de Firebase creado .....               | 78 |
| Figura 41. Ir a Configuración del Proyecto.....                                | 78 |
| Figura 42. Configuración del Proyecto de Firebase Liga Totoras .....           | 79 |
| Figura 43. Página principal del Proyecto de Firebase Liga Totoras .....        | 80 |
| Figura 44. Configuración de Firebase para la aplicación web Liga Totoras ..... | 80 |
| Figura 45. Página de inicio pagina web LigaPro .....                           | 81 |
| Figura 46. Archivos views o vistas en Vue.....                                 | 82 |

|   |     |
|---|-----|
| Figura 47. API de Whatsapp empresarial de LDPT.....   | 87  |
| Figura 48. Firebase Storage en la consola de Firebase .....                                   | 92  |
| Figura 49. Archivo almacenado en Firebase Storage.....  | 93  |
| Figura 50. Firebase Authentication en la consola de Firebase.....                             | 94  |
| Figura 51. Proveedores de Firebase Authentication .....                                       | 95  |
| Figura 52. Habilitar proveedor Correo electrónico/contraseña en Firebase Authentication ..... | 95  |
| Figura 53. Añadir usuario Correo electrónico/contraseña en Firebase Authentication .....      | 96  |
| Figura 54. Lista de usuarios Correo electrónico/contraseña en Firebase Authentication .....   | 96  |
| Figura 55. Estructura de los archivos del Proyecto en Spring .....                            | 105 |
| Figura 56. Directorio público “dist” compilado.....   | 128 |
| Figura 57. Web Server for Chrome.....   | 129 |
| Figura 58. Manifest desde Google Chrome .....   | 132 |
| Figura 59. Manifest desde Google Chrome .....   | 133 |
| Figura 60. Cloud Messaging en la consola de Firebase .....                                    | 135 |

## ÍNDICE DE TABLAS

|   |    |
|---|----|
| <i>Tabla 1. Población</i> .....                                       | 26 |
| <i>Tabla 2. Recolección de la información</i> .....                   | 27 |
| <i>Tabla 3. Cuadro comparativo de metodologías ágiles</i> .....       | 29 |
| <i>Tabla 4. Modelo historias de usuario</i> .....                     | 40 |
| <i>Tabla 5. Diseño de interfaces - PWA</i> .....                      | 41 |
| <i>Tabla 6. Visualizar información de LDPT</i> .....                  | 41 |
| <i>Tabla 7. Visualizar información de Clubes</i> .....                | 42 |
| <i>Tabla 8. Descargar documentación</i> .....                         | 42 |
| <i>Tabla 9. Inicio de Sesión</i> .....                                | 43 |
| <i>Tabla 10. Ingresar noticias</i> .....                              | 43 |
| <i>Tabla 11. Actualizar noticias</i> .....                            | 44 |
| <i>Tabla 12. Eliminar noticias</i> .....                              | 44 |
| <i>Tabla 13. Visualizar Noticias</i> .....                            | 45 |
| <i>Tabla 14. Ingresar Clubes</i> .....                                | 45 |
| <i>Tabla 15. Actualizar Clubes</i> .....                              | 46 |
| <i>Tabla 16. Eliminar Clubes</i> .....                                | 46 |
| <i>Tabla 17. Visualizar Clubes</i> .....                              | 47 |
| <i>Tabla 18. Buscar Clubes</i> .....                                  | 47 |
| <i>Tabla 19. Ingresar torneos</i> .....                               | 48 |
| <i>Tabla 20. Actualizar torneos</i> .....                             | 48 |
| <i>Tabla 21. Eliminar torneos</i> .....                               | 49 |
| <i>Tabla 22. Visualizar torneos</i> .....                             | 49 |
| <i>Tabla 23. Buscar torneos</i> .....                                 | 50 |
| <i>Tabla 24. Ingresar Categorías</i> .....                            | 50 |
| <i>Tabla 25. Actualizar Categorías</i> .....                          | 51 |
| <i>Tabla 26. Eliminar Categorías</i> .....                            | 51 |
| <i>Tabla 27. Visualizar Categorías</i> .....                          | 52 |
| <i>Tabla 28. Ingresar Jugadores</i> .....                             | 52 |
| <i>Tabla 29. Actualizar jugadores</i> .....                           | 53 |
| <i>Tabla 30. Eliminar jugadores</i> .....                             | 53 |
| <i>Tabla 31. Visualizar jugadores</i> .....                           | 54 |
| <i>Tabla 32. Buscar jugadores</i> .....                               | 54 |
| <i>Tabla 33. Ingresar equipos</i> .....                               | 55 |
| <i>Tabla 34. Actualizar equipos</i> .....                             | 55 |
| <i>Tabla 35. Eliminar equipos</i> .....                               | 56 |
| <i>Tabla 36. Visualizar equipos</i> .....                             | 56 |
| <i>Tabla 37. Buscar equipos</i> .....                                 | 57 |
| <i>Tabla 38. Habilitar equipos</i> .....                              | 57 |
| <i>Tabla 39. Inhabilitar equipos</i> .....                            | 58 |
| <i>Tabla 40. Visualizar equipos habilitados</i> .....                 | 58 |
| <i>Tabla 41. Habilitar jugadores</i> .....                            | 59 |
| <i>Tabla 42. Inhabilitar jugadores</i> .....                          | 59 |
| <i>Tabla 43. Visualizar jugadores habilitados</i> .....               | 60 |
| <i>Tabla 44. Visualizar jugadores habilitados</i> .....               | 60 |
| <i>Tabla 45. Estimación de historias de usuario</i> .....             | 62 |
| <i>Tabla 46. Plan de Entrega</i> .....                                | 62 |
| <i>Tabla 47. Histórico de versiones en historias de usuario</i> ..... | 63 |

|   |            |
|---|------------|
| <i>Tabla 48. Equipos y roles .....</i>  | <i>63</i>  |
| <i>Tabla 49. Estructura tabla league .....</i>                                | <i>65</i>  |
| <i>Tabla 50. Estructura tabla club .....</i>                                  | <i>66</i>  |
| <i>Tabla 51. Estructura tabla category .....</i>                              | <i>66</i>  |
| <i>Tabla 52. Estructura tabla team .....</i>                                  | <i>66</i>  |
| <i>Tabla 53. Estructura tabla player .....</i>                                | <i>67</i>  |
| <i>Tabla 54. Estructura tabla championship.....</i>                           | <i>67</i>  |
| <i>Tabla 54. Estructura tabla teamChampionship .....</i>                      | <i>67</i>  |
| <i>Tabla 56. Estructura tabla qualification.....</i>                          | <i>67</i>  |
| <i>Tabla 57. Estructura tabla news .....</i>                                  | <i>67</i>  |
| <i>Tabla 58. Prueba de aceptación - Información de LDPT.....</i>              | <i>137</i> |
| <i>Tabla 59. Prueba de aceptación - Información de Clubes Filiales .....</i>  | <i>138</i> |
| <i>Tabla 60. Prueba de aceptación - Descargar documentación de LDPT.....</i>  | <i>138</i> |
| <i>Tabla 61. Prueba de aceptación - Inicio de Sesión .....</i>                | <i>139</i> |
| <i>Tabla 62. Prueba de aceptación - Ingresar Noticias .....</i>               | <i>139</i> |
| <i>Tabla 63. Prueba de aceptación - Actualizar Noticias .....</i>             | <i>140</i> |
| <i>Tabla 64. Prueba de aceptación - Eliminar Noticias .....</i>               | <i>140</i> |
| <i>Tabla 65. Prueba de aceptación - Visualizar Noticias .....</i>             | <i>141</i> |
| <i>Tabla 66. Prueba de aceptación - Ingresar Clubes .....</i>                 | <i>142</i> |
| <i>Tabla 67. Prueba de aceptación - Actualizar Clubes .....</i>               | <i>142</i> |
| <i>Tabla 68. Prueba de aceptación - Eliminar Clubes.....</i>                  | <i>143</i> |
| <i>Tabla 69. Prueba de aceptación - Visualizar Clubes .....</i>               | <i>143</i> |
| <i>Tabla 70. Prueba de aceptación - Buscar Clubes.....</i>                    | <i>144</i> |
| <i>Tabla 71. Prueba de aceptación - Ingresar torneos .....</i>                | <i>144</i> |
| <i>Tabla 72. Prueba de aceptación - Actualizar torneos .....</i>              | <i>145</i> |
| <i>Tabla 73. Prueba de aceptación - Eliminar torneos.....</i>                 | <i>145</i> |
| <i>Tabla 74. Prueba de aceptación - Visualizar torneos.....</i>               | <i>146</i> |
| <i>Tabla 75. Prueba de aceptación - Buscar torneos.....</i>                   | <i>146</i> |
| <i>Tabla 76. Prueba de aceptación - Ingresar categorías .....</i>             | <i>147</i> |
| <i>Tabla 77. Prueba de aceptación - Actualizar categorías .....</i>           | <i>148</i> |
| <i>Tabla 78. Prueba de aceptación - Eliminar categorías .....</i>             | <i>148</i> |
| <i>Tabla 79. Prueba de aceptación - Visualizar categorías.....</i>            | <i>149</i> |
| <i>Tabla 80. Prueba de aceptación - Ingresar jugadores .....</i>              | <i>149</i> |
| <i>Tabla 81. Prueba de aceptación - Actualizar jugadores .....</i>            | <i>150</i> |
| <i>Tabla 82. Prueba de aceptación - Eliminar jugadores.....</i>               | <i>150</i> |
| <i>Tabla 83. Prueba de aceptación - Visualizar jugadores.....</i>             | <i>151</i> |
| <i>Tabla 84. Prueba de aceptación - Buscar jugadores.....</i>                 | <i>151</i> |
| <i>Tabla 85. Prueba de aceptación - Ingresar equipos.....</i>                 | <i>152</i> |
| <i>Tabla 86. Prueba de aceptación - Actualizar equipos .....</i>              | <i>152</i> |
| <i>Tabla 87. Prueba de aceptación - Eliminar equipos .....</i>                | <i>153</i> |
| <i>Tabla 88. Prueba de aceptación - Visualizar equipos .....</i>              | <i>154</i> |
| <i>Tabla 89. Prueba de aceptación - Buscar jugadores.....</i>                 | <i>154</i> |
| <i>Tabla 90. Prueba de aceptación - Habilitar equipos.....</i>                | <i>155</i> |
| <i>Tabla 91. Prueba de aceptación - Inhabilitar equipos .....</i>             | <i>155</i> |
| <i>Tabla 92. Prueba de aceptación - Visualizar equipos habilitados .....</i>  | <i>156</i> |
| <i>Tabla 93. Prueba de aceptación - Habilitar jugadores .....</i>             | <i>156</i> |
| <i>Tabla 94. Prueba de aceptación - Inhabilitar jugadores.....</i>            | <i>157</i> |
| <i>Tabla 95. Prueba de aceptación - Visualizar jugadores habilitados.....</i> | <i>157</i> |

## RESUMEN EJECUTIVO

En el presente proyecto de investigación, se ha implementado una Aplicación Web Progresiva para la automatización de los principales procesos de gestión e información en Liga Deportiva Parroquial Totoras.

Para el desarrollo de la investigación se realizó un levantamiento de requerimientos en la institución, basada en la información tabulada, luego de aplicar encuestas a la población directamente involucrada. También se realizó un estudio teórico a las tecnologías aplicables para el desarrollo de software.

Con un mayor enfoque en el Frontend: el uso del framework VueJs con un desglose de: definiciones, características y ventajas; la librería Vuetify para diseño y maquetación web; la librería Vuelidate, enfocada en validaciones para los formularios existentes en la aplicación. En el Backend, el framework Spring Boot de Java para el desarrollo de servicios web. En la construcción de la PWA, se estudia el Service Worker, las estrategias de caché y el archivo de manifiesto.

Se utiliza de la plataforma Firebase como una Base de Datos No Relacional en los procesos de información y PostgreSQL como una Base de Datos Relacional en los procesos de gestión. Se aplicó la metodología ágil XP (Xtreme Programming) para el desarrollo de software.

**Descriptores:** Aplicación Web Progresiva, Service Worker, VueJs, Firebase.

## **ABSTRACT (SUMMARY)**

In this research project, a Progressive Web App has been implemented for the automation of the main management and information processes in Liga Deportiva Parroquial Totoras.

For the development of the research, a survey of requirements was carried out in the institution, based on the tabulated information, after applying surveys to the population directly involved. A theoretical study of applicable technologies for software development was also carried out.

With a greater focus on the Frontend: the use of the VueJs framework with a breakdown of: definitions, characteristics and advantages; the Vuetify library for web design and layout; the Vuelidate library, focused on validations for the existing forms in the application. In the Backend, the Spring Boot Java framework for developing web services. In building the PWA, the Service Worker, the cache strategies, and the manifest file are studied.

It is used from the Firebase platform as a Non-Relational Database in the information processes and PostgreSQL as a Relational Database in the management processes. The agile methodology XP (Xtreme Programming) was applied for software development.

**Descriptors:** Progressive Web App, Service Worker, VueJs, Firebase



# **CAPÍTULO I**

## **MARCO TEÓRICO**

### **1.1 Antecedentes investigativos**

Para la presente investigación se realizó la revisión y análisis del material bibliográfico existente en los diferentes repositorios de las Universidades del Ecuador, que se relacionan con el tema de investigación y posteriormente se han tomado los siguientes aportes investigativos, detallando lo más relevante:

Byron Bladimir Cárdenas, en su proyecto de investigación desarrollado en 2012 denominado “Sistema automatizado para el registro y sanciones de los jugadores en la Liga Deportiva Parroquial Huachi Grande”, se enfoca en la necesidad de mantener un mayor control de los datos puesto que se manejaban en archivos planos generando problemas económicos y mala organización. [1]

Byron Moreno Moreno, en su proyecto de investigación desarrollado en 2015 denominado “Sistema automatizado para el control de la fiabilidad del control de la información en la gestión de los campeonatos de futbol en Liga Deportiva Cantonal Pelileo, concluye que implementar este tipo de aplicaciones permite optimizar y agilizar procesos en organización, administración y control de los campeonatos de futbol en mencionada liga ya que llevarlo de manera manual, genera un ineficiente procesamiento de la información sin adecuado y eficiente control. [2]

En el uso de las tecnologías PWA's se ha optado buscar en otros repositorios ya que en la Universidad Técnica de Ambato no se han desarrollado temas referentes a esta tecnología.

Ney Miranda Tigse y Génesis Torres Terranova en el proyecto de investigación desarrollado en 2018 denominado “Implementación de una Aplicación Web Progresiva (PWA), para la identificación y notificación de mascotas extraviadas

usando placas con código de respuesta rápida (qr) en collares.”, concluye que actualmente la aplicación web progresiva se acopla a todo tipo de dispositivos como tablets, smartphones, sin importar el sistema operativo, adicionalmente no necesita ir a tienda online del móvil para descargar el ejecutable, ya que se accede al sistema por medio del navegador y se guardará el icono de la aplicación en el escritorio como acceso directo al aplicativo. [3]

## **1.2 Objetivos**

### **1.2.1 General**

Implementar una Aplicación Web Progresiva (PWA) para la automatización de los procesos de gestión e información en Liga Deportiva Parroquial Totoras.

### **1.2.2 Específicos**

- Analizar los requerimientos del sistema mediante el estudio de determinados procesos informativos que realiza Liga Deportiva Parroquial Totoras previo y durante el desarrollo de un campeonato.
- Investigar sobre los fundamentos teóricos para el uso de herramientas en la construcción de una Aplicación Web Progresiva.
- Construcción de la PWA para Liga Deportiva Parroquial Totoras.

Cada objetivo planteado se ha cumplido eficazmente durante el desarrollo del proyecto de investigación, con el análisis y recolección de información mediante una encuesta dirigida a los involucrados con la institución Liga Deportiva Parroquial Totoras, obteniendo así, información relevante para establecer los requerimientos del sistema.

Con la experiencia y las ideas empresariales e institucionales del presidente de Liga Deportiva Parroquial Totoras, se ha planteado implementar este tipo de aplicación basada en una tecnología moderna. Se trata de una aplicación web que facilita el alcance de la información desde distintos dispositivos, debido al comportamiento idéntico a una aplicación nativa; para ello se ha investigado diferentes casos de éxito con complementos y herramientas óptimas para su desarrollo.

El estudio de las PWA's en el desarrollo de software permite a la institución crecer a nivel deportivo, ya que es pionera en el uso de esta tecnología, misma que ha sido desarrollada bajo un intenso proceso de investigación para su factibilidad.

Finalmente, se ha implementado la PWA de Liga Deportiva Parroquial Totoras luego de realizar diferentes pruebas en varios dispositivos de quienes deciden aventurarse en el uso de este aplicativo, facilitando así la automatización de los procesos de gestión e información.

### **1.3 Fundamentación teórica**

#### **1.3.1 Liga Deportiva Parroquial**

En el Art. 58 del Reglamento General a la Ley del Deporte, Educación Física y Recreación en la Sección 2 del Deporte Barrial, acerca de la constitución de las Ligas Deportivas Barriales/Parroquiales, resume que una Liga Deportiva Barrial y/o Parroquial es una institución que tiene al menos tres clubes deportivos básicos barriales y/o parroquiales con personería jurídica y/o reconocimiento deportivo y participación deportiva recreativa. [4]

#### **1.3.2 Automatización de procesos**

La informática es una ciencia que ha contribuido esencialmente en la automatización de procesos, con el principal objetivo de incrementar la eficiencia de una empresa con relación a la necesidad del cliente. En un proceso de automatización, el computador es la herramienta fundamental y deberá ser potenciado con la plataforma adecuada, todo ello en consonancia con la visión y la estrategia corporativa a nivel de informática. [5]

#### **1.3.3 Gestión (informática)**

La informática en el ámbito empresarial e institucional ha tomado fuerza hasta convertirse en una pieza esencial y desde sus orígenes, se aplica para facilitar a las personas involucradas en las actividades que generan una rutina tediosa, con ello mejora la optimización y la producción en la empresa.

### **1.3.4 Información (informática)**

La información se confunde con el dato, sin embargo, la diferencia es que la información son los datos procesados por el mecanismo informacional durante el proceso descrito, de allí nace el nombre “proceso informacional”, entendiéndose que los datos son la materia prima base y la información es el producto terminado fruto del proceso que puede ser utilizado inmediatamente. [6]

Se puede definir la información como una mercancía de cambio de un proceso donde alguien resulta informado, expresando un estado de ánimo y a través de la informática sentir una satisfacción tras alcanzar el acceso a la respuesta por una inquietud o simplemente conocer algo más. [6]

### **1.3.5 JavaScript**

JavaScript, cuya abreviatura es JS, nació en 1995 por la WWW o Internet Comercial. En un inicio fue diseñado por Netscape Communications hoy conocido como Mozilla Foundation que lo integraron como un tipo de plugin en los navegadores. [7]

JavaScript es un lenguaje de programación del tipo interpretado que se define también como un lenguaje orientado a objetos y en la actualidad JavaScript está integrado en el motor de los navegadores web para ser utilizado en el lado del cliente, enfocado en mejoras de la interfaz de usuario y páginas web dinámicas, sin embargo, se lo utiliza también en el lado del servidor, de allí nace un notable ejemplo llamado NodeJS. [7]

### **1.3.6 NodeJS**

En su página oficial, NodeJS se define como una plataforma construida con el motor de JavaScript de Chrome para construcción de aplicaciones rápidas y escalables. [8]



*Figura 1. Logo NodeJS*

*Fuente: [8]*

NodeJS es un entorno de ejecución para JavaScript creado por Ryan Dahl a inicios de 2009, en un principio su diseño fue orientado para desarrollar aplicaciones web. [9]

NodeJS contiene a más de su entorno de ejecución, un conjunto de librerías básicas o módulos nativos para construir aplicaciones orientadas a redes de comunicación, facilitando funciones para el manejo de archivos y parte de estas librerías o módulos permiten interactuar con el sistema operativo mediante funciones que cumplen un estándar POSIX. [9]

### **1.3.7 Framework**

En desarrollo de software, un framework es una estructura genérica definida con el propósito de facilitar patrones automatizados para el desarrollo e implementación de aplicaciones, reduciendo tiempos y orientándolas a un determinado propósito.

Un framework tiene el propósito de estructurar y normalizar el código de un sistema brindando una base sólida en el desarrollo de aplicaciones concretas. Por lo general, son construidos con base en lenguajes orientados a objetos con características de modularidad en sus componentes y optima reutilización de código. [10]

### **1.3.8 VueJS**

VueJS es un framework de desarrollo web progresivo basado en JavaScript, creado por Evan You, quien fue desarrollador en Google por varios años. La herramienta es progresiva por la división de sus librerías de acuerdo con una responsabilidad específica, donde el desarrollador tiene la potestad de añadir varios módulos, acorde a

las necesidades del contexto, a diferencia de otros frameworks de desarrollo web que incluyen todas las funcionalidades. [11]

VueJS soporta las Single-Page Application (SPA), conocidas como aplicaciones de una sola página, es decir que todas las pantallas se muestran en una misma página sin necesidad de recargarla, debido a que existe un único punto de entrada donde se aloja el archivo index.html y la aplicación trabaja mediante vistas que se pueden intercambiar dentro de la misma página. [12]

VueJS está orientado a ser un framework de uso incremental con una librería central enfocada a la visualización, facilitando la integración con otros proyectos y librerías existentes. [13]



*Figura 2. Logo VueJs*

*Fuente: [13]*

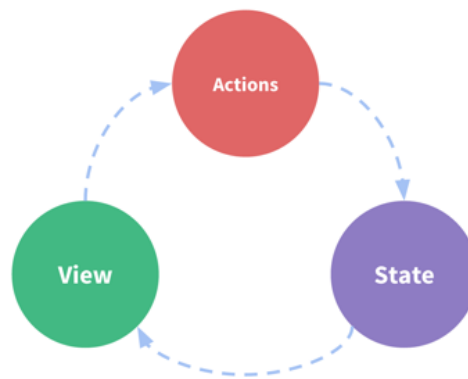
### **Características de VueJS**

- Patrón MVVM (Model-View-ViewModel). Este patrón separa a la lógica de la aplicación, de la interfaz de usuario ya que los componentes contienen un mecanismo doble 'data-binding' que manipulan el estado de la aplicación y además VueJS se encarga de renderizar el HTML. [11]
- Reactividad. Los componentes visuales de VueJS reaccionan a masivos eventos sin perjudicar el rendimiento
- Virtual DOM (Modelo de Objeto del Documento). VueJS debido a su reactividad hace cambios en el DOM en todo momento, si se cambia el valor de datos, la lógica proporcionada modificará automáticamente el DOM.

## Patrón de gestión de estado

Es una aplicación autónoma que tiene las siguientes partes:

- State. Es la fuente de verdad que impulsa la aplicación
- View. Mapeo declarativo del state
- Actions. Las formas en que el state puede cambiar de acuerdo con las entradas que realiza el usuario desde el View.



*Figura 3. Flujo de datos unidireccional de Vue*

*Fuente: [13]*

### 1.3.8.1 Vuex

Vuex es una librería externa creada por VueJS como un patrón de gestión de estado, que permite crear datos y funciones globales de manera centralizada a través de una tienda y así tenerlos a disposición desde los componentes para que puedan ser manipulados por el desarrollador acorde a sus necesidades. [13]

#### Ciclo de Vuex

Antes de conocer la funcionalidad de Vuex, se definen los siguientes términos con mayor enfoque técnico:

**State:** es el árbol de estado único, más conocido como “fuente de la verdad” ya que contiene todo el estado de nivel de la aplicación desde una única tienda.

Un solo árbol de estado da facilidad para localizar una parte específica del state y permite tomar snapshots del state actual de la aplicación con fines de depuración.

**Getters:** Esta propiedad permite filtrar una determinada lista de artículos en la tienda cuando algunos componentes lo requieren.

**Mutation:** La única forma de cambiar el state en determinada tienda de Vuex es a través de mutaciones.

**Actions:** Son similares a las mutaciones, a diferencia que el lugar de mutar el state, un action comente mutaciones, además estas acciones pueden contener operaciones asincrónicas arbitrarias.

Vuex contiene un almacén o tienda de datos que contiene el state (estado) de la aplicación y se altera por las mutations (mutaciones) que se ejecutan como respuesta a los actions (acciones) que invocan un evento de mutación a través del dispatch (despachador).

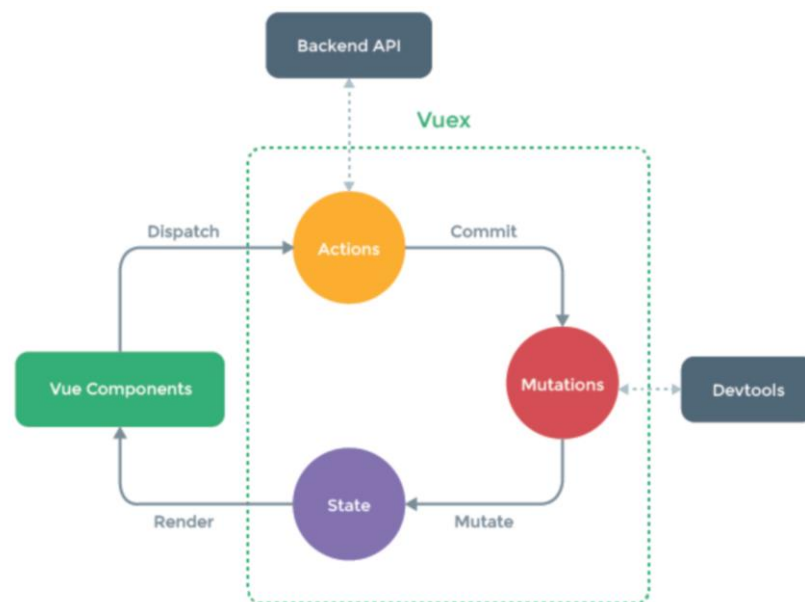


Figura 4. Flujo de datos en una aplicación con Vuex

Fuente: [13]

Los cambios establecidos, son automáticamente reflejados dentro de la aplicación, debido a que los componentes de Vue están vinculados reactivamente a través de sus captadores.



### 1.3.8.2 VueJS CLI

VueJS proporciona a los desarrolladores un CLI (interfaz de línea de comandos) en su ecosistema, que permite dar instrucciones a través de una línea de texto simple ayudando a la rápida construcción de una SPA.

Vue CLI permite que el desarrollador estructure el código, siguiendo una serie de buenas prácticas para la escalabilidad del producto según su necesidad, de esta manera genera plantillas y componentes con la opción de personalizarlos a gusto. [11]

#### Componentes de VueJS CLI

CLI, es un paquete npm instalado globalmente que proporciona el comando ‘vue’ para la utilización de comandos como: ‘vue create’ para crear nuevos proyectos, ‘vue serve’ para ejecutar el servidor local y poder visualizar el proyecto en un navegador, ‘vue ui’ para acceder a la interfaz gráfica donde se pueden administrar los proyectos. Finalmente, con la instalación de CLI, se incluye la librería Vuex.

#### Interfaz gráfica de Vue (UI)

VueJS contiene una interfaz gráfica que permite la administración de proyectos.

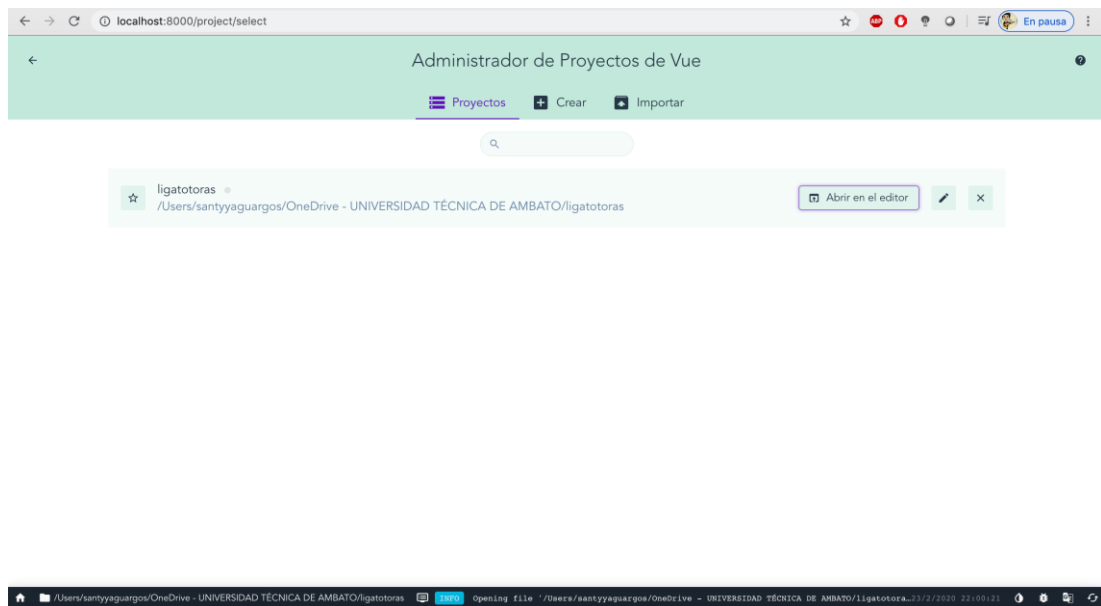


Figura 5. Interfaz Gráfica (UI) Administrador de proyectos de Vue

Fuente: [13]

### 1.3.8.3 Vuetify

Vuetify es una librería de componentes para interfaz de usuario lanzada desde 2016, dedicados para el marco de Vue, que proporciona a los desarrolladores todo lo necesario para la creación de aplicaciones con Material Design. [14]









*Figura 6. Logo de Vuetify*

*Fuente:*

Vuetify es expansivo, potente y soporta todos los navegadores modernos, facilitando todas las herramientas para crear aplicaciones robustas, proporcionando las siguientes características:

- Más de 80 componentes
- Directivas personalizables por el usuario
- Construido en base a Material Design
- Diseñado para dispositivos móviles y ordenadores
- Soporte para desarrolladores
- Se puede añadir a un proyecto de Vue CLI como plugin.

Vuetify tiende a ser una de las mejores, debido a una gran cantidad de componentes, soporte empresarial, soporte a largo plazo, actualizaciones semanales en su ecosistema, entre otras.

| Vue Framework Comparison 2020              |   |   |   |   |  |  |
|--|---|---|---|---|--|--|
| Features                                   |  Vuetify |  Vue Bootstrap |  Buefy |  Element UI |  Vue Material |  Quasar Framework |
| Total components*                          | 82+   | 45+   | 41+   | 86+   | 56+  | 81+  |
| Full Accessibility and Section 508 support | ✓   | ✓   | ✓   |   |  |  |
| Server Side Rendering support              | ✓   | ✓   | ✓   | ✓   | ✓  | ✓  |
| Long-term Support                          | ✓   |   |   |   |  |  |
| Release cadence**                          | Weekly  | Bi-Weekly   | Bi-Monthly  | Bi-Weekly   | Yearly   | Bi-Weekly  |
| Treeshaking                                | Automatic   | Manual  | Manual  | Manual  | Manual   | Automatic  |
| RTL support                                | ✓   | ✓   |   | ✓   |  | ✓  |
| Premium Themes                             | ✓   | ✓   |   |   | ✓  |  |
| Business and Enterprise support            | ✓   |   |   |   |  |  |

\*Based on Top-level components only.  
\*\*Based on average of all Major/Minor/Patch releases over the last 12 months.

Figura 7. Tabla de comparaciones entre librerías de diseño para Vue

Fuente: [14]

### 1.3.8.4 Vuelidate

Vuelidate es una librería en el marco de Vue dedicada a la validación de manera simple y ligera basada en modelos. [15]

Características:

- Desacoplado de plantillas
- Librería minimalista libre de dependencias
- Soporte para validación de colecciones
- Soporte para modelos anidados
- Valida diferentes fuentes de datos: captadores de Vuex, valores calculados.



Figura 8. Logo de Vuelidate

Fuente: [15]

### **1.3.9 Progressive Web App (PWA)**

Las Progressive Web App (PWA) o Aplicaciones Web Progresivas son tecnologías que combinan características de una web y una app nativa, es decir utilizan las últimas tecnologías disponibles en los navegadores, que ofrecen a las aplicaciones un comportamiento casi idéntico al de una aplicación nativa.

El navegador web es el cliente universal que permite acceder e interactuar con la información alojada en diferentes sitios web y está disponible en cualquier plataforma, de allí el salto evolutivo para el desarrollo de aplicaciones enfocadas a un entorno móvil que envuelve la tecnología web y recursos locales del dispositivo.

Este tipo de aplicaciones, a diferencia de una app nativa, son inicialmente accesibles desde un navegador en lugar de ser instaladas en el dispositivo luego de visitar la respectiva tienda.

Al ejecutarse una PWA, ofrece una visibilidad y experiencia de usuario idéntica al uso de una aplicación nativa y de allí se manejan recursos del dispositivo una vez se crea el acceso directo en la pantalla inicial, empleando así contenido en el cache a través de estrategias que mejoran la experiencia de usuario cuando no tiene conexión a internet.

#### **Proggresive (progresivas)**

En este término, las PWA son aplicaciones que incrementan su funcionalidad, en base a la capacidad del dispositivo, por lo cual se vuelven poderosas a medida que se utilizan. [16]

#### **Web**

Este tipo de aplicaciones se construyen con estándares de desarrollo web, utilizando lenguajes de etiquetas, estilos y programación. [16]

## **App**

Las PWA se caracterizan por tomar las características más destacables de las App ya que pueden instalarse en los dispositivos móviles, ejecutarse offline, tener un diseño bajo parámetros de estilo y el uso de notificaciones push. [16]

- **Características de una Aplicación Web Progresiva**

**Rapidez:** este tipo de aplicaciones deben tener el mínimo tiempo de carga para que la experiencia del usuario mantenga la comodidad necesaria cada vez que acceda a la PWA, siendo una de las características que mayor demanda tiene en cualquier ámbito tecnológico.

**Atractivo:** el diseño de estas aplicaciones debe tener el mayor cuidado ya que se deben cumplir parámetros de adaptabilidad en cualquier tipo de dispositivo que alcance esta tecnología, mostrando un fuerte impacto de amigabilidad al usuario.

**Fiabilidad:** estas aplicaciones deben estar disponibles más aun cuando el dispositivo pierda conexión evitando los errores incómodos que un navegador despliega.

- **Ventajas de una Aplicación Web Progresiva**

- La instalación de una PWA es directamente desde el navegador sin ocupar demasiado espacio en el dispositivo.
- La no dependencia del acceso a este servicio desde una tienda de Apple, Google o Microsoft, sin esperas de actualizaciones ni revisión.
- Funcionamiento offline, es decir sin conexión a internet.
- Una rapidez de hasta 4 veces más que cualquier web.
- Es ligera a diferencia de las aplicaciones nativas.
- Diseño responsivo (multiplataforma), que se adapte a los diferentes tamaños de pantalla en los dispositivos, sean estos ordenadores, tabletas o smartphones.
- Utilizan funcionalidades de aplicaciones nativas, como las notificaciones push.

Las Aplicaciones Web Progresivas mantienen el lenguaje de programación a nivel web, diferenciándolas de las aplicaciones nativas y también contienen otros elementos que los diferencian de las aplicaciones web.

La estructura de una página web (HTML, CSS y JavaScript) es el elemento básico que contiene una PWA y los elementos diferenciadores son el Service Worker y la descripción de la aplicación contenida en un manifiesto (manifest).

- **Service Worker**

Es un archivo de JavaScript pendiente de sucesos que permiten a un sitio web trabajar en segundo plano, está definido principalmente en la estructura de una PWA siendo sustento de su funcionamiento y proporcionándola para gestionar memoria caché, trabajar offline, notificaciones push y algunas otras características.

Un Service Worker, técnicamente, se considera un archivo con una secuencia de comandos que se ejecutan en segundo plano dentro de un navegador y escucha eventos independientemente de si una página web está o no abierta.

### Ciclo de Vida de un Service Worker

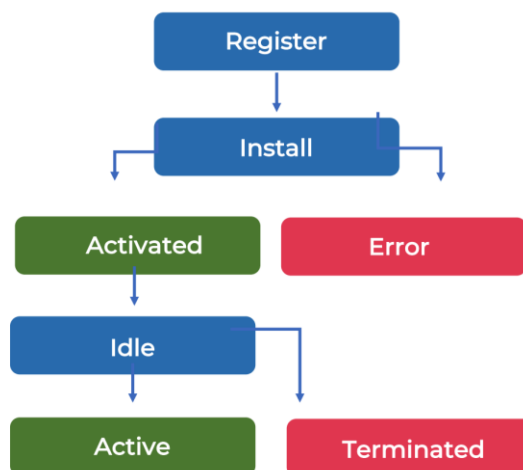


Figura 9. Instalación de un Service Worker

Elaborado por: Santiago Yaguargos

El ciclo de vida de un Service Worker se ejecuta en segundo plano independientemente de la web, una vez se instale por primera vez al iniciar la aplicación, facilitando grandes experiencias al usuario.

Inicialmente, un Service Worker deber ser registrado para que proceda a instalarse automáticamente en el navegador, este evento ocurre una sola vez y puede fallar únicamente cuando la promesa que llama al evento sea errónea.

Durante la etapa de instalación, se almacenarán los datos de cache estáticos y otros datos en cache que se establezcan a través de estrategias cuando el aplicativo maneje la conexión a internet.

Una vez que el Service Worker esté instalado correctamente, tomará el control del cliente, listo para utilizarlo y controlar eventos, de esta manera pasa al estado activo.

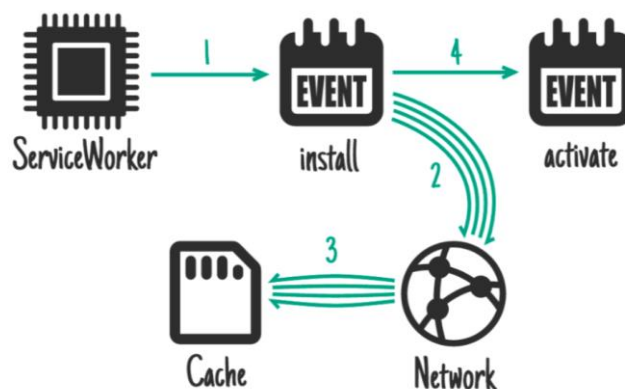


Figura 10. Proceso de inicialización de un Service Worker

Fuente:

Para el funcionamiento correcto de un Service Worker deben cumplirse los siguientes requerimientos:

### **Soporte del navegador**

En la actualidad, la mayoría de los navegadores ha implementado el soporte para los Service Worker y algunos restantes continúan trabajando en el proceso de desarrollo.

El navegador Chrome a través de la herramienta para desarrolladores muestra el estado de un Service Worker, como indica la Fig. 11.

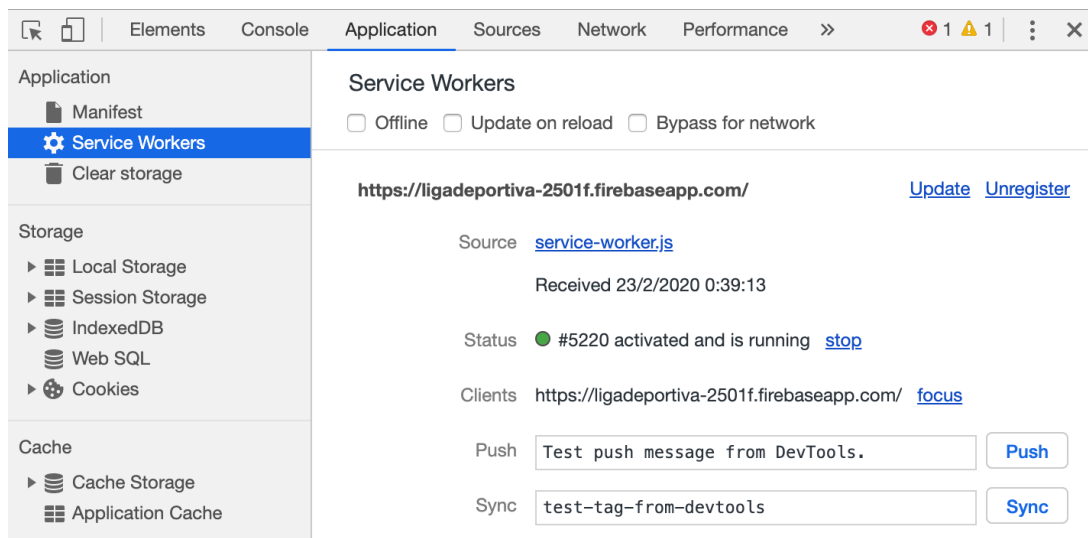


Figura 11. Service Worker instalado en una aplicación web

Elaborado por: Santiago Yaguargos

## Protocolo https

Para el registro de un Service Worker es un requisito esencial que el sitio web entregue un protocolo seguro, sin embargo, en la fase de desarrollo es oportuno utilizar el localhost ya que se considera un servidor seguro.

- **Manifiesto**

El Manifest o Manifiesto es un archivo descrito en formato JSON que comunica al navegador el comportamiento de una Aplicación Web Progresiva cuando se instala en un dispositivo. El archivo de manifiesto proporciona información como nombre, autor, íconos, descripción y la URL a la que se dirige cuando se inicia la aplicación.

El comportamiento cuando inicializa la aplicación de acuerdo con el archivo de manifiesto es el siguiente:

- Establecer el nombre corto de la aplicación y los distintos tamaños de imagen para que se adapten a diferentes dispositivos.
- Durante el proceso de descarga de recursos, establecer un estado intermedio para dinamizar el tiempo de carga.
- Establecer una configuración predeterminada en el navegador que evite transiciones bruscas.



Proporcionar la información base y necesaria para que se ejecuten las características de una Aplicación Web Progresiva.

## Elementos del manifiesto

```
{
  "name": "LIGA DEPORTIVA PARROQUIAL TOTORAS",
  "short_name": "LDPT",
  "icons": [
    {
      "src": "./img/icons/android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "./img/icons/android-chrome-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
  "start_url": "./",
  "display": "fullscreen",
  "background_color": "#012e67",
  "theme_color": "#012e67"
}
```

*Figura 12. Elementos del manifiesto*

*Elaborado por: Santiago Yaguargos*

**name:** determina el nombre que aparece en el banner de instalación de la aplicación.

**short\_name:** establece el texto que se incluye junto al icono en el inicio de pantalla.

**icons:** define un array de imágenes, que son los iconos que se establecerán en el inicio de pantalla. Cada objeto contiene los siguientes valores:

- **src:** ruta de acceso al archivo de la imagen,
- **sizes:** establece las dimensiones de la imagen,
- **type:** según el tipo de archivo de la imagen.

**schema:** es un elemento opcional pero recomendado, que contiene una URL que proporciona propiedades al navegador en la validación del manifiesto.

start\_url: aloja la URL de inicio de la aplicación que será cargada en el navegador, de esta manera evita que la aplicación inicie en la vista donde el usuario se encuentra al usar la aplicación mientras la agrega a la pantalla de inicio.

display: determina al navegador, la interfaz de usuario que se visualizará al iniciar la aplicación desde un dispositivo. Los valores disposición del desarrollador son:

- fullscreen: la aplicación web ocupa la totalidad del área visual disponible, sin elementos que controlan la navegación.
- standalone: la aplicación se visualizará como una aplicación independiente, ocultando los elementos que controlan la navegación.
- minimal-ui: similar a standalone, proporcionando elementos mínimos del que controlan la navegación y pueden variar dependiendo del navegador.
- browser: la aplicación se abrirá como en un navegador estándar.

scope: determina al conjunto de direcciones URL que el navegador web considera en su aplicación y se utiliza para decidir cuando el usuario abandona la aplicación. Es decir, cuando el usuario navega fuera del contexto de la aplicación, continuara como un web normal.

background\_color: define el color de fondo para cuando la aplicación se inicialice.

theme\_color: determina el color del tema de la interfaz de usuario en la barra de herramientas.

orientation: establece la orientación de la aplicación en los diferentes dispositivos y el usuario la podrá visualizar, ya sea horizontal o vertical, determinando los valores landscape o portrait respectivamente.

description: determina una descripción general de lo que hace la aplicación. [17]

#### • Estrategias de caché en el Service Worker

Las PWA se caracterizan por brindar al usuario el acceso offline o sin conexión, por lo tanto, es necesario manejar la configuración y el uso de los archivos almacenados

en caché mediante diferentes estrategias, como respuesta a las solicitudes que realiza el usuario:

- **cacheFirst:** prioriza a los datos que existan en la caché frente a los archivos que se encuentren online, limitando el ancho de banda. Si no existe una respuesta de la caché, la estrategia intentará una respuesta con archivos desde la red.
- **cacheOnly:** responde únicamente cuando existan datos en la caché, caso contrario falla la petición.
- **networkFirst:** prioriza el acceso a datos existentes en la red, en caso de fallo a la petición, se entregarán los datos de la caché.
- **networkOnly:** requiere obligatoriamente la existencia de conexión a la red, para que puedan accederse a datos del servidor y se utiliza cuando se trate de peticiones que no están almacenadas en la caché.

### **1.3.10 Base de Datos No SQL**

Una Base de Datos No SQL también conocida como Base de Datos No Relacional tienen su diseño enfocado para modelar datos específicos y contienen esquemas flexibles para la creación de aplicativos modernos. La principal característica en el funcionamiento de este tipo de base de datos es su rendimiento a gran escala. [18]

Las bases de datos No SQL son optimizadas para el desarrollo de aplicaciones que utilizan grandes volúmenes de datos debido a que utilizan una gran variedad de modelos de datos para acceder y administrarlos. [18]

Google desarrollo inicialmente este tipo de base de datos para resolver necesidades específicas, siendo el caso de los los primeros sistemas No SQL conocidos como BigTable y por otro lado, el sistema DynamoDB desarrollado por Amazon. [19]

#### **Características de una Base de Datos No SQL**

- **Flexibilidad:** son ideales para datos semiestructurados y no estructurados debido a su modelo de datos flexible, que permite un desarrollo rápido e iterativo.

- **Escalabilidad:** una base de datos No SQL mantiene su funcionamiento y calidad, aunque el número de usuarios aumente.
- **Alto rendimiento:** son optimizadas para patrones de acceso y modelos de datos específicos, permitiendo un mayor rendimiento.
- **Altamente funcional:** este tipo de base de datos facilitan API's con un alto nivel de funcionalidad y tipos de datos que están diseñados específicamente de acuerdo al respectivo modelo de datos. [18]

### 1.3.11 Firebase

Es una plataforma que facilita el desarrollo de aplicaciones web y móviles. Fue creada por James Tamplin y Andrew Lee en 2012, luego fue adquirida por Google en 2014.

Firebase esta integrada en la nube de Google Cloud Platform, aportando a los desarrolladores varias facilidades durante la creación de aplicaciones, además de una rica documentación gratuita de todos los servicios que esta plataforma ofrece. [20]



*Figura 13. Logo de Firebase*

*Fuente: [21]*

- **Firestore Realtime Database**

Es una base de datos No SQL con el formato de datos tipo JSON alojada en la nube que Firebase ofrece para el almacenamiento y sincronización de datos en tiempo real, además proporciona acceso a un BaaS (backend as a service). [21]

La sincronización que maneja Firebase Realtime Database al compartir una instancia, proporciona a los diferentes usuarios el acceso desde diferentes dispositivos en tiempo real y cuando se modifique la instancia, se almacenará en la nube e inmediatamente se notificará al resto de dispositivos.

Una ventaja al usar este servicio de Firebase, es que si existe una pérdida de conexión cuando un determinado usuario realiza cambios, los datos se guardan en una caché local y se sincronizarán automáticamente cuando exista conexión.

- **Firebase Authentication**

Es un servicio que ofrece Firebase, también conocido como Firebase Auth, que permite la autenticación de los usuarios proporcionando servicios de backend, SDK y bibliotecas UI. Firebase Auth soporta inicio de sesión a través de correo electrónico, números de teléfono y los proveedores de identidad federada como son Facebook, GitHub, Twitter, Google, entre otros. [21]

Firebase Auth facilita el desarrollo de servicios de autenticación manteniendo las características de seguridad para los usuarios para con ello evitar el clásico desarrollo de métodos de inicio de sesión, además aporta funcionalidades como verificación y recuperación de cuenta a través de correo electrónico o SMS en tiempo real. [21]

- **Firebase Cloud Storage**

Es un servicio similar Firebase Realtime Database, con enfoque específico en el almacenamiento de objetos, archivos, imágenes, audio, video y contenido generado por los usuarios, proporcionando seguridad en la carga y descarga, sin importar la calidad de la red. [21]

- **Firebase Hosting**

Es un servicio de hosting seguro y rápido para las aplicaciones web. Facilita la implementación de aplicaciones mediante un solo comando. [21]

- **Firestore Cloud Messaging**

Es un servicio de mensajería, relacionado con el envío de notificaciones push en las diferentes aplicaciones. [21]

### **1.3.12 PostgreSQL**

PostgreSQL es un sistema gestor de base de datos relacional de código abierto orientado a objetos. [22]

Las principales características son:

- Flexible a un amplio conjunto de lenguajes de programación.
- Multiplataforma.
- Herramienta de administración PGAdmin
- Estable, eficiente y robusto.



PostgreSQL

*Figura 14. Logo de PostgreSQL*

*Fuente: [22]*

### **1.3.13 Spring Framework**

Spring es un framework para desarrollo de aplicaciones basado en Java Enterprise (JEE), que fue escrito por Rod Johnson y lanzado oficialmente en el año 2004 como plataforma de código abierto.

Spring Framework se caracteriza por facilitar la construcción de aplicaciones con código de alto rendimiento, liviano y reutilizable, en conjunto con la robustez brindada por Java Enterprise.

### 1.3.13.1 Spring Boot

Es un proyecto de Spring Framework utilizado para las configuraciones y proyectos de arranque, enfocado principalmente en una filosofía de configuración mínima en el inicio de un proyecto con Spring, puesto que valores predeterminados satisfacen las necesidades del usuario.

Spring Boot se caracteriza por la facilidad que le da usar al desarrollador y de allí el éxito del proyecto, entre ellas el servicio en línea llamado spring.io para la creación de proyectos. [23]



*Figura 15. Logo de SpringBoot Java*

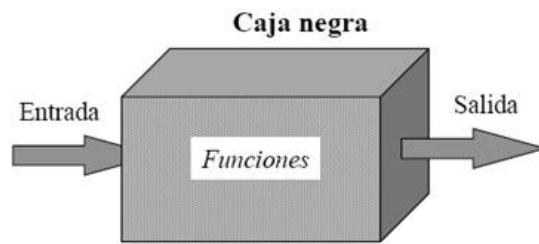
*Fuente: [24]*

### 1.3.14 Metodología ágil

Una metodología ágil para el desarrollo de software es un método y técnica que se utilizan para construir software de manera rápida y estructurada con un enfoque al desarrollo iterativo e incremental para la administración proyectos, fueron creadas para mejorar la calidad y los métodos tradicionales de desarrollo.

### 1.3.15 Pruebas de caja negra

Son pruebas que se aplican enfocándose en los requisitos funcionales del sistema ignorando la estructura de control y permiten obtener un conjunto de entradas que ejerciten completamente estos requisitos. [25]



*Figura 16. Estructura de la prueba de caja negra*

*Fuente: [25]*

Varios autores determinan que estas pruebas pueden encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.



## **CAPÍTULO II METODOLOGÍA**

### **2.1. Materiales**

- Computadora portátil MacBook
- Framework de desarrollo Frontend VueJS
- Framework de desarrollo Backend Spring
- Plataforma Firebase
- Editor de código Visual Studio Code
- Entorno de programación Eclipse
- Administrador de base de datos Postgre PGAdmin

### **2.2. Métodos**

#### **2.2.1. Modalidad de la investigación**

El presente trabajo tiene las siguientes modalidades de investigación:

##### **Modalidad de Campo**

Se considera esta modalidad debido a que el investigador se dirigirá al lugar donde ocurren los hechos para explorar y obtener la información relacionada con los procesos que Liga Deportiva Parroquial Totoras requiere automatizar previo y durante el desarrollo de un campeonato deportivo. La técnica utilizada será: encuesta.

##### **Modalidad Bibliográfica o Documentada**

Se considera esta modalidad ya que se recurre a diferentes fuentes obtenidas de libros, artículos científicos, tesis desarrolladas en Universidades para profundizar un enfoque claro con la normativa adecuada respecto al tema de la investigación.

## Modalidad aplicada

Por la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria.

### 2.2.2. Población

La población para el proyecto de investigación se conforma de la siguiente manera.

| No | Tipo de Población      | Descripción  | Cantidad |
|----|------------------------|--|----------|
| 1  | Dueño del problema     | Encargado de poner a consideración un problema institucional.  | 1        |
| 2  | Directorio             | Autoridades encargadas de la administración de la institución y organización de torneos deportivos.  | 10       |
| 3  | Secretario/a           | Persona encargada de llevar a cabo el proceso para inscripciones de nuevas instituciones deportivas filiales con sus respectivos jugadores y publicar información en tiempo real durante el desarrollo de un campeonato. | 1        |
| 4  | Comisión de Disciplina | Personal encargado de sanciones y registro de información de jugadores durante la ejecución de un campeonato.  | 5        |
| 5  | Comisión Técnica       | Personal encargado de la organización de encuentros deportivos, tablas de posiciones, goleadores, entre otros.   | 3        |
| 6  | Usuarios filiales      | Presidentes de las instituciones deportivas filiales encargados de inscribir sus equipos participantes en un determinado campeonato.   | 25       |
| 7  | Desarrollador          | Encargado del desarrollo de la PWA y su implementación   | 1        |

*Tabla 1. Población.*

*Elaborado por: Santiago Yaguargos*

### 2.2.3. Recolección de Información

Para la recolección de información se utilizará un cuestionario para encuesta.

| Preguntas Básicas             | Explicación   |
|-------------------------------|---|
| ¿Para qué?                    | Analizar los procesos de gestión de datos e información que requieren ser automatizados previo y durante un campeonato deportivo. |
| ¿De qué personas u objetos?   | Involucrados directamente a Liga Deportiva Parroquial Totoras.  |
| ¿Sobre qué aspectos?          | Gestión de Datos e Información  |
| ¿Quién, quiénes?              | Investigador  |
| ¿Cuándo?                      | septiembre - enero 2019   |
| ¿Dónde?                       | Liga Deportiva Parroquial Totoras   |
| ¿Cuántas veces?               | 1 vez   |
| ¿Qué técnicas de recolección? | Encuesta  |
| ¿Con qué?                     | Guía de encuesta.   |

*Tabla 2. Recolección de la información.*

*Elaborado por: Santiago Yaguargos*

### 2.2.4. Procesamiento y análisis de datos

Para el procesamiento y análisis de la información que determinan los procesos específicos que requieren ser automatizados previo y durante un campeonato deportivo. Se aplicó los siguientes procedimientos:

- **Procesamiento de la Información**

- 1) Elaboración de cuestionario para encuesta.
- 2) Aplicar la encuesta a los involucrados

- 3) Tabulación de la información obtenida
- 4) Estudio estadístico de datos para presentación de resultados

• **Análisis e interpretación de resultados**

- 1) Análisis crítico de la información recolectada
- 2) Análisis de los resultados estadísticos en base a los objetivos
- 3) Interpretación de resultados
- 4) Comprobación de los objetivos
- 5) Establecimiento de conclusiones y recomendaciones

**2.2.5. Metodología de Desarrollo**

En el presente proyecto se aplica la metodología XP (Xtreme Programming) para el desarrollo de la PWA, ya que esta es una metodología ágil, donde se puede considerar los cambios no previstos en el desarrollo de sus aplicaciones.

|             | <b>XP</b>   | <b>SCRUM</b>   | <b>KANBAN</b>  |
|-------------|---|--|--|
| Descripción | Metodología ágil basada en flexibilidad, satisfaciendo necesidades del cliente.   | Metodología ágil pensada para aumentar la productividad.   | Metodología ágil complementaria a Scrum que proporciona un flujo de trabajo para dividir el proceso en fases.                                  |
| Principios  | <ul style="list-style-type: none"> <li>- Simplicidad</li> <li>- Retroalimentación</li> <li>- Comunicación</li> <li>- Valentía</li> <li>- Respeto</li> </ul> | <ul style="list-style-type: none"> <li>- Desarrollo iterativo</li> <li>- Entrega continua</li> <li>- Control empírico</li> <li>- Colaboración</li> </ul> | <ul style="list-style-type: none"> <li>- Visualización</li> <li>- En proceso</li> <li>- Priorización</li> <li>- Medición del tiempo</li> </ul> |
| Ventajas    | <ul style="list-style-type: none"> <li>- Adaptable a cambios.</li> <li>- Pruebas continuas</li> </ul>   | <ul style="list-style-type: none"> <li>- Mejora continua</li> <li>- Flexibilidad y adaptación</li> </ul>   | <ul style="list-style-type: none"> <li>- Flexibilidad</li> <li>- Eficacia</li> <li>- Promueve el trabajo en equipo.</li> </ul>                 |

|             |  |  |   |
|-------------|--|--|---|
|             | - Programación organizada<br>- Comunicación entre cliente y desarrolladores.             | - Predicción de tiempos.<br>- Revisión continua  | - Distribución del trabajo  |
| Desventajas | - Utilizada en proyectos a corto plazo.<br>- No se definen costo y tiempo de desarrollo. | - Demasiadas reuniones<br>- Delega responsabilidades fijas al equipo<br>- No apta en todos los proyectos | - Plazo de abastecimiento muy grande.<br>- No óptimo en todos los proyectos.<br>- Limitado. |
| Complejidad | Baja   | Alta   | Alta  |

*Tabla 3. Cuadro comparativo de metodologías ágiles.*

*Elaborado por: Santiago Yaguargos*

### **2.2.6. Desarrollo del Proyecto**

Las actividades para llevar a cabo el desarrollo del proyecto de investigación se basan en la metodología XP (Xtreme Programming) donde se contemplan los métodos, procesos y actividades dentro de las siguientes fases.

- **Fase 1. Exploración**

- Análisis de los procesos de gestión que requieren ser automatizados en Liga Deportiva Parroquial Totoras previo y durante un determinado campeonato deportivo.
- Análisis de los procesos de información que requieren ser automatizados en Liga Deportiva Parroquial Totoras previo y durante un determinado campeonato deportivo.

- **Fase 2. Planificación del proyecto**

- Levantamiento de historias de usuario.
- Estimación de tiempos y costos

- **Fase 3. Iteraciones**
  - Establecer un plan de entrega de la Aplicación Web Progresiva
  
- **Fase 4. Producción**
  - Modelado de la Base de Datos
  - Diseño de interfaces
  - Desarrollo de Software
  
- **Fase 5. Pruebas**
  - Implementación de la Aplicación Web Progresiva
  - Pruebas de funcionamiento

## CAPÍTULO III

### RESULTADOS Y DISCUSIÓN

#### 3.1. Análisis y discusión de resultados

##### 3.1.1. Recolección de la información

Para el desarrollo del proyecto se realizó la recolección de información mediante una encuesta, cuyos resultados servirán para determinar la factibilidad de la aplicación y el enfoque a los procesos que requieren ser automatizados.

La encuesta se aplicó a una población de 50 personas directamente involucradas con Liga Deportiva Parroquial Totoras, tales como, Directorio, Comisiones, presidentes y delegados de los diferentes clubes deportivos filiales.

Anexo 1. Cuestionario utilizado para la recolección y análisis de la información.

##### 3.1.2. Análisis de la información

El cuestionario para encuesta consta de 6 preguntas:

#### 1) ¿Cuántas horas al día utiliza algún servicio de internet (redes sociales, páginas informativas, entre otros)

##### Resultados:

50 respuestas

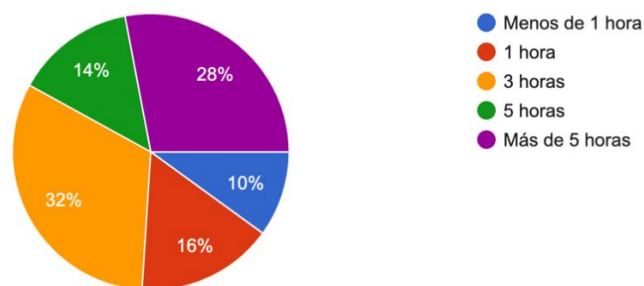


Figura 17. Resultado de Encuesta - Pregunta 1

Elaborado por: Santiago Yaguargos

**Análisis:** El 32% de las personas encuestadas utilizan un servicio de internet alrededor de 3 horas al día, el 28% más de 5 horas, el 16% aproximadamente 1 hora, el 14% aproximadamente 5 horas y el 10% menos de 1 hora.

**Conclusión:** El 74% de las personas encuestadas utilizan un servicio de internet de 3 a más horas en el día y, en conclusión, es factible que se desarrolle un servicio en línea.

## 2) ¿Qué dispositivos usa usualmente para acceder a un servicio con internet?

### Resultados:

50 respuestas

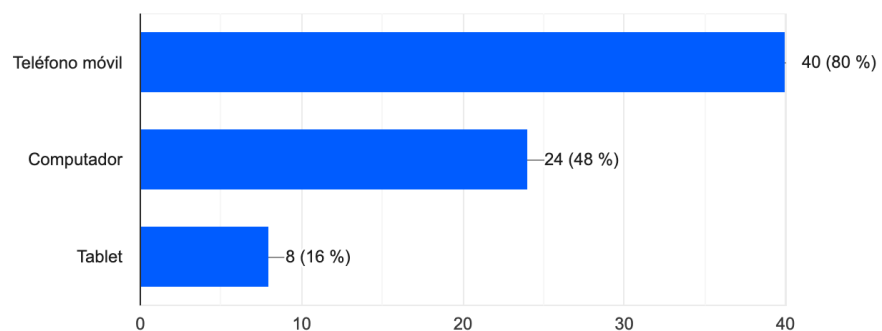


Figura 18. Resultado de Encuesta - Pregunta 2

Elaborado por: Santiago Yaguargos

**Análisis:** El 80% de las personas encuestadas utilizan un teléfono móvil para acceder al servicio de internet, el 48% utiliza un computador y el 16% utilizan tabletas.

**Conclusión:** Los teléfonos móviles y computadores son los dispositivos más comunes con los cuales los involucrados utilizan un servicio en internet, por tanto, es factible el desarrollo de una PWA y se adaptaran en los tres tipos de dispositivos.

## 3) ¿Usted considera que se ha llevado una correcta administración en el almacenamiento de datos en los clubes, equipos y jugadores?

### Resultados:



50 respuestas

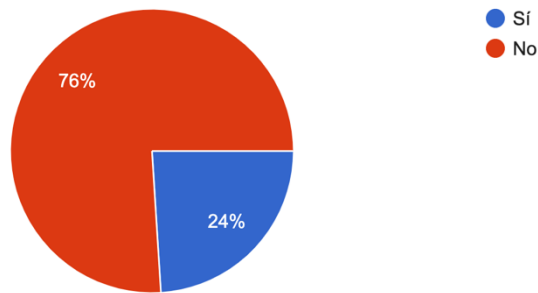


Figura 19. Resultado de Encuesta - Pregunta

Elaborado por: Santiago Yaguargos

**Análisis:** El 76% de las personas encuestadas consideran que no se ha llevado una correcta administración de datos en los clubes, equipos y jugadores, mientras que el 24% manifiesta que si se considera correcta.

**Conclusión:** Un alto porcentaje manifiesta la deficiente administración de datos por lo cual se generan problemas en la información.

**4) ¿Qué tan importante es para usted la transparencia de datos y la información que Liga Deportiva Parroquial Totoras brinda, previo y durante un campeonato?**

**Resultados:**

50 respuestas

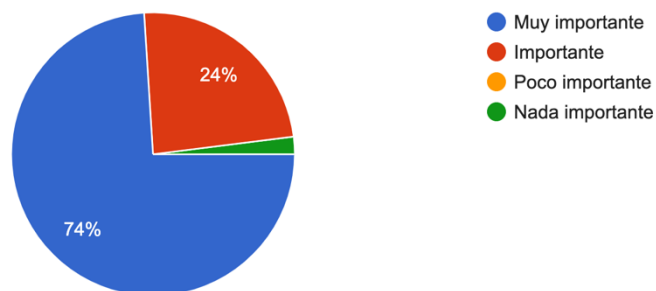


Figura 20. Resultado de Encuesta - Pregunta 4

Elaborado por: Santiago Yaguargos

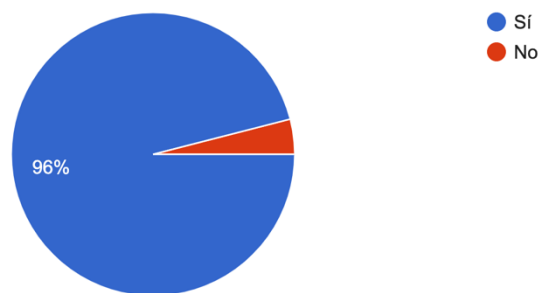
**Análisis:** El 74% de las personas encuestadas manifiestan que la transparencia de datos e información que Liga Deportiva Parroquial Totoras es muy importante, el 24% considera importante y el 2% no lo considera nada importante.

**Conclusión:** El 98% de las personas encuestadas muestran su importancia a como se dan tratamientos a los datos, por lo tanto, es vital el adecuado manejo de esta información que el ente almacena.

**5) ¿Desearía tener acceso a la información de los clubes, equipos y jugadores mediante una aplicación en tiempo real a través de su dispositivo móvil?**

**Resultados:**

50 respuestas



*Figura 21. Resultado de Encuesta - Pregunta 5*

*Elaborado por: Santiago Yaguargos*

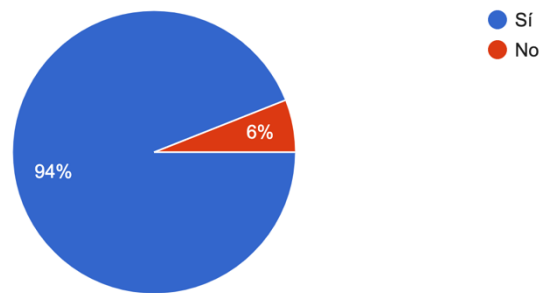
**Análisis:** El 96% de las personas encuestadas expresan su deseo al acceso de información de los clubes, equipos y jugadores que pertenecen a Liga Deportiva Parroquial Totoras mediante un aplicativo.

**Conclusión:** Es factible el desarrollo de una PWA ya que utilizan los recursos adecuados para su funcionamiento a nivel de dispositivos móviles, ordenadores y tabletas.

**6) ¿Desearía recibir noticias y toda la información que maneja Liga Deportiva Parroquial Totoras a través de notificaciones en su dispositivo móvil?**

**Resultados:**

50 respuestas



*Figura 22. Resultado de Encuesta - Pregunta 6*

*Elaborado por: Santiago Yaguargos*

**Análisis:** El 94% de las personas encuestadas expresa su deseo de acceder a la información que maneja Liga Deportiva Parroquial Totoras y recibir noticias a través de sus dispositivos móviles mientras que un 6% no muestra interés.

**Conclusión:** El desarrollo de la PWA es factible ya que abarca el uso de notificaciones push cuando se utiliza desde un dispositivo móvil y también es utilizado a nivel web desde mediante diferentes dispositivos que permitan el acceso mediante diferentes tipos de navegadores que soportan esta tecnología.

## **3.2. Desarrollo de la metodología**

### **3.2.1. Análisis de las herramientas para el desarrollo**

- **Base de Datos**

Para el presente proyecto de investigación, de acuerdo a la necesidad del usuario, se ha establecido el uso de Sistema Gestor de Base de Datos PostgreSQL siendo uno de los mas populares en la actualidad, que tiene un entorno gráfico llamado PGAdmin que facilita la administración de los datos durante el desarrollo, caracterizado por su estabilidad, confiabilidad, robustez, permitiendo una instalación ilimitada y multiplataforma con licencia pública GNU(GLP) que ayuda en la economía destinada al desarrollo de la aplicación.

- **Framework de desarrollo Frontend**

Para el presente proyecto de investigación se ha establecido el uso del Framework VueJs para el Frontend, siendo uno de los mas populares junto a React y AngularJS. Se realiza un análisis comparativo de VueJS frente a las dos tecnologías mencionadas, basándose en las características que presentan en el mundo del desarrollo web.

### **VueJS vs React**

Son dos frameworks que comparten muchas similitudes, como el uso de un DOM Virtual, el enfoque a una librería central y los componentes de vista reactivos y disponibles.

Debido a las similitudes mencionadas en el anterior párrafo, aunque React es un framework que supera en algunas características a VueJS por la riqueza en su ecosistema, es necesario un refinamiento en el análisis de ambas herramientas.

- **Rendimiento**

El rendimiento de ambas herramientas es comparable debido a los casos de uso mas comunes, VueJS tiene una implementación mayormente ligera del DOM Virtual.

React suele activar un re-renderizado de todo el subárbol de componentes iniciando por la raíz, cuando un componente cambia y para optimizarlo requiere el uso de estructuras de datos inmutables y implementación de otros componentes, a diferencia de VueJS ya que las dependencias de un componente se rastrean automáticamente mientras se renderiza, facilitando que el sistema sepa precisamente que componente debe renderizarse cuando cambia el estado.

Para facilidad de los desarrolladores, VueJS evita la necesidad de crear una clase dedicada a optimizadores de rendimiento y se enfoca en el desarrollo de la aplicación en si misma a medida que se vaya escalando.

## - **Escala**

Ambas tecnologías ofrecen soluciones de enrutamiento robustas cuando se trata de grandes aplicaciones. En cuanto a gestores de estado React utiliza soluciones como Flux/Redux, que pueden integrarse sin problema en Vue, sin embargo, Vue ofrece un modelo llamado Vuex que ofrece una superioridad en la experiencia de desarrollo.

Vue facilita que las librerías de enrutamiento y gestión de estados tengan soporte oficial y se actualicen con la librería central, en cambio React deja estas preocupaciones en manos de la comunidad y su ecosistema se vuelve fragmentado a pesar de tener mayor riqueza que Vue.

Vue facilita a la comunidad el uso de una interfaz grafica estable que sirve para la generación de proyectos CLI, incluyendo webpack, Browserify o ningún paquete de compilación, además permite la configuración personalizada con una amplia variedad de plantillas, según el propósito del proyecto. Estas características aun son limitadas para la comunidad React.

## **VueJS vs AngularJS**

Tienen similitud en la sintaxis, debido al acierto de muchas cosas en Angular JS que inspiró a Vue desde el inicio de su desarrollo intentando una mejora significativa.

## - **Complejidad**

Vue es mucho más simple que AngularJS en su diseño y en términos de API. Aunque el aprendizaje para crear aplicaciones no triviales toma menos de un día, AngularJS no es el caso.

## - **Flexibilidad y modularidad**

Vue ofrece una plantilla webpack que permite una configuración rápida y facilita el acceso a funciones avanzadas para que el desarrollador pueda tomar sus propias decisiones, siendo una solución flexible y modular ante la fuerte opinión que tiene AngularJS sobre la estructura de sus aplicaciones.

- **Vinculación de datos**

VueJS se enfoca en un flujo de datos unidireccional entre componentes, facilitando su razonamiento en aplicaciones no triviales, AngularJS utiliza la vinculación de dos vías entre diferentes alcances.

- **Rendimiento**

Vue tiene un rendimiento mucho más fácil de optimizar al no utilizar la denominada “comprobación sucia” a diferencia de AngularJS que se torna lento cuando existen muchos watchers ya que necesitan ser reevaluados cuando cambian de estado.

- **Framework de desarrollo Backend**

Para el presente proyecto de investigación se ha establecido el uso del Framework SpringBoot Java para el Backend, que permite desarrollar código de alto rendimiento y reutilizable, este último, de acuerdo con las intenciones del producto final, facilitará el desarrollo y la implementación de este tipo de aplicaciones, a nuevas instituciones que deseen adquirirla en un futuro.

### **3.2.2. Metodología ágil**

El desarrollo del presente proyecto de investigación se basa en la metodología ágil XP (Xtreme Programming), por ser diseñada para acatar cambios repentinos durante cualquier etapa del desarrollo, orientándose a la programación rápida y de calidad.

- **Fase 1. Exploración**

El objetivo durante esta fase del proyecto es el análisis de los procesos a automatizar. Para el desarrollo de la investigación se tiene una aplicación web que contendrá la parte administrativa bajo el rol Administrador quien será responsable de los procesos de gestión e información. El aplicativo deberá tomar características de una PWA y para lo cual es necesario un desarrollo web responsivo acorde a las necesidades de un dispositivo móvil.

## **Procesos de Gestión**

Liga Deportiva Parroquial Totoras abarca procesos de gestión con datos de sus clubes filiales con sus equipos y jugadores que participan durante un campeonato, sin embargo, se almacena físicamente fichas de los jugadores siendo un problema para manipular y realizar búsquedas de determinados datos que se requieren en cualquier momento previo o durante el desarrollo de un torneo.

Por otro lado, existe el proceso de habilitaciones, mismo que genera una gran cantidad de problemas por la transparencia de los datos por no mantener la información correcta de los jugadores que estén participando en determinado equipo bajo la norma legal de transferencias, préstamos y pases entre clubes.

## **Procesos de Información**

Liga Deportiva Parroquial Totoras cuenta con un sistema de escritorio utilizado por la Comisión Técnica, mismo que permite generar partidos y tablas de posiciones, mostrando un reporte en formato de imagen o pdf, que se publica a través de fotografías en redes sociales o en vitrinas de la institución durante extensos lapsos de tiempo y su alcance no es el necesario ya que para acceder a la información necesariamente deben activar una red social o visitar presencialmente la sede.

Liga Deportiva Parroquial Totoras guarda archivos de manera física de sus normativas como estatutos, reglamentos y documentos importantes que se mantiene al contacto con el Ministerio del Deporte, los cuales corren el riesgo de sufrir daños o pérdidas al no tenerlo al alcance de manera virtual, debido a que los clubes filiales tienen derecho al acceso de estos documentos en cualquier momento.

Liga Deportiva Parroquial Totoras genera información a través de redes sociales que no tiene el 100% del alcance al receptor ya que muchos de ellos no utilizan estos servicios a pesar de contar con dispositivos inteligentes.

- **Fase 2. Planificación del Proyecto**

### **Historias de Usuario**

El objetivo durante esta fase del proyecto es definir las historias de usuario con los principales involucrados con los procesos que se requieren automatizar.

Las historias de usuario manejan el lenguaje natural que resalta las necesidades del usuario final con la mayor simplicidad en la comunicación.

Para el levantamiento de las historias de usuarios se utilizará el siguiente modelo:

| <b>Historia de Usuario</b>      |                              |
|---------------------------------|------------------------------|
| <b>Número:</b>                  | <b>Usuario:</b>              |
| <b>Nombre Historia:</b>         |                              |
| <b>Prioridad en Negocio:</b>    | <b>Riesgo en Desarrollo:</b> |
| <b>Puntos Estimados:</b>        | <b>Iteración Asignada:</b>   |
| <b>Programador Responsable:</b> |                              |
| <b>Descripción:</b>             |                              |
| <b>Observaciones:</b>           |                              |

*Tabla 4. Modelo historias de usuario.*

*Elaborado por: Santiago Yaguargos*

El modelo contiene las siguientes especificaciones:

- **Número:** Identificador asignado a cada historia de usuario
- **Usuario:** Persona que describe la historia en lenguaje natural
- **Nombre Historia:** Título asignado a la historia de usuario acorde a los descrito
- **Prioridad en Negocio:** Cuantificador asignado a la historia de usuario, basado en la necesidad del negocio ya sea alta, media, baja.
- **Riesgo en Desarrollo:** Cuantificador asignado a la historia de usuario, basado en el impacto al desarrollar la misma en caso de fallo; ya sea alta, media, baja.
- **Puntos Estimados:** Es el lapso que toma para concluir la historia de usuario. Su valor es en días enteros, es decir aproximando al inmediato superior.
- **Iteración Asignada:** Numero de iteración en que se designa la historia de usuario.



- Programador Responsable: Persona encargada del desarrollo de la historia de usuario
- Descripción: Se relata en lenguaje natural la solución, en respuesta a la pregunta ¿Qué hacer?
- Observaciones: Descripción específica de los procesos relacionados a la historia de usuario.

### Historias de Usuario Aplicación Web Progresiva

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 1</b>  | <b>Usuario:</b> Usuario Final     |
| <b>Nombre Historia:</b> Diseño de interfaces - PWA  |                                   |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 5  | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> Diseñar las interfaces con características responsivas que permitan un acople a navegadores desde ordenadores, tabletas y dispositivos móviles. |                                   |
| <b>Observaciones:</b> El diseño de interfaces deberá tener un enfoque hacia aplicativos móviles, debido a que una PWA toma estas características.                   |                                   |

*Tabla 5. Diseño de interfaces - PWA.*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 2</b>  | <b>Usuario:</b> Usuario Final     |
| <b>Nombre Historia:</b> Visualizar información de LDPT  |                                   |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Baja |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El usuario final podrá acceder a la información básica acerca de la institución como son: historia breve, directorio y organigrama deportivo. |                                   |
| <b>Observaciones:</b> El requerimiento debe ser interactivo y la posibilidad de visualizar imágenes de mayor importancia.   |                                   |

*Tabla 6. Visualizar información de LDPT*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 3</b>   | <b>Usuario:</b> Usuario Final     |
| <b>Nombre Historia:</b> Visualizar información de Clubes   |                                   |
| <b>Prioridad en Negocio:</b> Alta  | <b>Riesgo en Desarrollo:</b> Bajo |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El usuario final podrá acceder a la información de los Clubes Filiales a Liga Deportiva Parroquial Totoras de manera interactiva.                          |                                   |
| <b>Observaciones:</b> Deberá reflejarse el logo del club deportivo, datos del presidente, año de fundación, un enlace a red social y los jugadores con los que tiene derechos. |                                   |

*Tabla 7. Visualizar información de Clubes*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 4</b>   | <b>Usuario:</b> Usuario Final     |
| <b>Nombre Historia:</b> Descargar documentación  |                                   |
| <b>Prioridad en Negocio:</b> Alta  | <b>Riesgo en Desarrollo:</b> Baja |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El usuario final podrá acceder a enlaces de descarga sobre documentación necesaria como estatutos, reglamentos, campeonatos, formatos de oficios, transparencia y actas. |                                   |
| <b>Observaciones:</b> El requerimiento se refiere a documentos necesarios de Liga Deportiva Parroquial Totoras.  |                                   |

*Tabla 8. Descargar documentación*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 5</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Inicio de Sesión  |                                   |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El usuario administrador podrá acceder al aplicativo web mediante una correcta autenticación. |                                   |
| <b>Observaciones:</b> El usuario administrador se autenticará con el correo de LDPT.                              |                                   |

*Tabla 9. Inicio de Sesión*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 6</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Ingresar noticias   |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá ingresar noticias acerca de Liga Deportiva Parroquial Totoras. |                                   |
| <b>Observaciones:</b> Es necesario llenar todos los campos requeridos.                                    |                                   |

*Tabla 10. Ingresar noticias*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 7</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Actualizar noticias  |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá actualizar los datos de las noticias acerca de Liga Deportiva Parroquial Totoras. |                                   |
| <b>Observaciones:</b> El administrador puede editar la imagen.   |                                   |

*Tabla 11. Actualizar noticias*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 8</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Eliminar noticias   |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 1      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá eliminar noticias acerca de Liga Deportiva Parroquial Totoras. |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción.                       |                                   |

*Tabla 12. Eliminar noticias*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                    |
|--|------------------------------------|
| <b>Número: 9</b>   | <b>Usuario:</b> Usuario Final      |
| <b>Nombre Historia:</b> Visualizar Noticias  |                                    |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Medio |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 1       |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                    |
| <b>Descripción:</b> El usuario final requiere acceder a un modulo donde se visualicen noticias publicadas de acontecimientos institucionales de mayor importancia. |                                    |
| <b>Observaciones:</b>  |                                    |

*Tabla 13. Visualizar Noticias*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                    |
|---|------------------------------------|
| <b>Número: 10</b>   | <b>Usuario:</b> Administrador      |
| <b>Nombre Historia:</b> Ingresar Clubes   |                                    |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Medio |
| <b>Puntos Estimados:</b> 3  | <b>Iteración Asignada:</b> 2       |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                    |
| <b>Descripción:</b> El administrador puede ingresar los datos para un nuevo Club o también conocida como Institución Deportiva filial a LDPT                                  |                                    |
| <b>Observaciones:</b> Los campos deben ser de carácter obligatorio. El logo del Club es importante, por su visualización e información en la página principal del aplicativo. |                                    |

*Tabla 14. Ingresar Clubes*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                    |
|---|------------------------------------|
| <b>Número: 11</b>   | <b>Usuario:</b> Administrador      |
| <b>Nombre Historia:</b> Actualizar Clubes   |                                    |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Medio |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 2       |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                    |
| <b>Descripción:</b> El administrador puede actualizar los datos de un Club o también conocida como Institución Deportiva filial a LDPT. |                                    |
| <b>Observaciones:</b> Todos los campos pueden ser editables.  |                                    |

*Tabla 15. Actualizar Clubes*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 12</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Eliminar Clubes  |                                   |
| <b>Prioridad en Negocio:</b> Alta  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador puede eliminar un Club o también conocida como Institución Deportiva filial a LDPT. |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción.                                      |                                   |

*Tabla 16. Eliminar Clubes*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 13</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar Clubes   |                                   |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador puede visualizar el detalle de un club filial de Liga Deportiva Parroquial Totoras seleccionado. |                                   |
| <b>Observaciones:</b> El listado de clubes contiene todos campos.   |                                   |

*Tabla 17. Visualizar Clubes*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 14</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Buscar Clubes  |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador puede buscar por el nombre de un club filial de Liga Deportiva Parroquial Totoras seleccionado. |                                   |
| <b>Observaciones:</b> Es necesario una búsqueda inteligente  |                                   |

*Tabla 18. Buscar Clubes*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 15</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Ingresar torneos  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador puede ingresar un torneo también conocido como campeonato o temporada. |                                   |
| <b>Observaciones:</b> Todos los campos serán obligatorios para detallar el torneo                           |                                   |

*Tabla 19. Ingresar torneos*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 16</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Actualizar torneos  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador puede actualizar los datos de un torneo existente, también conocido como campeonato o temporada. |                                   |
| <b>Observaciones:</b>   |                                   |

*Tabla 20. Actualizar torneos*

*Elaborado por: Santiago Yaguargos*



| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 17</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Eliminar torneos   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador puede eliminar un torneo existente, también conocido como campeonato o temporada. |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción.                                    |                                   |

*Tabla 21. Eliminar torneos*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 18</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar torneos   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador puede visualizar un torneo existente, también conocido como campeonato o temporada. |                                   |
| <b>Observaciones:</b>  |                                   |

*Tabla 22. Visualizar torneos*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 19</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Buscar torneos   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos                                       |                                   |
| <b>Descripción:</b> El administrador puede buscar un torneo en el listado por el año.    |                                   |
| <b>Observaciones:</b> La lista puede ser ordenada según el año que se realizó un torneo. |                                   |

*Tabla 23. Buscar torneos*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 20</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Ingresar Categorías  |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá ingresar categorías a las que pertenecerán los diferentes equipos.            |                                   |
| <b>Observaciones:</b> Todos los campos deben ser obligatorios para mantener claramente la información de las Categorías. |                                   |

*Tabla 24. Ingresar Categorías*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 21</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Actualizar Categorías   |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá actualizar datos de las determinadas categorías a las que pertenecen los diferentes equipos. |                                   |
| <b>Observaciones:</b>   |                                   |

*Tabla 25. Actualizar Categorías  
Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 22</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Eliminar Categorías   |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos                                  |                                   |
| <b>Descripción:</b> El administrador podrá eliminar Categorías existentes           |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción. |                                   |

*Tabla 26. Eliminar Categorías  
Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 23</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar Categorías                                    |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos                               |                                   |
| <b>Descripción:</b> El administrador podrá visualizar las categorías existentes. |                                   |
| <b>Observaciones:</b> Deberán mostrarse todos los campos en la lista.            |                                   |

*Tabla 27. Visualizar Categorías  
Elaborado por: Santiago Yaguargos*

| Historia de Usuario   |                                   |
|---|-----------------------------------|
| <b>Número: 24</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Ingresar Jugadores  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 3  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá ingresar datos de nuevos jugadores, estos tendrán que pertenecer a un Club o también conocida como Institución Deportiva.  |                                   |
| <b>Observaciones:</b> Todos los campos deben ser obligatorios para mantener correctamente archivada la información, y se debe generar el código de forma incremental. |                                   |

*Tabla 28. Ingresar Jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 25</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Actualizar jugadores  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá actualizar los datos de los jugadores registrados. |                                   |
| <b>Observaciones:</b> El campo código no podrá ser editado.                                   |                                   |

*Tabla 29. Actualizar jugadores*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 26</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Eliminar jugadores  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos                                  |                                   |
| <b>Descripción:</b> El administrador podrá eliminar jugadores registrados.          |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción. |                                   |

*Tabla 30. Eliminar jugadores*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 27</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar jugadores   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá visualizar los jugadores registrados en una lista.        |                                   |
| <b>Observaciones:</b> Es necesario mantener una adecuada paginación para la carga correcta de datos. |                                   |

*Tabla 31. Visualizar jugadores*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 28</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Buscar jugadores  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá realizar búsquedas de los jugadores registrados ingresando cédula o apellidos. |                                   |
| <b>Observaciones:</b> Es necesario una búsqueda inteligente   |                                   |

*Tabla 32. Buscar jugadores*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 29</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Ingresar equipos   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá ingresar nuevos equipos en su respectiva categoría y perteneciente a un Club o también llamada Institución Deportiva. |                                   |
| <b>Observaciones:</b> Es necesario llenar todos los campos requeridos.   |                                   |

*Tabla 33. Ingresar equipos*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 30</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Actualizar equipos   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá actualizar datos de los equipos registrados en su respectiva categoría y perteneciente a un Club o también llamada Institución Deportiva. |                                   |
| <b>Observaciones:</b> El campo Categoría puede ser editable ya que, en una siguiente temporada, este puede ascender o descender  |                                   |

*Tabla 34. Actualizar equipos*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 31</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Eliminar equipos  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá eliminar equipos registrados y pertenecientes a un Club o también llamada Institución Deportiva. |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción.   |                                   |

*Tabla 35. Eliminar equipos*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 32</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar equipos  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá visualizar el listado de equipos en su respectiva categoría y perteneciente a un Club o también llamada Institución Deportiva. |                                   |
| <b>Observaciones:</b> Es necesario mantener una adecuada paginación para la carga correcta de datos.  |                                   |

*Tabla 36. Visualizar equipos*

*Elaborado por: Santiago Yaguargos*



| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 33</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Buscar equipos  |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1  | <b>Iteración Asignada:</b> 2      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá realizar búsquedas de equipos por su nombre, categoría o club. |                                   |
| <b>Observaciones:</b> Es necesario una búsqueda inteligente   |                                   |

*Tabla 37. Buscar equipos*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 34</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Habilitar equipos  |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 3      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá habilitar equipos que participaran en un torneo existente.            |                                   |
| <b>Observaciones:</b> Los equipos pertenecientes a los clubes pueden ser habilitados en un y según su categoría. |                                   |

*Tabla 38. Habilitar equipos.*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 35</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Inhabilitar equipos  |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 3      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá inhabilitar a los equipos que no participaran en un determinado torneo. |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción.                                |                                   |

*Tabla 39. Inhabilitar equipos*

*Elaborado por: Santiago Yaguargos*

| Historia de Usuario  |                                   |
|--|-----------------------------------|
| <b>Número: 36</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar equipos habilitados   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 1   | <b>Iteración Asignada:</b> 3      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El usuario final podrá visualizar los equipos que participaran actualmente en un torneo. |                                   |
| <b>Observaciones:</b>  |                                   |

*Tabla 40. Visualizar equipos habilitados*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 37</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Habilitar jugadores   |                                   |
| <b>Prioridad en Negocio:</b> Media  | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2  | <b>Iteración Asignada:</b> 3      |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador podrá habilitar jugadores en los equipos que están habilitados en una determinada temporada, ingresando su cédula. |                                   |
| <b>Observaciones:</b> Los jugadores pertenecientes a los clubes pueden ser habilitados en determinados equipos y según su categoría asignada.           |                                   |

*Tabla 41. Habilitar jugadores*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 38</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Inhabilitar jugadores  |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 3      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El administrador podrá inhabilitar a los jugadores de los equipos habilitados. |                                   |
| <b>Observaciones:</b> Es necesario un mensaje en pantalla para confirmar la acción.                |                                   |

*Tabla 42. Inhabilitar jugadores*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>   |                                   |
|--|-----------------------------------|
| <b>Número: 39</b>  | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Visualizar jugadores habilitados   |                                   |
| <b>Prioridad en Negocio:</b> Media   | <b>Riesgo en Desarrollo:</b> Alto |
| <b>Puntos Estimados:</b> 2   | <b>Iteración Asignada:</b> 3      |
| <b>Programador Responsable:</b> Santiago Yaguargos   |                                   |
| <b>Descripción:</b> El usuario final podrá visualizar los jugadores habilitados que participaran en un determinado equipo según el torneo. |                                   |
| <b>Observaciones:</b>  |                                   |

*Tabla 43. Visualizar jugadores habilitados*

*Elaborado por: Santiago Yaguargos*

| <b>Historia de Usuario</b>  |                                   |
|---|-----------------------------------|
| <b>Número: 40</b>   | <b>Usuario:</b> Administrador     |
| <b>Nombre Historia:</b> Validación de formularios   |                                   |
| <b>Prioridad en Negocio:</b> Alta   | <b>Riesgo en Desarrollo:</b> Alta |
| <b>Puntos Estimados:</b> 4  | <b>Iteración Asignada:</b> 12     |
| <b>Programador Responsable:</b> Santiago Yaguargos  |                                   |
| <b>Descripción:</b> El administrador requiere que todos los formularios estén validados para la administración de datos |                                   |
| <b>Observaciones:</b> El requerimiento debe cumplir con mensajes de información.  |                                   |

*Tabla 44. Visualizar jugadores habilitados*

*Elaborado por: Santiago Yaguargos*

### **Estimación historias de usuario**

El proyecto se enfoca en una Aplicación Web Progresiva que se adapte a distintos dispositivos como son ordenadores, tabletas, smartphones, sin embargo, también puede ser utilizada únicamente como una aplicación web.

- Aplicación web, donde será encargará de manipularla el administrador con el objetivo de realizar determinados procesos de gestión de datos.
- Aplicación web progresiva, netamente enfocada al usuario final quien optara por ella debido a sus características parecidas a una aplicación móvil nativa.

Es importante mencionar que, el administrador podrá optar por la aplicación web progresiva cuando se la instale en ordenadores.

En conclusión, este tipo de aplicaciones podrá ser utilizada como si fuese una aplicación web natural o instalarla en los diferentes dispositivos que soporten PWA.

En la estimación se considero un trabajo de 3 horas por día dedicadas al desarrollo de las historias de usuario.

| Nro. | Historia de Usuario              | Tiempo estimado |                 |
|------|----------------------------------|-----------------|-----------------|
|      |                                  | Días estimados  | Horas estimadas |
| 1    | Diseño de interfaces-PWA         | 5               | 15              |
| 2    | Visualizar información de LDPT   | 2               | 6               |
| 3    | Visualizar información de Clubes | 2               | 6               |
| 4    | Descargar documentación          | 2               | 6               |
| 5    | Inicio de Sesión                 | 2               | 6               |
| 6    | Ingresar noticias                | 2               | 6               |
| 7    | Actualizar noticias              | 1               | 3               |
| 8    | Eliminar noticias                | 1               | 3               |
| 9    | Visualizar noticias              | 2               | 6               |
| 10   | Ingresar clubes                  | 3               | 9               |
| 11   | Actualizar clubes                | 2               | 6               |
| 12   | Eliminar clubes                  | 1               | 3               |
| 13   | Visualizar clubes                | 1               | 3               |
| 14   | Buscar clubes                    | 1               | 3               |
| 15   | Ingresar torneos                 | 2               | 6               |
| 16   | Actualizar torneos               | 1               | 3               |
| 17   | Eliminar torneos                 | 1               | 3               |
| 18   | Visualizar torneos               | 1               | 3               |
| 19   | Buscar torneos                   | 1               | 3               |
| 20   | Ingresar categorías              | 2               | 6               |
| 21   | Actualizar categorías            | 1               | 3               |
| 22   | Eliminar categorías              | 1               | 3               |
| 23   | Visualizar categorías            | 1               | 3               |
| 24   | Ingresar jugadores               | 3               | 9               |
| 25   | Actualizar jugadores             | 2               | 6               |
| 26   | Eliminar jugadores               | 1               | 3               |
| 27   | Visualizar jugadores             | 1               | 3               |
| 28   | Buscar jugadores                 | 1               | 3               |
| 29   | Ingresar equipos                 | 2               | 6               |
| 30   | Actualizar equipos               | 1               | 3               |
| 31   | Eliminar equipos                 | 1               | 3               |
| 32   | Visualizar equipos               | 1               | 3               |
| 33   | Buscar equipos                   | 1               | 3               |
| 34   | Habilitar equipos                | 2               | 6               |
| 35   | Inhabilitar equipos              | 2               | 6               |
| 36   | Visualizar equipos habilitados   | 1               | 3               |
| 37   | Habilitar jugadores              | 2               | 6               |

|        |                                  |    |     |
|--------|----------------------------------|----|-----|
| 38     | Inhabilitar jugadores            | 2  | 6   |
| 39     | Visualizar jugadores habilitados | 2  | 6   |
| 40     | Validación de formularios        | 4  | 12  |
| Total: |                                  | 67 | 201 |

Tabla 45. Estimación de historias de usuario  
Elaborado por: Santiago Yaguargos

- **Fase 3. Iteraciones**

### Plan de entrega

El objetivo en esta fase es diseñar la presentación de un plan de entrega del producto según las funciones específicas de la aplicación, manteniendo correctamente la comprensión durante esta fase.

| N° | Historia de Usuario              | Tiempo estimado |                | Interacción asignada |   |   | Entrega asignada |   |   |
|----|----------------------------------|-----------------|----------------|----------------------|---|---|------------------|---|---|
|    |                                  | Días estimados  | Días estimados | 1                    | 2 | 3 | 1                | 2 | 3 |
| 1  | Diseño de interfaces-PWA         | 5               | 15             | X                    |   |   | X                |   |   |
| 2  | Visualizar información de LDPT   | 2               | 6              | X                    |   |   | X                |   |   |
| 3  | Visualizar información de Clubes | 2               | 6              | X                    |   |   | X                |   |   |
| 4  | Descargar documentación          | 2               | 6              | X                    |   |   | X                |   |   |
| 5  | Inicio de Sesión                 | 2               | 6              | X                    |   |   | X                |   |   |
| 6  | Ingresar noticias                | 2               | 6              | X                    |   |   | X                |   |   |
| 7  | Actualizar noticias              | 1               | 3              | X                    |   |   | X                |   |   |
| 8  | Eliminar noticias                | 1               | 3              | X                    |   |   | X                |   |   |
| 9  | Visualizar Noticias              | 2               | 6              | X                    |   |   | X                |   |   |
| 10 | Ingresar clubes                  | 3               | 9              |                      | X |   |                  | X |   |
| 11 | Actualizar clubes                | 2               | 6              |                      | X |   |                  | X |   |
| 12 | Eliminar clubes                  | 1               | 3              |                      | X |   |                  | X |   |
| 13 | Visualizar clubes                | 1               | 3              |                      | X |   |                  | X |   |
| 14 | Buscar clubes                    | 1               | 3              |                      | X |   |                  | X |   |
| 15 | Ingresar torneos                 | 2               | 6              |                      | X |   |                  | X |   |
| 16 | Actualizar torneos               | 1               | 3              |                      | X |   |                  | X |   |
| 17 | Eliminar torneos                 | 1               | 3              |                      | X |   |                  | X |   |
| 18 | Visualizar torneos               | 1               | 3              |                      | X |   |                  | X |   |
| 19 | Buscar torneos                   | 1               | 3              |                      | X |   |                  | X |   |
| 20 | Ingresar categorías              | 2               | 6              |                      | X |   |                  | X |   |
| 21 | Actualizar categorías            | 1               | 3              |                      | X |   |                  | X |   |
| 22 | Eliminar categorías              | 1               | 3              |                      | X |   |                  | X |   |
| 23 | Visualizar categorías            | 1               | 3              |                      | X |   |                  | X |   |
| 24 | Ingresar jugadores               | 3               | 9              |                      | X |   |                  | X |   |
| 25 | Actualizar jugadores             | 2               | 6              |                      | X |   |                  | X |   |
| 26 | Eliminar jugadores               | 1               | 3              |                      | X |   |                  | X |   |
| 27 | Visualizar jugadores             | 1               | 3              |                      | X |   |                  | X |   |
| 28 | Buscar jugadores                 | 1               | 3              |                      | X |   |                  | X |   |
| 29 | Ingresar equipos                 | 2               | 6              |                      | X |   |                  | X |   |
| 30 | Actualizar equipos               | 1               | 3              |                      | X |   |                  | X |   |
| 31 | Eliminar equipos                 | 1               | 3              |                      | X |   |                  | X |   |
| 32 | Visualizar equipos               | 1               | 3              |                      | X |   |                  | X |   |
| 33 | Buscar equipos                   | 1               | 3              |                      | X |   |                  | X |   |
| 34 | Habilitar equipos                | 2               | 6              |                      |   | X |                  |   | X |
| 35 | Inhabilitar equipos              | 2               | 6              |                      |   | X |                  |   | X |
| 36 | Visualizar equipos habilitados   | 1               | 3              |                      |   | X |                  |   | X |
| 37 | Habilitar jugadores              | 2               | 6              |                      |   | X |                  |   | X |
| 38 | Inhabilitar jugadores            | 2               | 6              |                      |   | X |                  |   | X |
| 39 | Visualizar jugadores habilitados | 2               | 6              |                      |   | X |                  |   | X |
| 40 | Validación de formularios        | 4               | 12             |                      |   | X |                  |   | X |

Tabla 46. Plan de Entrega  
Elaborado por: Santiago Yaguargos

## Iteraciones

El objetivo en esta fase es representar y estimar el tiempo de desarrollo en cada una de las historias de usuario. Los tiempos se plantean luego de interactuar entre el usuario final y el desarrollador. Se definen en días de 3 horas de trabajo.

| Iteración | Nº                  | Historia de Usuario              | Prioridad | Actividad | Riesgo | Versión  | Estado de Desarrollo | Pruebas  |
|-----------|---------------------|----------------------------------|-----------|-----------|--------|----------|----------------------|----------|
| Primera   | 1                   | Diseño de interfaces-PWA         | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 2                   | Visualizar información de LDPT   | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 3                   | Visualizar información de Clubes | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 4                   | Descargar documentación          | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 5                   | Inicio de Sesión                 | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 6                   | Ingresar noticias                | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 7                   | Actualizar noticias              | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 8                   | Eliminar noticias                | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 9                   | Visualizar Noticias              | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 10                  | Ingresar clubes                  | 1         | Nueva     | Medio  | 1        | Completo             | Aprobado |
| Segunda   | 11                  | Actualizar clubes                | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 12                  | Eliminar clubes                  | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 13                  | Visualizar clubes                | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 14                  | Buscar clubes                    | 1         | Nueva     | Medio  | 1        | Completo             | Aprobado |
|           | 15                  | Ingresar torneos                 | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 16                  | Actualizar torneos               | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 17                  | Eliminar torneos                 | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 18                  | Visualizar torneos               | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 19                  | Buscar torneos                   | 1         | Nueva     | Medio  | 1        | Completo             | Aprobado |
|           | 20                  | Ingresar categorías              | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 21                  | Actualizar categorías            | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 22                  | Eliminar categorías              | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 23                  | Visualizar categorías            | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 24                  | Ingresar jugadores               | 1         | Nueva     | Medio  | 1        | Completo             | Aprobado |
|           | 25                  | Actualizar jugadores             | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 26                  | Eliminar jugadores               | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 27                  | Visualizar jugadores             | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 28                  | Buscar jugadores                 | 1         | Nueva     | Medio  | 1        | Completo             | Aprobado |
|           | 29                  | Ingresar equipos                 | 1         | Nueva     | Bajo   | 1        | Completo             | Aprobado |
|           | 30                  | Actualizar equipos               | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
| 31        | Eliminar equipos    | 1                                | Nueva     | Alto      | 1      | Completo | Aprobado             |          |
| 32        | Visualizar equipos  | 1                                | Nueva     | Alto      | 1      | Completo | Aprobado             |          |
| 33        | Buscar equipos      | 1                                | Nueva     | Medio     | 1      | Completo | Aprobado             |          |
| 34        | Habilitar equipos   | 1                                | Nueva     | Bajo      | 1      | Completo | Aprobado             |          |
| 35        | Inhabilitar equipos | 1                                | Nueva     | Alto      | 1      | Completo | Aprobado             |          |
| Tercera   | 36                  | Visualizar equipos habilitados   | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 37                  | Habilitar jugadores              | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 38                  | Inhabilitar jugadores            | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 39                  | Visualizar jugadores habilitados | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |
|           | 40                  | Validación de formularios        | 1         | Nueva     | Alto   | 1        | Completo             | Aprobado |

Tabla 47. Histórico de versiones en historias de usuario  
Elaborado por: Santiago Yaguargos

## Equipos y roles

| Miembro                                   | Roles                                 | Metodología              |
|---|---------------------------------------|--------------------------|
| Santiago Yaguargos                        | Programador                           | Extreme Programming (XP) |
| Encuestados                               | Cliente                               |                          |
| Santiago Yaguargos<br>Ing. Edison Álvarez | Encargado de pruebas<br>(tester)      |                          |
| Ing. Edison Álvarez                       | Entrenador (coach)                    |                          |
| Santiago Yaguargos<br>Ing. Edison Álvarez | Encargado de<br>seguimiento (tracker) |                          |
| Santiago Yaguargos                        | Consultor                             |                          |

Tabla 48. Equipos y roles  
Elaborado por: Santiago Yaguargos

- **Fase 4. Producción**

### Modelado de la Base de Datos (Procesos de gestión)

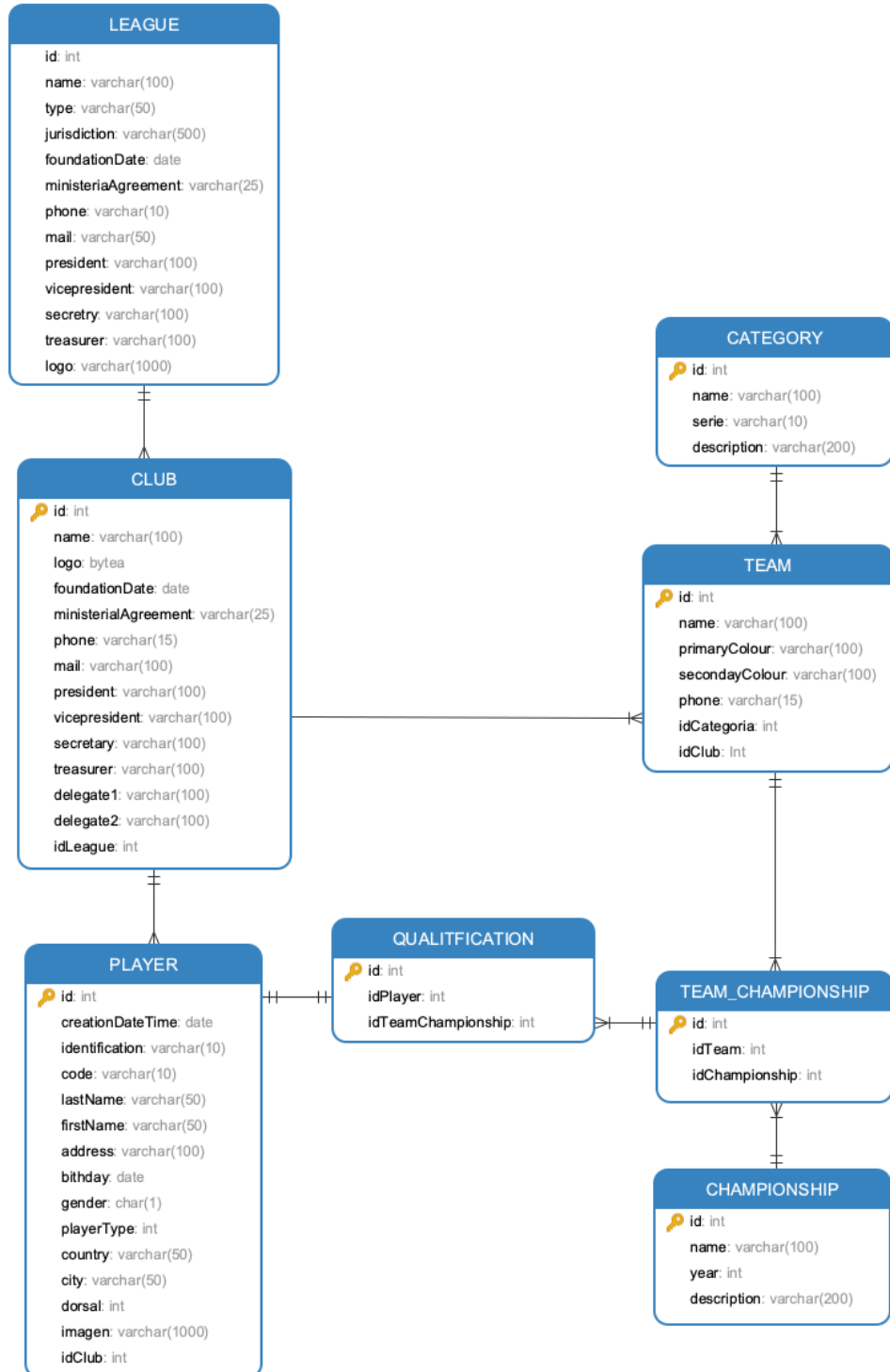


Figura 23. Modelo Base de Datos relacional

Elaborado por: Santiago Yaguargos



## Modelado de la Base de Datos (Procesos de información)



Figura 24. Modelo Base de Datos no relacional

Elaborado por: Santiago Yaguargos

## Diccionario de datos

Nombre de la tabla: league

Descripción: Almacena la información de Liga Deportiva Parroquial Totoras

| Campo                | Tipo   | Descripción                                   |
|----------------------|--------|---|
| id                   | Int    | Clave primaria de la liga                     |
| name                 | String | Nombre completo de la liga                    |
| type                 | String | Tipo de liga (barrial, parroquial, cantonal)  |
| jurisdiction         | String | Nombre de la jurisdicción a la que pertenece. |
| foundationDate       | Int    | Fecha de fundación de la institución.         |
| ministerialAgreement | String | Código del Acuerdo Ministerial                |
| president            | String | Nombre completo del presidente                |
| vicepresident        | String | Nombre completo del vicepresidente            |
| secretary            | String | Nombre completo del secretario o secretaria   |
| treasurer            | String | Nombre completo del tesorero o tesorera       |
| logo                 | Byte   | Logo representativo de la institución         |

Tabla 49. Estructura tabla league

Elaborado por: Santiago Yaguargos

Nombre de la tabla: club

Descripción: Almacena la información de los clubes

| Campo                | Tipo   | Descripción                                 |
|----------------------|--------|---|
| id                   | Int    | Clave primaria del club deportivo           |
| name                 | String | Nombre completo del club deportivo          |
| logo                 | Byte   | Logo institucional                          |
| foundationDate       | String | Fecha de fundación del club deportivo       |
| ministerialAgreement | String | Código del Acuerdo Ministerial              |
| phone                | String | Número telefónico del club deportivo        |
| mail                 | String | Correo del club o representante del club    |
| president            | String | Nombre completo del presidente              |
| vicepresident        | String | Nombre completo del vicepresidente          |
| secretary            | String | Nombre completo del secretario o secretaria |
| treasurer            | String | Nombre completo del tesorero o tesorera     |
| delegate1            | String | Nombre completo del delegado principal      |

|           |        |   |
|-----------|--------|---|
| delegate2 | String | Nombre completo del delegado secundario |
| idLiga    | Int    | Clave foránea del código de liga        |

*Tabla 50. Estructura tabla club*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: category

Descripción: Almacena la información de las categorías

| <b>Campo</b> | <b>Tipo</b> | <b>Descripción</b>                      |
|--------------|-------------|---|
| id           | Int         | Clave primaria de la categoría          |
| name         | String      | Nombre de la categoría                  |
| serie        | String      | Nombre de la serie                      |
| description  | String      | Descripción referencial de la categoría |

*Tabla 51. Estructura tabla category*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: team

Descripción: Almacena la información de los equipos

| <b>Campo</b>    | <b>Tipo</b> | <b>Descripción</b>                     |
|-----------------|-------------|--|
| id              | Int         | Clave primaria del equipo              |
| name            | String      | Nombre completo del club               |
| primaryColour   | String      | Nombre del color principal del equipo  |
| secondaryColour | String      | Nombre del color secundario del equipo |
| phone           | String      | Número telefónico de contacto          |
| idCategory      | Int         | Clave foránea del código de categoría  |
| idClub          | Int         | Clave foránea del código del club      |

*Tabla 52. Estructura tabla team*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: player

Descripción: Almacena la información de los jugadores

| <b>Campo</b>   | <b>Tipo</b> | <b>Descripción</b>                               |
|----------------|-------------|--|
| id             | Int         | Clave primaria del jugador                       |
| identification | String      | Cedula de ciudadanía del jugador                 |
| code           | String      | Código generado para el jugador                  |
| firstName      | String      | Nombres del jugador                              |
| lastName       | String      | Apellidos del jugador                            |
| address        | String      | Dirección del jugador                            |
| birthday       | String      | Fecha de nacimiento del jugador                  |
| gender         | String      | Género: M o F del jugador                        |
| typePlayer     | Int         | Tipo de jugador: Nacional, Foráneo o Comunitario |
| country        | String      | País de origen del jugador                       |
| city           | String      | Ciudad de origen del jugador                     |
| image          | String      | Imagen digital del jugador                       |

|        |     |  |
|--------|-----|--|
| idClub | Int | Clave foránea del código del club al que pertenece |
|--------|-----|--|

*Tabla 53. Estructura tabla player*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: championship

Descripción: Almacena la información de los torneos

| <b>Campo</b> | <b>Tipo</b> | <b>Descripción</b>                 |
|--------------|-------------|------------------------------------|
| Id           | Int         | Clave primaria del torneo          |
| name         | String      | Nombre del torneo                  |
| year         | Int         | Año de realización del torneo      |
| description  | String      | Descripción referencial del torneo |

*Tabla 54. Estructura tabla championship*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: teamChampionship

Descripción: Almacena la información de los equipos inscritos en un torneo.

| <b>Campo</b>   | <b>Tipo</b> | <b>Descripción</b>        |
|----------------|-------------|---------------------------|
| id             | Int         | Clave primaria del torneo |
| idTeam         | Int         | Clave foránea del equipo  |
| idChampionship | Int         | Clave foránea del torneo  |

*Tabla 55. Estructura tabla teamChampionship*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: qualification

Descripción: Almacena la información de las habilitaciones de jugadores en equipos inscritos en un torneo.

| <b>Campo</b>       | <b>Tipo</b> | <b>Descripción</b>                               |
|--------------------|-------------|--|
| id                 | Int         | Clave primaria del torneo                        |
| idTeamChampionship | Int         | Clave foránea del equipo habilitado en un torneo |
| idPlayer           | Int         | Clave foránea del id del jugador                 |

*Tabla 56. Estructura tabla qualification*

*Elaborado por: Santiago Yaguargos*

Nombre de la tabla: news

Descripción: Almacena las noticias acerca de Liga Deportiva Parroquial Totoras.

| <b>Campo</b> | <b>Tipo</b> | <b>Descripción</b>                  |
|--------------|-------------|-------------------------------------|
| id           | Int         | Clave primaria del torneo           |
| tittle       | String      | Título de la noticia                |
| date         | Date        | Fecha de la noticias                |
| description  | String      | Descripción detallada de la noticia |

*Tabla 57. Estructura tabla news*

*Elaborado por: Santiago Yaguargos*

## **Instalación y configuración de las herramientas de desarrollo**

Para iniciar con el desarrollo de la aplicación, se requieren instalar las herramientas que servirán para construir la PWA. La configuración adecuada del framework es primordial, debido a los complementos que Vue brinda al desarrollador a través de el entorno visual para creación de proyectos.

- **Instalación de NodeJS**

Inicialmente se procede a la instalación del entorno de ejecución NodeJS, obteniendo su instalador de la pagina oficial <https://nodejs.org/es/> donde automáticamente detecta el sistema operativo y al ejecutarlo se establece una instalación global.

- **Instalación de VueJS y creación del proyecto**

Para el desarrollo frontend de la aplicación se establece el framework VueJS, mismo que requiere un editor de código y para en el presente proyecto se accede a la herramienta Visual Studio Code.

En la terminal de Visual Studio Code se procede con la instalación y configuración de Vue CLI 3, utilizando *npm* (sistema gestor de paquetes por defecto de NodeJS), a través del siguiente comando:

```
~npm install -g @vue/cli
```

Se instalará la ultima versión disponible, y de manera global al utilizar el comando *-g*. Se verifica la versión que se instalo utilizando el comando:

```
~vue --version
```

- **Configuración y ejecución de un proyecto con VueJS**

Para crear el proyecto con Vue CLI, se procede desde la interfaz grafica de usuario (GUI) con el siguiente comando:

~vue ui

Se abrirá el *Administrador de Proyectos de Vue* en el navegador predeterminado.

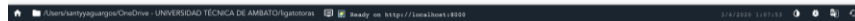


Figura 25. GUI de Vue

Elaborador por: Santiago Yaguargos

Para crear un proyecto nuevo mediante el GUI de Vue, seleccionar la pestaña *Crear* y establecer los detalles indicados en la Fig. 26 y seleccionar *Siguiente*.

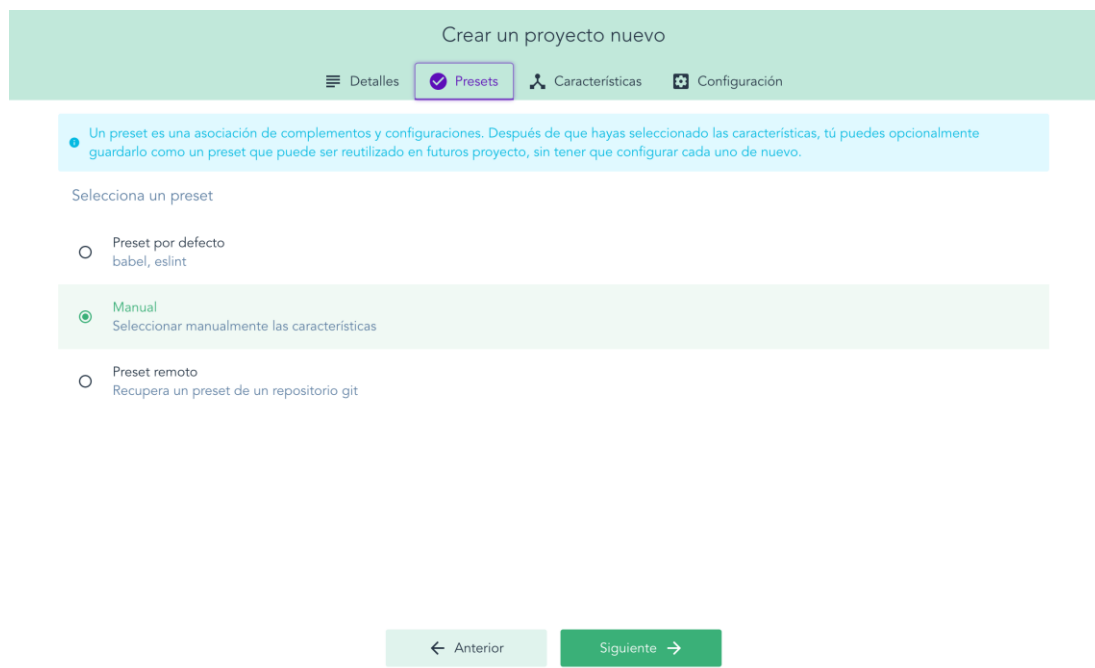


Figura 26. Creación de un proyecto de Vue

Elaborador por: Santiago Yaguargos

Cuando se trata de la creación de un proyecto por primera vez a través de la GUI, seleccionar un *preset* de forma manual.

Un preset es un conjunto de configuraciones establecidas por el usuario en un proyecto, que se puede guardar para reutilizado en futuros proyectos, evitando la configuración de uno nuevo.



*Figura 27. Presets en la creación de un proyecto en Vue  
Elaborador por: Santiago Yaguargos*

Las características que se habilitaran en el presente proyecto son las siguientes:

**Babel:** transpilador moderno de JavaScript.

**Progressive Web App (PWA) Support:** soporte para construir una Aplicación Web Progresiva con Vue.

**Router:** genera un archivo de rutas, para estructurar la app con paginas dinámicas.

**Vuex:** maneja el estado de la app con una tienda centralizada.

Finalmente, se crea el proyecto, sin antes, la opción de guardar como un nuevo Preset.



Figura 28. Panel del proyecto creado en Vue  
Elaborador por: Santiago Yaguargos

Mediante la GUI para administrar el nuevo proyecto creado, en el ítem de *Tareas* y seguidamente en el ítem *serve*, se ejecuta y compila el proyecto, que se lo puede visualizar inmediatamente en el navegador como muestra la Fig. 30.

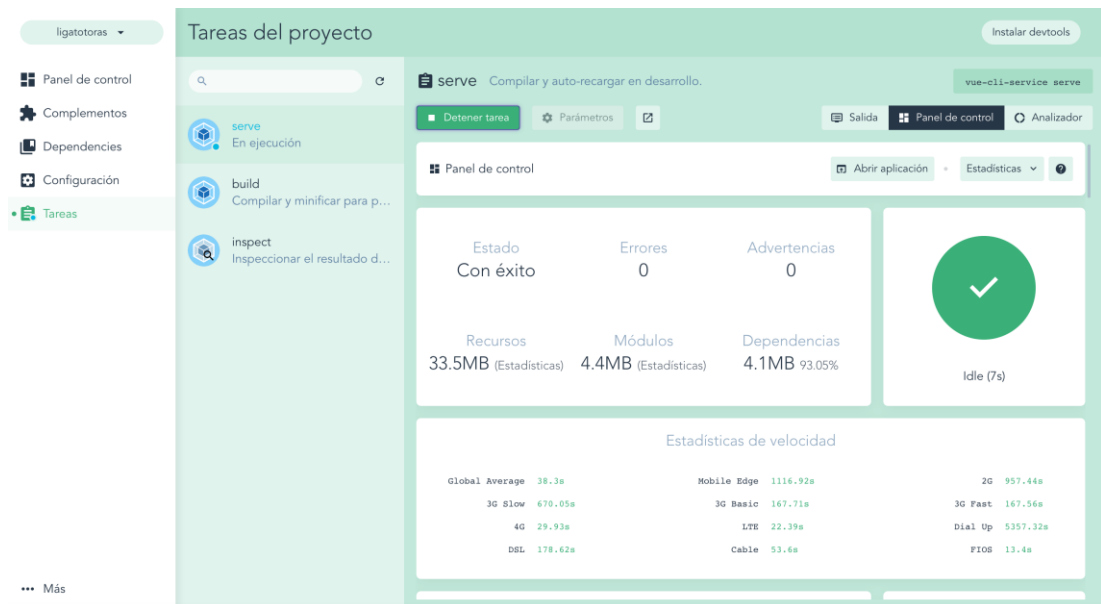


Figura 29. Tareas del proyecto creado en Vue  
Elaborador por: Santiago Yaguargos

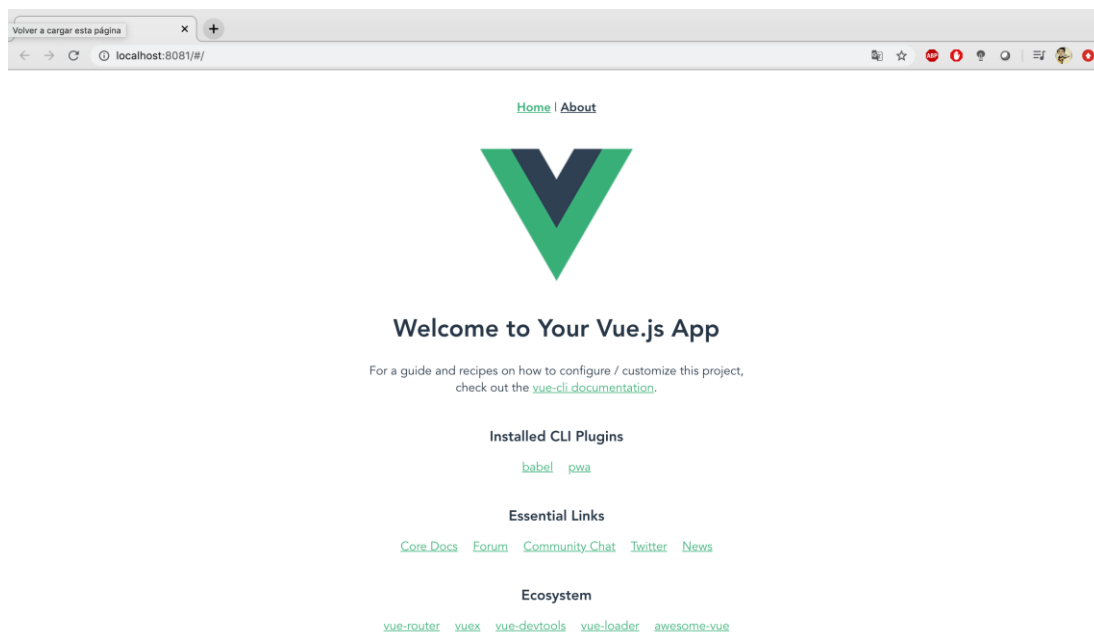


Figura 30. Pantalla de inicio de la aplicación creada en Vue

Elaborador por: Santiago Yaguargos

Para iniciar con el desarrollo de la aplicación, una vez abierto el proyecto desde el editor de código, se muestran las características habilitadas en módulos requeridos por el desarrollador.

La estructura de los archivos creados en el proyecto de Vue se encuentran dentro de carpetas, distribuidas de la siguiente manera:

**node\_modules:** contiene las dependencias instaladas y bibliotecas descargadas con el gestor de paquetes *npm*.

**public:** contiene dos archivos importantes: *favicon.ico* y *index.html*, además otra carpeta *img/icons*, donde se guardan las imágenes e iconos de forma pública.

**src:** contiene los archivos fuentes de la página web distribuidos en carpetas, es decir donde se va a desarrollar el frontend, además contiene los archivos que la hacen funcionar. Las carpetas de los archivos fuente son:

- **assets:** aquí se guardan imágenes, fuentes y archivos que se los llama en las vistas.



- components: donde se crean todos los componentes web, pueden ser separados por carpetas, para una adecuada organización del proyecto.
- views: contiene todas las vistas que se crean a nivel web y lógicamente existirá una ruta por cada componente vista.

También se generan dentro de la carpeta src los siguientes archivos:

- App.vue es el componente principal o componente de archivo único.
- router.js, donde se codifican las rutas de la pagina.
- store.js, donde se maneja la tienda a través de Vuex.
- registerServiceWorker.js, donde se registra el archivo Service Worker.

En la versión Vue CLI 4, los archivos router y store toman el nombre de index.js y sus carpetas contenedoras los nombres anteriores del archivo.

A futuro se crearán otros archivos necesarios para el desarrollo de la aplicación.

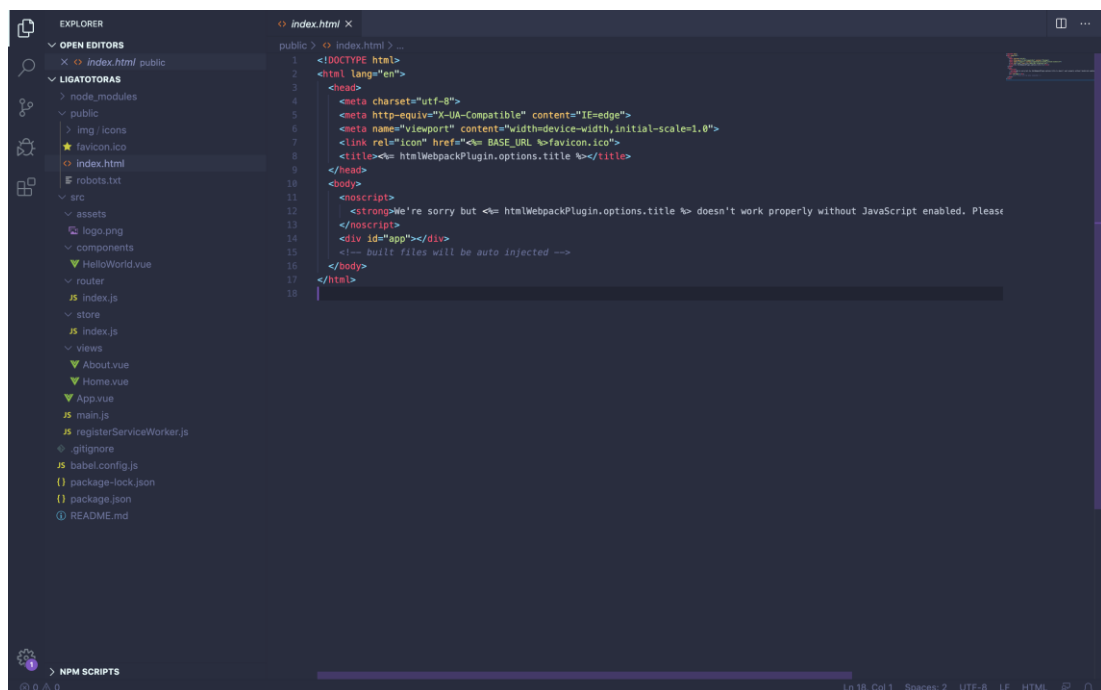


Figura 31. Estructura del proyecto con Vue

Elaborador por: Santiago Yaguargos

- **Instalación y configuración de Vuetify**

Vuetify es una librería que facilita en diseño de la aplicación y se ha establecido la herramienta, por su adaptación con Vue al agregar Material Design y una rica documentación.

Para instalar Vuetify en el proyecto creado, se accede al GUI de Vue, y en el menú, se selecciona el ítem *Complementos* y en efecto, se visualiza una pantalla con todos los complementos instalados en el proyecto, como muestra la Fig. 32.

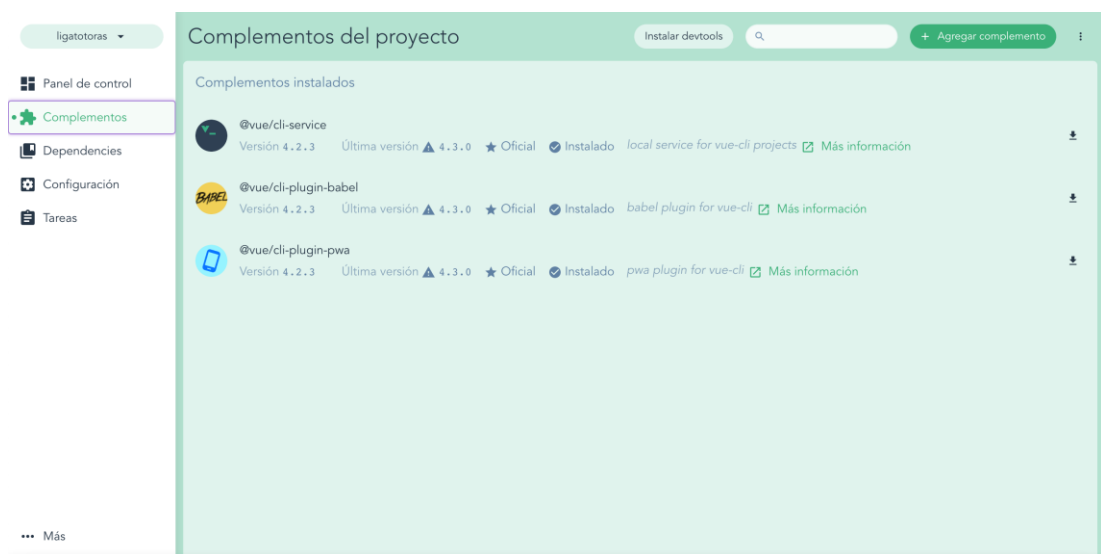


Figura 32. Complementos del proyecto en Vue

Elaborador por: Santiago Yaguargos

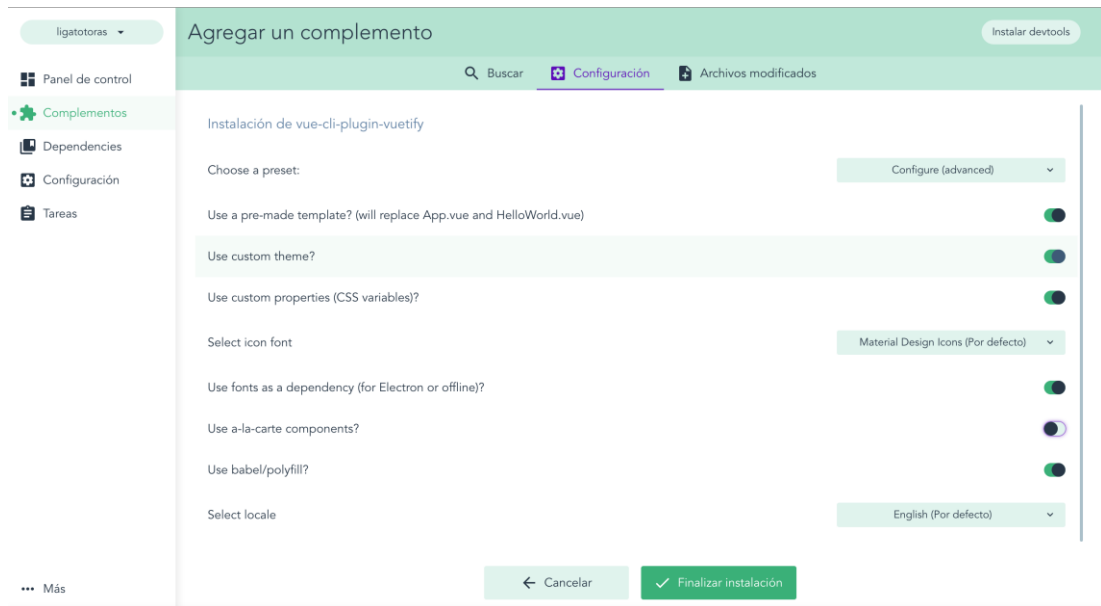
En el botón *Agregar complemento*, se procede a buscar *vue-cli-plugin-vuetify*, y se filtra con la última versión disponible. Finalmente se continua con la instalación.



Figura 33. Vuetify Framework Plugin para Vue

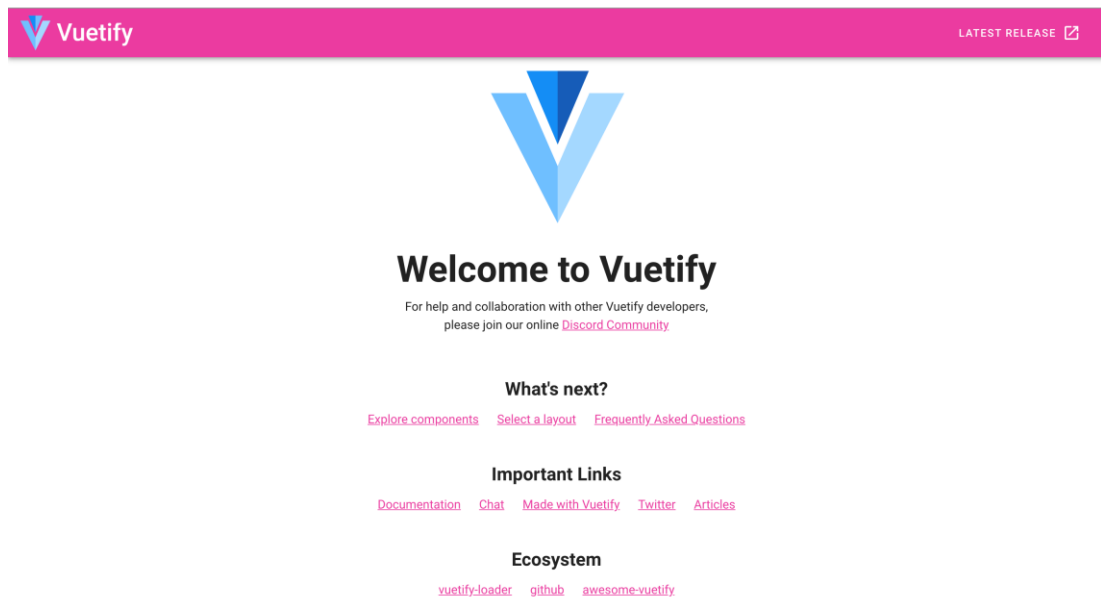
Elaborador por: Santiago Yaguargos

La configuración avanzada y requerida en base al proyecto, se muestra en la Fig. 34.



*Figura 34. Instalación de Vuetify Framework Plugin para Vue  
Elaborador por: Santiago Yaguargos*

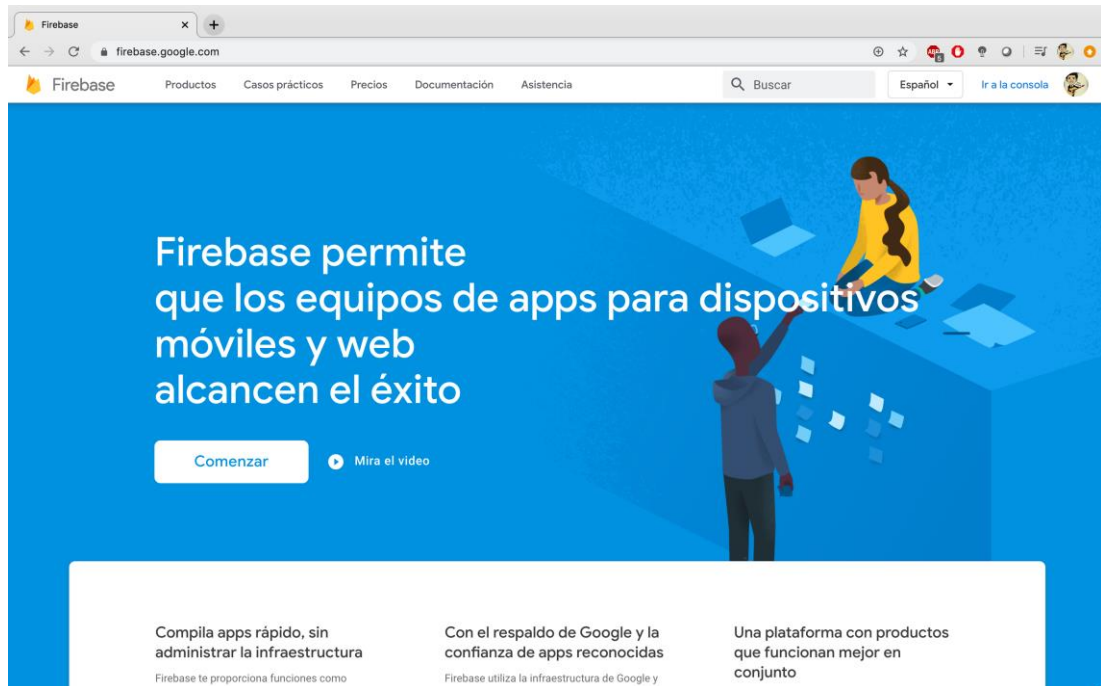
Una vez finalizada la instalación del complemento, el proyecto cambia su apariencia.



*Figura 35. Pantalla de inicio de la aplicación con Vuetify  
Elaborador por: Santiago Yaguargos*

- **Creación y configuración del proyecto con Firebase**

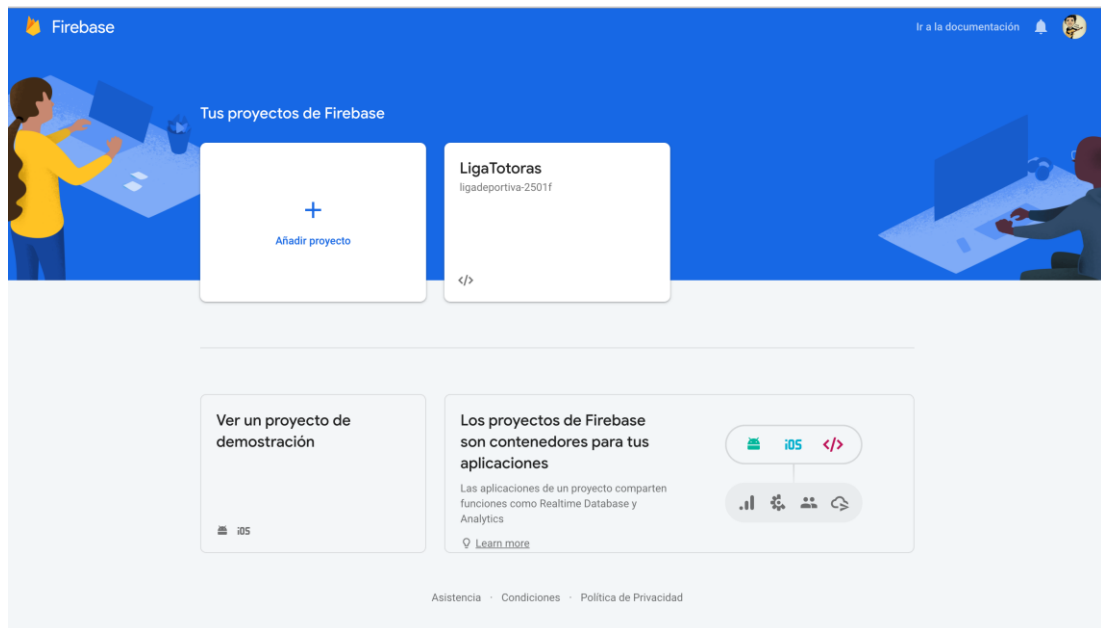
Firebase, ubicado en <https://firebase.google.com/>, requiere que el desarrollador tenga una cuenta en Google para acceder al servicio y crear un proyecto desde la consola.



*Figura 36. Página inicial de Firebase*

*Elaborador por: Santiago Yaguargos*

En la parte superior derecha de la pagina principal de Firebase, seleccionar en enlace llamado *Ir a la consola*, seguidamente seleccionar *Añadir Proyecto*, como indica la Fig.37.



*Figura 37. Crear un proyecto con Firebase*

*Elaborador por: Santiago Yaguargos*

Firestore facilita al desarrollador crear el proyecto mediante dos pasos simples:

- Paso 1: Nombre del proyecto.

El nombre del proyecto es **Liga Totoras** y su id configurado es **ligadeportivatotoras**.



Figura 38. Nombre del proyecto con Firestore

Elaborador por: Santiago Yaguargos

- Paso 2: Google Analytics en el proyecto de Firestore.

Se desactiva la opción de Habilitar Google Analytics (se puede agregar en un futuro).



Figura 39. Google Analytics en el proyecto a crear de Firestore

Elaborador por: Santiago Yaguargos

Se crea el proyecto con el nombre y listo para utilizar servicios de Firebase.

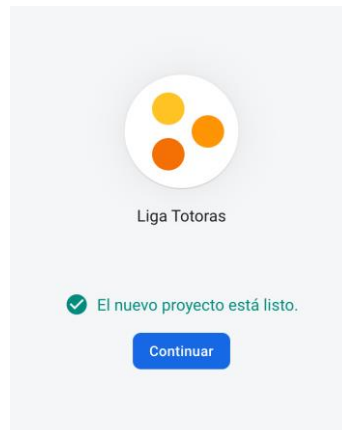


Figura 40. Mensaje exitoso del Proyecto de Firebase creado

Elaborador por: Santiago Yaguargos

Para configurar y agregar una aplicación al proyecto Liga Totoras de Firebase se procede a establecer algunas opciones. En la pantalla inicial del proyecto, seleccionar *Configuración del proyecto*, como indica la Fig. 41.

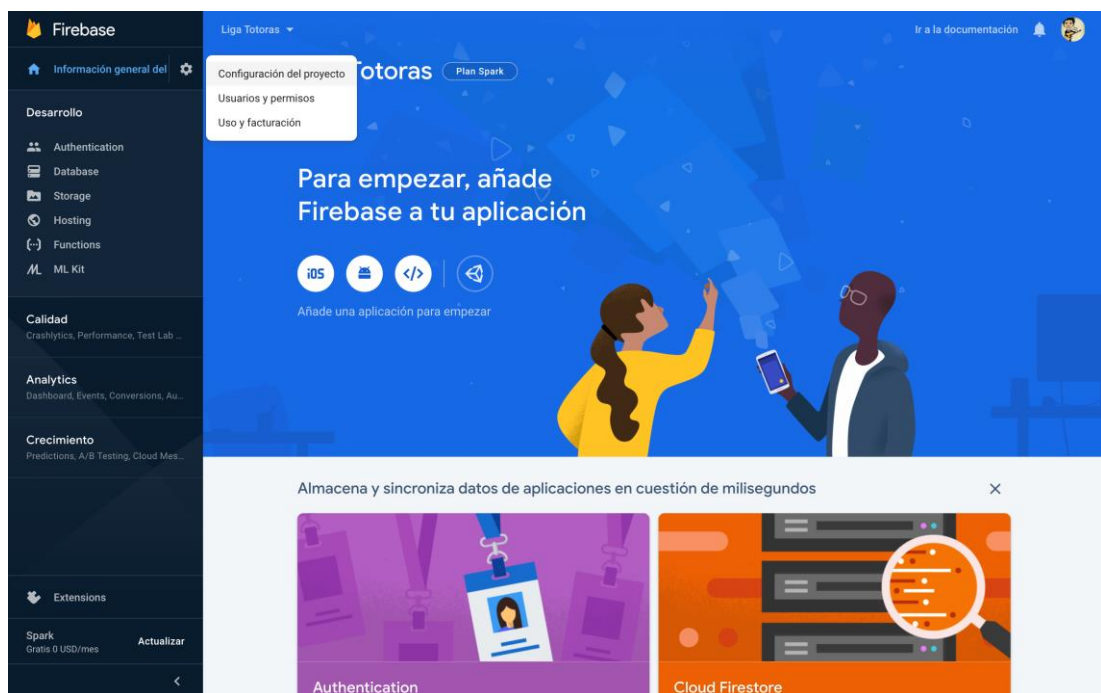
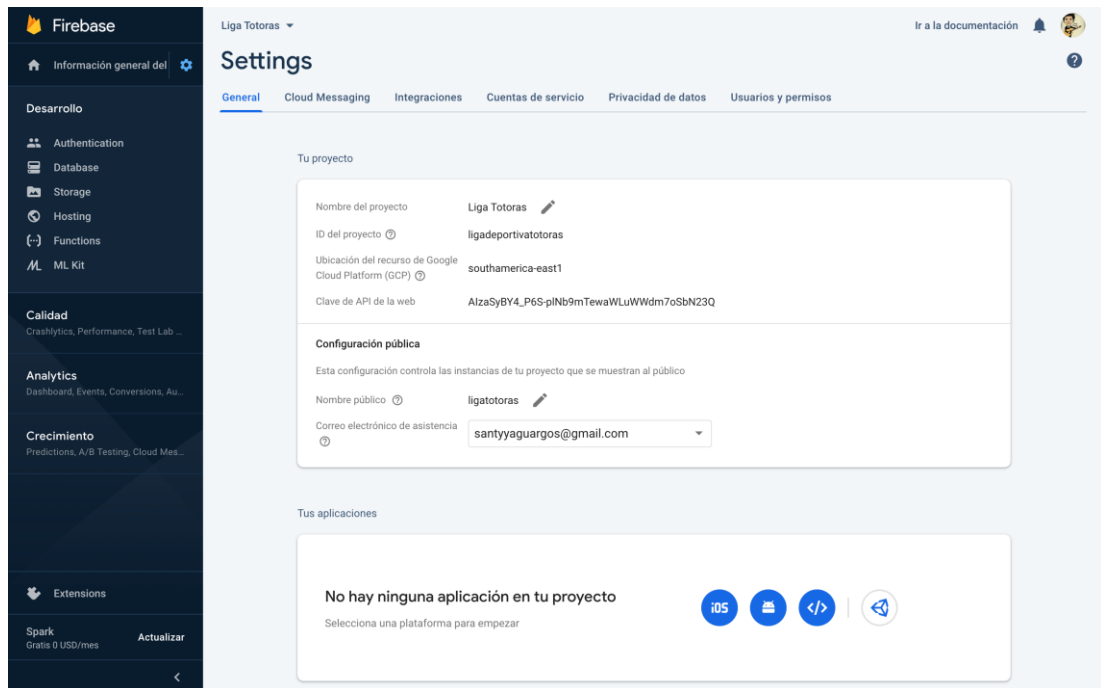


Figura 41. Ir a Configuración del Proyecto

Elaborador por: Santiago Yaguargos

En *Configuraciones*, se establece *southamerica-east1*, como ubicación del recurso de Google Cloud Platform y finalmente se agrega un correo electrónico de asistencia.



*Figura 42. Configuración del Proyecto de Firebase Liga Totoras  
Elaborador por: Santiago Yaguargos*

- **Instalar y añadir Firebase al proyecto creado con Vue**

Dado que Firebase puede ser añadido en diferentes tipos de aplicaciones y en el actual proyecto específicamente web, se procede de la siguiente manera:

En la pagina principal del proyecto de Firebase denominado *Liga Totoras* como indica la Fig.43, se encuentra el acceso a la configuración para añadir la aplicación de Vue.

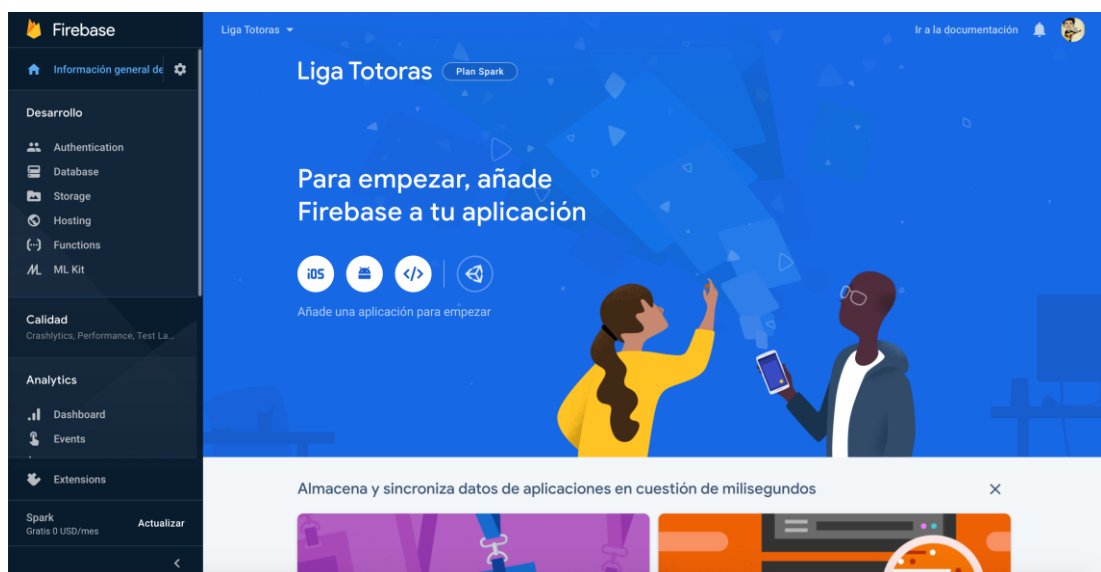


Figura 43. Página principal del Proyecto de Firebase Liga Totoras

Elaborador por: Santiago Yaguargos

Seleccionar el ícono para añadir Firebase a una aplicación web y configurar de la siguiente manera:

1. Registrar la aplicación con un apodo (la configuración de hosting se la realizará en un futuro procedimiento).
2. Añadir SDK de Firebase (no necesario ya que se instalará la CLI de Firebase)
3. Instalar CLI de Firebase

Abrir una terminal en Visual Studio Code desde el proyecto de Vue ya agregado anteriormente y ejecutar el siguiente comando para instalar el CLI de Firebase en el proyecto:

***~npm install firebase --save***

4. Desplegar en Firebase Hosting (la configuración se realizará mas adelante)

Una vez finalizada la instalación, se requiere añadir la configuración generada desde la consola de Firebase como indica la Fig.44, en el archivo *main.js* ubicado en el src del proyecto de Vue. La configuración generada es únicamente para el actual proyecto:

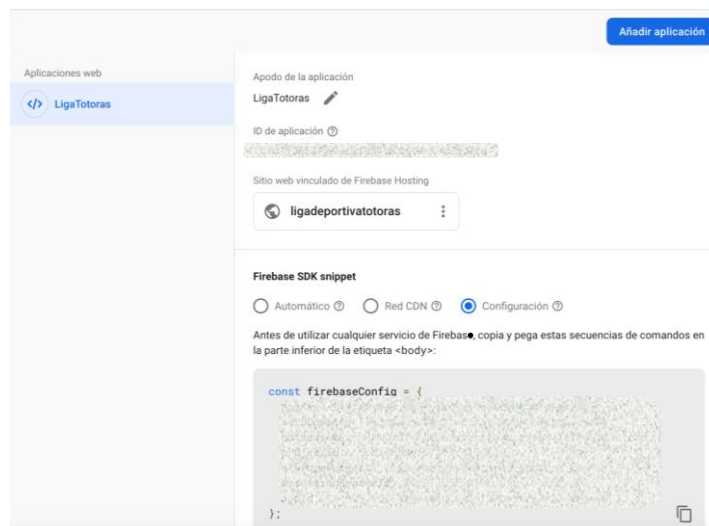


Figura 44. Configuración de Firebase para la aplicación web Liga Totoras

Elaborador por: Santiago Yaguargos



## Diseño de interfaces

Para el diseño de interfaces, Liga Deportiva Parroquial Totoras pretende mantener un estándar parecido, en base a la página web de LigaPro Ecuador que se concentra en el nivel informativo en tiempo real, acerca de la comunidad deportiva a nivel nacional, con características intuitivas.



Figura 45. Página de inicio pagina web LigaPro

Fuente: [26]

## Desarrollo de Software Frontend y Backend

- Views

Las vistas en el proyecto de Vue se van creando de acuerdo con las pantallas requeridas en base a las historias de usuario.

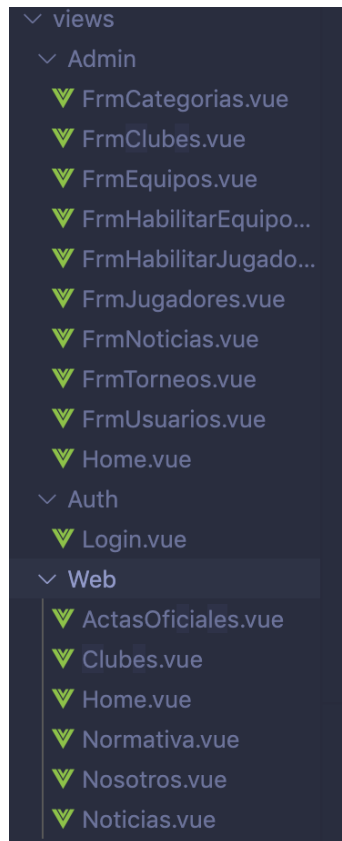


Figura 46. Archivos views o vistas en Vue

Elaborador por: Santiago Yaguargos

- **Archivo router.js**

Los archivos views o vistas de Vue son declarados en el archivo router.js, que permite generar enlaces. La estructura básica de cada ruta contiene el path, el nombre y la ruta de la vista. Se crean las rutas de acuerdo a las vistas existentes.

```
routes: [  
  //INICIO  
  {  
    path: '/',  
    name: 'home',  
    component: () => import('./views/Web/Home.vue'),  
  },  
  //NOTICIAS  
  {  
    path: '/noticias',  
    name: 'noticias',  
    component: () => import('./views/Web/Noticias.vue')  
  },  
]
```

```

//NORMATIVA
{
  path: '/normativa',
  name: 'normativa',
  component: () => import('./views/Web/Normativa.vue')
},
//ACTAS OFICIALES
{
  path: '/actas',
  name: 'actas',
  component: () => import('./views/Web/ActasOficiales.vue')
},
//NOSOTROS
{
  path: '/nosotros',
  name: 'nosotros',
  component: () => import('./views/Web/Nosotros.vue')
},
{
  path: '/clubes/:id',
  name: 'clubes',
  component: () => import('./views/Web/Clubes.vue')
},

//AUTENTICACION
{
  path: '/login',
  name: 'login',
  component: () => import('./views/Auth/Login.vue'),
},

//FORMULARIOS ADMINISTRADOR
{
  path: '/admin',
  name: 'admin',
  component: () => import('./views/Admin/Home.vue'),
},
{
  path: '/admin_user',
  name: 'adminUser',
  component: () => import('./views/Admin/FrmUsuarios.vue'),
},

//FORMULARIOS CRUD ADMINISTRADOR
//CLUBES
{
  path: '/admin_club',
  name: 'adminClub',

```

```

    component: () => import('./views/Admin/FrmClubes.vue'),
  },
  //JUGADORES
  {
    path: '/admin_player',
    name: 'adminPlayer',
    component: () => import('./views/Admin/FrmJugadores.vue'),
  },
  {
    path: '/admin_team',
    name: 'adminTeam',
    component: () => import('./views/Admin/FrmEquipos.vue'),
  },
  //TORNEOS
  {
    path: '/admin_championship',
    name: 'adminChampionship',
    component: () => import('./views/Admin/FrmTorneos.vue'),
  },
  //CATEGORIAS
  {
    path: '/admin_category',
    name: 'adminCategory',
    component: () => import('./views/Admin/FrmCategorias.vue'),
  },
  // NOTICIAS
  {
    path: '/admin_news',
    name: 'adminNews',
    component: () => import('./views/Admin/FrmNoticias.vue'),
  },
  // HABILITAR EQUIPOS
  {
    path: '/admin_team_qualification',
    name: 'adminTeamQualifications',
    component: () => import('./views/Admin/FrmHabilitarEquipos.vue'),
  },
  // HABILITAR JUGADORES
  {
    path: '/admin_player_qualification',
    name: 'adminPlayerQualification',
    component: () => import('./views/Admin/FrmHabilitarJugadores.vue'),
  },
]

```

- **Archivo App.vue**

Un archivo .vue se compone de un template que agrupa el lenguaje de etiquetas y un script para las funcionalidades.

El archivo App.vue es el archivo inicial del proyecto y componentes alojados allí, se reflejan en toda la aplicación en forma de contenedor. Vue permite crear componentes para tener una estructura organizada del proyecto y estos puedan ser reutilizados durante el desarrollo de la aplicación.

Finalmente es importante la etiqueta router-view, que reconoce el archivo router.js que contiene las rutas configuradas para las páginas de la aplicación.

```
<template>
  <v-app>
    <ContainerWeb></ContainerWeb>
    <v-content>
      <router-view />
    </v-content>
    <Footer></Footer>
  </v-app>
</template>

<script>
import Footer from "./components/Footer.vue";
import ContainerWeb from "./components/ContainerWeb.vue";
export default {
  name: "App",
  components: {
    Footer,
    ContainerWeb
  }
};
</script>
```

Dentro de las funcionalidades, se importan los componentes creados desde su ruta de origen y se los declara en el modulo *components* con la siguiente estructura:

```
import NombreComponente from "@rutaArchivoComponente"  
export default {  
  components: {  
    NombreComponente  
  }  
}
```

- **Componente Footer**

```
<v-footer dark padless>  
  <v-card flat tile class="indigo darken-4 white--text text-center">  
    <v-card-text>  
      <v-btn v-for="item in socialNetworks" :key="item.icon" :href="item.href"  
icon>  
        <v-icon>{{ item.icon }}</v-icon>  
      </v-btn>  
    </v-card-text>  
    <v-card-text class="white--text pt-0">  
      Liga Deportiva Parroquial Totoras ha crecido notablemente en su auge  
deportivo, uniendo masas de todas nuestras localidades, para con ello, expresar la  
actividad física y recreacional. Trabajamos a diario para dejar en grande nuestra  
institución, se vienen nuevos proyectos rumbo al {{ new Date().getFullYear() }}.  
    </v-card-text>  
    <v-divider></v-divider>  
    <v-card-text class="white--text">  
      <strong>{{ new Date().getFullYear() }} — @LigaTotoras All Right  
Reserved</strong>  
    </v-card-text>  
  </v-card>  
</v-footer>
```

Mediante la directiva *v-for* se recorre el arreglo alojada en el *data* llamado *socialNetworks* que contiene el ícono y la url de referencia.

socialNetworks: [

```
{ icon: "mdi-facebook", href: "https://www.facebook.com/LigaTotoras/" },
```

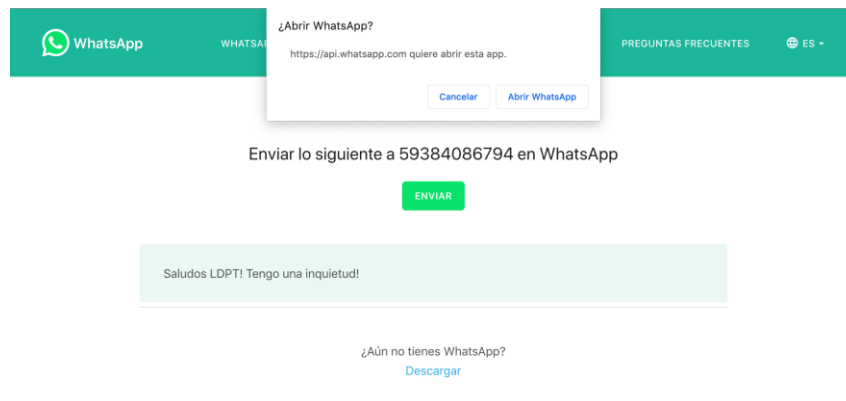
```
{ icon: "mdi-instagram", href: "https://www.instagram.com/ligatotoras/" },
```

```
{ icon: "mdi-whatsapp", href:
```

```
"https://api.whatsapp.com/send?phone=59384086794&text=Saludos%20LDPT!%20Tengo%20una%20inquietud!" }
```

],

Para la referencia de Whatsapp se utiliza el API de Whatsapp Bussines, en la Fig. 47.



*Figura 47. API de Whatsapp empresarial de LDPT*

*Elaborador por: Santiago Yaguargos*

## Automatización de los procesos de información

La automatización de procesos de información, es decir la administración de noticias y documentos, se utiliza Firebase como backend.

- **Componente ContainerView**

Se crea un componente personalizado llamado Container Web donde se diseña una estructura estándar de la barra de aplicaciones, botones de login y el menú.

```
<v-system-bar app height="64" dark color="transparent">
  <v-app-bar absolute color="indigo darken-4" dark
    src="@/assets/fondoBar.jpg" persistent>
    <v-app-bar-nav-icon @click="drawer = !drawer"></v-app-bar-nav-icon>
    <router-link to="/">
      
    </router-link>
    <v-divider light class="mx-1" inset vertical></v-divider>
    <v-toolbar-title>LIGA DEPORTIVA PARROQUIAL TOTORAS
    </v-toolbar-title>
    <v-spacer></v-spacer>
    <v-btn to="/admin" class="ma-1" outlined color="white">
      <v-icon>mdi-home</v-icon>
    </v-btn>
    <v-btn to="/login" class="ma-1" outlined color="white">
      <v-icon dark left>mdi-login</v-icon>
    </v-btn>
    <v-btn @click="logout" class="ma-1" outlined color="white">
      <v-icon dark left>mdi-logout</v-icon>
    </v-btn>
  </v-app-bar>
</v-system-bar>
<v-navigation-drawer
  id="app-drawer"
```



```

app
v-model="drawer"
temporary
dark
floating
mobile-break-point="991"
persistent
width="250"
>
<template v-slot:img="attrs">
  <v-img
src="https://i.pinimg.com/originals/da/b8/95/dab895a256053f09af23742ebeaed127.j
pg"
  v-bind="attrs"
  gradient="to top, rgba(0, 0, 0, .7), rgba(0, 0, 0, .7)"
  />
</template>
<v-list>
  <v-list-item two-line>
    <v-list-item-content>
      <v-list-item-avatar width="64" height="64">
        
      </v-list-item-avatar>
      <v-list-item-title align="center" class="title">MENÚ PRINCIPAL</v-list-
item-title>
      <v-list-item-subtitle align="center">Bienvenido!</v-list-item-subtitle>
    </v-list-item-content>
  </v-list-item>
</v-list>
<v-divider light class="mx-3 mb-3" />
<v-list dense nav>
  <v-list-item v-for="item in menu" :key="item.id" :to="item.to"
:href="item.href" indigo>

```

```

    <v-list-item-icon>
      <v-icon>{{ item.icon }}</v-icon>
    </v-list-item-icon>
    <v-list-item-content>
      <v-list-item-title>{{ item.title }}</v-list-item-title>
    </v-list-item-content>
  </v-list-item>
</v-list>
</v-navigation-drawer>

```

El arreglo *menú* contiene los ítems con un ícono, el título y la ruta establecida en el archivo `router.js`.

```

menu: [
  { title: "Inicio", icon: "mdi-home", to: "/" },
  { title: "Noticias", icon: "mdi-newspaper", to: "noticias" },
  { title: "Normativa", icon: "mdi-view-dashboard", to: "normativa" },
  { title: "Actas Oficiales", icon: "mdi-book", to: "actas" },
  { title: "Nosotros", icon: "mdi-lifebuoy", to: "nosotros" },
  {
    title: "Ir a Facebook",
    icon: "mdi-facebook",
    href: "https://www.facebook.com/LigaTotoras/"
  }
]

```

Mediante un dato booleano a través de la directiva *v-model*, muestra el menú al seleccionar su ícono ubicado en la barra de aplicaciones.

### Componente **SlideClubs**

Se desarrolla este componente debido al requerimiento del diseño de interfaces basado en el estándar de la página web de LigaPro

```

<v-sheet class="mx-auto" elevation="1">
  <v-slide-group class="pa-0" center-active show-arrows>

```

```

    <v-slide-item v-for="item of logoClubs" :key="item.id" v-slot:default="{ active,
toggle }">
      <router-link :to="{ name: 'clubes', params: { id: item.id } }">
        <v-card outlined color="white" class="ma-1" height="80" width="80"
@click="toggle">
          <v-layout column align-center justify-center>
            <v-img :width="item.width" :src="item.rutaLogo"></v-img>
          </v-layout>
          <v-row class="fill-height" align="center" justify="center">
            <v-scale-transition></v-scale-transition>
          </v-row>
        </v-card>
      </router-link>
    </v-slide-item>
  </v-slide-group>
</v-sheet>

```

- **Archivo store.js**

El archivo store.js es la tienda central de una aplicación con Vuex, allí se alojan el state, mutations, actions, getters.

El arreglo denominado *logoClubs* es declarado en el state de la tienda, de manera que se almacenan datos como la ruta del logo, el nombre del club y su presidente y de esta manera se recorre en el componente SlideClubs accediendo a dichos datos globales.

Para acceder a la tienda a través de un componente o una vista, es primordial importar desde Vuex dentro del script, los datos que van a ser mapeados de la siguiente manera:

```
import {mapState} from "vuex";
```

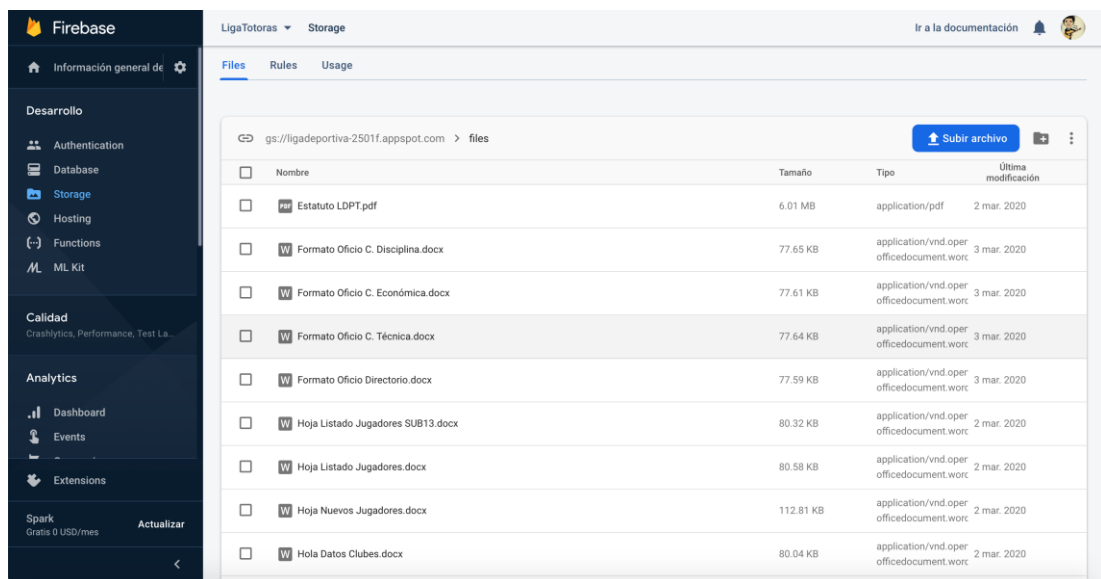
Para obtener datos del state se importa el mapState y se lo declara en las propiedades computadas con la siguiente estructura:

```
computed: {  
  ...mapState(["logoClubs"])  
}
```

- **Uso del servicio Firebase Storage**

Para la versión actual de la aplicación únicamente se almacenan los documentos manualmente en el Storage y son consultados a través de la url que se genera a través de la plataforma.

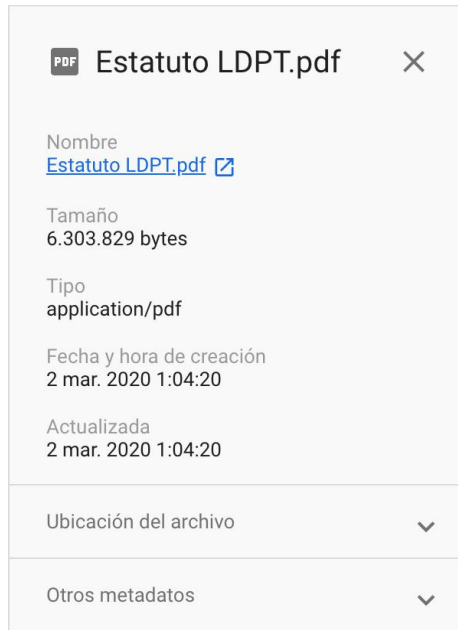
En el menú *Desarrollo*, al seleccionar el ítem *Storage* de Firebase se configura el servicio, luego se agregan los archivos a través de carpetas para una correcta organización, como indica la Fig. 48.



*Figura 48. Firebase Storage en la consola de Firebase*

*Elaborador por: Santiago Yaguargos*

Cada archivo almacenado en el Storage de Firebase genera una url de descarga como indica la Fig. 49.



*Figura 49. Archivo almacenado en Firebase Storage  
Elaborador por: Santiago Yaguargos*

En la vista de *Normativa* se reflejan los botones para descargar los determinados documentos que maneja la institución con sus clubes filiales, cumpliendo con el requerimiento.

Mediante la directiva *v-for* se realiza un doble recorrido al arreglo que contiene los enlaces de descarga con los nombres de los botones.

```
<v-card v-for="items in normatives" :key="items.id" flat class="transparent">
  <v-card-title>
    <v-toolbar class="indigo--text transparent" flat>
      <v-toolbar-title>
        <h2>{{ items.name }}</h2>
      </v-toolbar-title>
    </v-toolbar>
  </v-card-title>
  <v-divider class="mx-3 mb-3" />
  <v-layout row wrap>
    <v-flex xs12 sm6 md3 v-for="item in items.documents" :key="item.id">
      <div class="text-center">
```

```

<v-btn
  :href="item.href"
  class="ma-2"
  width="250"
  rounded
  outlined
  color="indigo"
>{{ item.name }}</v-btn>
</div>
</v-flex>
</v-layout>
</v-card>

```

- **Desarrollo del Inicio de Sesión**

Entre de los principales servicios que ofrece Firebase se encuentra *Authentication*, como indica la Fig. 50.

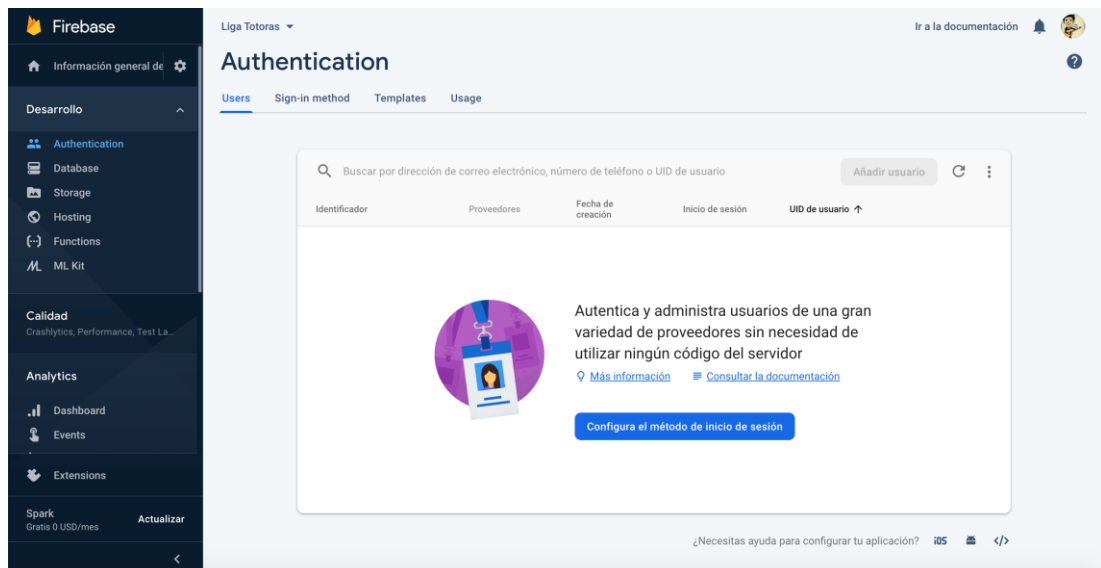


Figura 50. Firebase Authentication en la consola de Firebase

Elaborador por: Santiago Yaguargos

Al seleccionar en *Configurar el método de inicio de sesión*, se despliegan los proveedores que soporta Firebase.

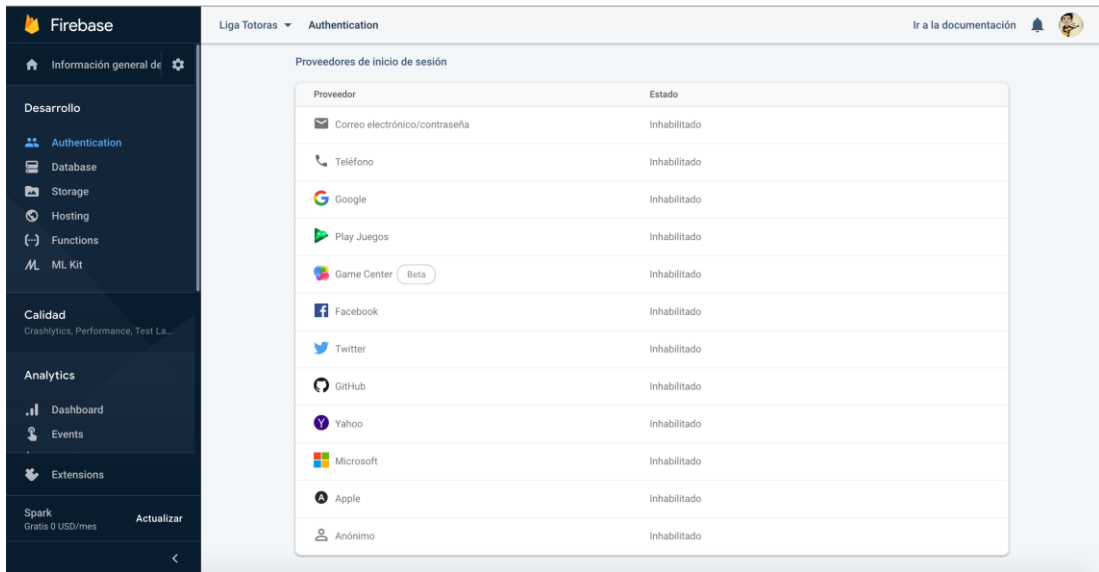


Figura 51. Proveedores de Firebase Authentication

Elaborador por: Santiago Yaguargos

Se procede con la habilitación, luego de seleccionar *correo electrónico/contraseña*, como proveedor.

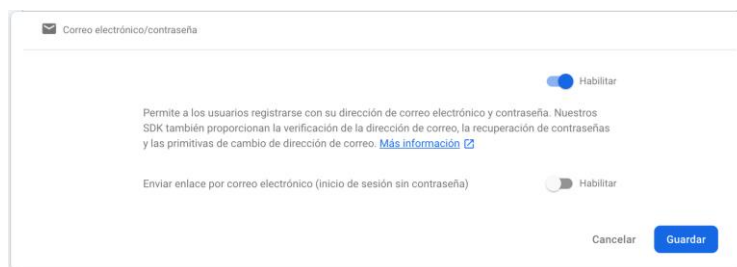


Figura 52. Habilitar proveedor Correo electrónico/contraseña en Firebase Authentication

Elaborador por: Santiago Yaguargos

Para cumplir con el requerimiento, se añade al usuario administrador, utilizando el correo electrónico que maneja la institución y se establece una contraseña que será modificada una vez entregado el producto.



Figura 53. Añadir usuario Correo electrónico/contraseña en Firebase Authentication

Elaborador por: Santiago Yaguargos

Una vez añadido el usuario, Firebase Authentication genera automáticamente un UID de usuario que es el código único que servirá para el desarrollo en Vue.

| Identificador                    | Proveedores | Fecha de creación ↓ | Inicio de sesión | UID de usuario ↑             |
|----------------------------------|-------------|---------------------|------------------|------------------------------|
| ligaparroquialtotoras@hotmail... | ✉           | 14 abr. 2020        |                  | b8PwtKTeFFOgnreD4YisH1WtDK02 |

Figura 54. Lista de usuarios Correo electrónico/contraseña en Firebase Authentication

Elaborador por: Santiago Yaguargos

- **Inicio de Sesión mediante Firebase Authentication**

En la propiedad *actions* de la tienda *store.js* se programan los métodos que realizan consultas a Firebase y el consumo de servicios web.

- **Método Inicio de Sesión**

```
login({ commit }, payload) {
  firebase.auth().signInWithEmailAndPassword(payload.email, payload.password)
    .then(res => {
      commit('setUser', { email: res.user.email, uid: res.user.uid })
      router.push({ name: 'admin' })
    })
    .catch(err => {
      commit('setError', err.code)
    })
},
```



Cambio del state a través de la Propiedad mutations.

```
setUser(state, payload) {  
  state.user = payload  
},  
setError(state, payload) {  
  state.error = payload  
},
```

En la vista de *Login* se crea un formulario con componentes propios de Vuetify, para la importación de los métodos almacenados en la tienda se procede de una manera parecida que en el componente *SlideClubs*. Para este caso se añade el mapeo de las acciones en la propiedad methods a través de *mapActions* desde vuex:

```
import { mapActions, mapState } from "vuex";  
export default {  
  methods: {  
    ...mapActions(["login"])  
  },  
  computed: {  
    ...mapState(["error"])  
  },
```

- **Método Cerrar Sesión**

```
logout({ commit }) {  
  firebase.auth().signOut()  
  commit('setUser', null)  
  router.push({ name: 'home' })  
},
```

Cuando el usuario este logueado, deben cambiar las anclas HTML de los botones que administran el inicio de sesión, es decir cuando no existe un usuario únicamente debe visualizarse el ícono para *Login* y caso contrario debe visualizarse el ícono de *Home*

*Administrador* y *Logout*, para ello se utiliza la directiva *v-if* y un getter *userExists* que devuelve un booleano con la existencia del usuario.

```
userExists(state) {  
  if (state.user === null || state.user === "" || state.user === undefined) {  
    return false  
  } else {  
    return true  
  }  
}
```

En el componente *ContainerWeb* se invocan el getter que detecta el usuario y el método que cierra sesión a través de propiedades.

```
import { mapActions, mapGetters } from "vuex";  
export default {  
  name: "ContainerWeb",  
  methods: {  
    ...mapActions(["logout"])  
  },  
  computed: {  
    ...mapGetters(["userExists"])  
  },  
}
```

La directiva *v-if* con el getter que devuelve un booleano, se utiliza en los botones que administran sesión.

```
<v-btn v-if="userExists" to="/admin" class="ma-1" outlined color="white">  
  <v-icon>mdi-home</v-icon>  
</v-btn>  
<v-btn v-if="!userExists" to="/login" class="ma-1" outlined color="white">  
  <v-icon dark left>mdi-login</v-icon>  
</v-btn>
```

```
<v-btn v-if="userExists" @click="logout" class="ma-1" outlined color="white">
  <v-icon dark left>mdi-logout</v-icon>
</v-btn>
```

- **Rutas protegidas**

Una vez desarrollado el proceso de autenticación con Firebase, se requiere proteger las rutas a las que se permitirán el acceso después de loguearse. De esta manera si la aplicación no detecta un usuario logueado, por mas que se escriba la dirección de la ruta en el url del navegador, no tendrá acceso.

En el archivo *router.js* se protegen las rutas que contienen las vistas del Usuario Administrador con la siguiente sentencia:

```
meta: { requiresAuth: true }
```

Una vez desarrollado el inicio de sesión, es necesaria una acción para que las rutas protegidas detecten un usuario autenticado y así permitir su acceso.

```
detectUser({ commit }, payload) {
  if (payload !== null) {
    commit('setUser', { email: payload.email, uid: payload.uid })
  } else {
    commit('setUser', null)
  }
},
```

Para llenar el arreglo user del state, se crea una mutación:

```
setUser(state, payload) {
  state.user = payload
},
```

La configuración requerida en el archivo *router.js* debe recorrer el estado de las rutas e invocar al método que detecta el usuario autenticado para permitir el acceso y almacenarlo en la constante *router* que es inicializada con la instancia de *Router*.

```

router.beforeEach((to, from, next) => {
  const rutaProtegida = to.matched.some(record => record.meta.requiresAuth)
  const user = firebase.auth().currentUser;
  if (rutaProtegida === true && user === null) {
    next({ name: 'login' })
  } else {
    next()
  }
})

```

Si se detecta que la ruta es protegida y no existe un usuario autenticado, esta configuración redirige a la vista de *Login*, sin embargo, al refrescar la página con una ruta autenticada se redirigirá a la vista de *Login* y este error se soluciona configurando un observador de estado en el archivo main.js, donde inicia la aplicación.

```

auth.onAuthStateChanged((user) => {
  if (user) {
    store.dispatch('detectUser', { email: user.email, uid: user.uid })
  } else {
    store.dispatch('detectUser', null)
  }
  new Vue({
    router,
    store,
    vuetify,
    render: h => h(App)
  }).$mount('#app')
})

```

Cuando el usuario se autentica correctamente, accede al aplicativo como administrador, en el cual maneja un su propio menú con los ítems: CLUBES, JUGADORES, EQUIPOS, CATEGORIAS, TORNEOS, NOTICIAS, HABILITAR EQUIPOS, HABILITAR JUGADORES, que son los procesos de gestión que requiere automatizar Liga Deportiva Parroquial Totoras.

- **Administrar Noticias**

Para desarrollar el CRUD de Noticias se accede al servicio *Cloud Firestore de Firebase* para almacenar y manipular colecciones.

- **Método obtener lista de noticias ordenadas descendientemente por fecha**

El método obtiene todas las colecciones de noticias almacenadas en la base de datos de Firebase ordenadas descendientemente de acuerdo a su fecha y en las limita para proceder a paginarlas. Se calcula el numero total de noticias y se divide para 6, que es el numero de noticias que se visualizaran por pagina, mediante el resultado se obtiene el numero de páginas necesarias y el componente se actualizará de acuerdo a dicha cantidad.

```
getAllNews({ commit }) {
  const news = []
  const forPages = 6
  var pages = 0
  db.collection('news').limit(forPages).orderBy('date', 'desc').get()
    .then(res => {
      res.forEach(doc => {
        let notice = doc.data();
        notice.id = doc.id
        news.push(notice)
      })
      commit('setAllNews', news)
    })
  db.collection('news').get()
    .then(res => {
      var total = res.size;
      pages = Math.ceil((total / forPages))
      commit('setPages', pages)
    })
},
```

- **Metodo para obtener una noticia por su Id**

El método obtiene una colección de noticia según el id de la misma.

```
getNewsById({ commit }, id) {
  db.collection('news').doc(id).get()
  .then(doc => {
    let notice = doc.data();
    notice.id = doc.id
    commit('setNewsById', notice)
  })
},
```

- **Metodo para editar una noticia.**

El método actualiza una colección de acuerdo a la noticia obtenida por su id.

```
editNotice({ commit }, notice) {
  db.collection('news').doc(notice.id).update({
    name: notice.name,
    date: notice.date,
    summary: notice.summary,
    urlImagen: notice.urlImagen,
    description: notice.description
  })
},
```

- **Metodo para eliminar una noticia por su Id**

El método elimina una colección de acuerdo con el id de la noticia.

```
deleteNews({ commit }, id) {
  db.collection('news').doc(id).delete()
  .then(() => {
    confirm("¿Está seguro de eliminar?") &&
```

```

        commit('deleteNews', id)
    })
},

```

- **Método para previsualizar una imagen**

El método permite previsualizar una imagen, una vez seleccionado el archivo tipo imagen desde el dispositivo que este en uso el aplicativo.

```

searchImage(event) {
    this.file = event;
    const reader = new FileReader();
    reader.readAsDataURL(this.file);
    reader.onload = e => {
        this.urlTemp = e.target.result;
    };
    return this.file;
},

```

- **Método para almacenar una imagen en Firebase**

El método carga una imagen al *Storage* de Firebase en una carpeta llamada “news”, el archivo se almacena con la fecha y el respectivo nombre del archivo, además guarda la dirección url de descarga del archivo en una variable.

```

async uploadImage() {
    try {
        const refImage = firebase
            .storage()
            .ref("/news/" + new Date().get + "-" + this.file.name);
        const res = await refImage.put(this.file);
        const urlDownload = await refImage.getDownloadURL();
        this.urlImage = urlDownload;
    } catch (error) {
        console.log(error);
    }
}

```

```
    }  
  }  
},
```

- **Método para obtener noticias cuando cambia el ítem de la paginación**

El siguiente método se utiliza para obtener la colección de noticias de acuerdo con la estructura de su paginación, es decir, al seleccionar otra página se realiza la consulta de las siguientes seis noticias.

```
onPageChange() {  
  const news = [];  
  const forPages = 6;  
  var startAfter = this.pages * (this.page - 1);  
  db.collection("news")  
    .limit(forPages)  
    .orderBy("date", "desc")  
    .startAfter(startAfter)  
    .get()  
    .then(res => {  
      res.forEach(doc => {  
        let notice = doc.data();  
        notice.id = doc.id;  
        news.push(notice);  
        commit("setAllNews", news);  
      })  
    })  
}
```

- **Reglas de seguridad en Database Firebase**

*Database Firebase* requiere una configuración de seguridad para que la escritura en la base de datos sea posible únicamente cuando existe un usuario autenticado.



Para su configuración, se accede al ítem *Database* en la consola de Firebase y en la pestaña *Reglas*, se modifica de acuerdo con las necesidades del proyecto.

Las reglas permiten la lectura de la base de datos para todos y limita su escritura para usuarios autenticados, protegiendo la manipulación de datos.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if true;
      allow write: if request.auth.uid !=null;
    }
  }
}
```

### **Automatización en procesos de gestión**

En base a los requerimientos, los procesos de gestión en Liga Deportiva Parroquial Totoras que necesitan ser automatizados, se desarrollan servicios web en Spring.

- **Packages del proyecto en Spring**

La estructura de los archivos contiene los packages creados para los diferentes archivos necesarios para el desarrollo del Backend de la aplicación.



Figura 55. Estructura de los archivos del Proyecto en Spring

Elaborador por: Santiago Yaguargos

- **Configuración del archivo `application.yml`**

Inicialmente se configura el archivo de propiedades `application.yml`. Se agrega una propiedad personalizada *league*, necesaria para conectar Firebase con Spring y el uso del interceptor para el nivel de seguridad. Se configuran propiedades para la conexión con la base de datos PostgreSQL.

`league:`

`firebase:`

`configuration-file: /firebase/ligadeportivatotoras-firebase-adminsdk.json`

`database-url: https://ligadeportivatotoras.firebaseio.com`

`server:`

`port : 8095`

`spring:`

`datasource:`

`url: jdbc:postgresql://localhost:5432/ligaTotoras`

`username: postgres`

`password: root`

`jpa:`

`properties:`

`hibernate:`

`dialect: org.hibernate.dialect.PostgreSQLDialect`

`jdbc:`

`lob:`

`non_contextual_creation: true`

`hibernate:`

`ddl-auto: update`

`logging:`

`level:`

`root: INFO`

La configuración de la propiedad *datasource* debe contener las credenciales idénticas con las que se accede desde el administrador base de datos PGAdmin.

En el package config se crean archivos de configuración del proyecto de Spring como son: *FirebaseInitializer*, *JWTFilter* y *SpringSecurity* donde se usa la denotación `@Configuration`.

- **Configuración del archivo *FirebaseInitializer***

El archivo llamado *FirebaseInitializer.java* permite la configuración para la conexión de Firebase y facilita un interceptor desde Spring.

```
package com.ldpt.LigaDeportiva.config;
import java.io.IOException;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.io.ClassPathResource;
import com.google.auth.oauth2.GoogleCredentials;
import com.google.firebase.FirebaseApp;
import com.google.firebase.FirebaseOptions;
```

```
@Configuration
```

```
public class FirebaseInitializer {
```

```
    @Value("${league.firebase.configuration-file}")
```

```
    private String firebaseConfigPath;
```

```
    @Value("${league.firebase.database-url}")
```

```
    private String databaseUrl;
```

```
    @Bean
```

```
    FirebaseApp initializeFirebaseBaseApp() throws IOException {
```

```
        FirebaseOptions options = new FirebaseOptions.Builder()
```

```
            .setCredentials(
```

```
                GoogleCredentials.fromStream(new
```

```
                    ClassPathResource(firebaseConfigPath).getInputStream()))
```

```
            .setDatabaseUrl(databaseUrl).build();
```

```

        return FirebaseApp.initializeApp(options,
        FirebaseApp.DEFAULT_APP_NAME);
    }
}

```

- **Configuración del interceptor JWT**

El archivo llamado *JWTFilter* permite configurar un nivel de seguridad a las peticiones con la intercepción de tokens generados desde Firebase.

Al desarrollar servicios web que están destinados a ser alojados en una IP pública, se requiere un nivel de seguridad y para el presente proyecto se conecta Firebase y Spring a través de un interceptor, es decir, para una petición al Backend, primero detecta si existe un token de acceso generado desde Firebase, caso contrario la petición será denegada.

```

package com.ldpt.LigaDeportiva.config;
import java.io.IOException;
import javax.servlet.FilterChain;
import javax.servlet.ServletException;
import javax.servlet.ServletRequest;
import javax.servlet.ServletResponse;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.filter.GenericFilterBean;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseAuthException;
import com.ldpt.LigaDeportiva.controller.CategoryController;

@Configuration
public class JWTFilter extends GenericFilterBean {

    @Override

```

```
public void doFilter(ServletRequest req, ServletResponse resp, FilterChain
filterChain)
```

```
        throws IOException, ServletException {
    HttpServletRequest request = (HttpServletRequest) req;
    HttpServletResponse response = (HttpServletResponse) resp;
    String token = request.getHeader("Authorization");

    if ("OPTIONS".equalsIgnoreCase(request.getMethod())) {
        response.sendError(HttpServletResponse.SC_OK, "success");
        return;
    }

    if (allowRequestWithoutToken(request)) {
        response.setStatus(HttpServletResponse.SC_OK);
    } else {
        if (token == null) {

response.sendError(HttpServletResponse.SC_UNAUTHORIZED);
            return;
        }
        try {
            FirebaseAuth.getInstance().verifyIdToken(token);
        } catch (FirebaseAuthException e) {

response.sendError(HttpServletResponse.SC_UNAUTHORIZED);
            return;
        }
    }
    filterChain.doFilter(req, resp);
}
```

```
public boolean allowRequestWithoutToken(HttpServletRequest request) {
```

```

        return
        "/api".concat(CategoryController.CATEGORY_INSERT_URI).equals(request.getRequestURI());
    }
}

```

El método `allowRequestWithoutToken`, devuelve un booleano con la peticiones que no requieren un token.

- **Configuración de Spring Security**

En el archivo llamado *SecurityConfiguration.java* se configura parte de la seguridad, al permitir o bloquear el acceso a los servicios web.

```

package com.ldpt.LigaDeportiva.config;
import org.springframework.boot.context.properties.EnableConfigurationProperties;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityConf
igurerAdapter;
import org.springframework.security.config.http.SessionCreationPolicy;

@Configuration
@EnableConfigurationProperties
public class SecurityConfiguration extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {

        http.cors().and().csrf().disable().authorizeRequests().antMatchers("/api/auth/*
*").permitAll().anyRequest().authenticated().and().httpBasic().and().sessionManage
ment().sessionCreationPolicy(SessionCreationPolicy.STATELESS);
    }
}

```

- **Creación de entidades (Entidad Jugadores)**

Una facilidad que brinda el framework Spring es la creación de entidades y en la ejecución del proyecto, crea automáticamente las tablas y sus respectivas relaciones en la base de datos.

De acuerdo con la lógica de negocio, se requieren crear entidades según el modelo de base de datos. A continuación, se muestra la creación de la entidad *Jugadores*.

```
package com.ldpt.LigaDeportiva.model;
import java.io.Serializable;
import java.time.LocalDate;
import java.time.LocalDateTime;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.SequenceGenerator;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;
import org.hibernate.annotations.ColumnDefault;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
```

```
@Entity
```

```
@Table(name = "player")
```

```
@NoArgsConstructor
```

```
@AllArgsConstructor
```

```
@Data
```

```
public class Player implements Serializable {
```

```

private static final long serialVersionUID = -4515176312624967399L;
    @Id
    @SequenceGenerator(name = "pk_sequence_player", sequenceName =
"ldpt_player_seq", allocationSize = 1)
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"pk_sequence_player")
    private Long id;
    @NotNull
    @ColumnDefault("1")
    private Boolean active;
    @NotNull
    private LocalDateTime creationDateTime;
    @NotNull
    @Column(unique = true)
    private String identification;
    @Column(unique = true)
    private String code;
    @NotNull
    private String lastName;
    @NotNull
    private String firstName;
    @NotNull
    private String address;
    @NotNull
    private LocalDate birthday;
    @NotNull
    private Character gender;
    @ColumnDefault("0")
    private Integer playerType;
    @NotNull
    private String country;
    @NotNull
    private String city;

```



```

    private Integer dorsal;
    private String imagen;
    @ManyToOne
    @JoinColumn(name = "idClub")
    private Club club;
}

```

La entidad contiene anotaciones que permiten ir creando la base de datos de acuerdo con los datos y relaciones establecidas entre tablas. Las anotaciones actuales permiten la estructura de la tabla, generar un autoincremental, indicar los datos que son requeridos y la relación con otra entidad creada.

- **Creación de un Servicio de Registro**

Se deben crear las diferentes capas de acuerdo con el uso del patrón de diseño MVC en Spring Framework.

Las capas creadas son: *repository*, en el manejo de datos; *service*, para implementar servicios; y *controller*, como capa de presentación. Finalmente, se desarrollan interfaces en la capa *operations*.

- **Creación de una interfaz en la capa repository (Jugadores)**

Se crea el archivo *PlayerRepository.java* de tipo interfaz, encargada en el manejo de datos de los Jugadores. La interfaz contiene las consultas realizadas a la base de datos, en este caso una consulta para obtener una lista de jugadores según su cédula y otra consulta para buscar jugadores según su club, donde se crean Querys de acuerdo con la lógica de negocio y requerimientos del sistema.

```

package com.ldpt.LigaDeportiva.repository;
import java.util.List;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

```

```

import com.ldpt.LigaDeportiva.model.Player;

@Repository
public interface PlayerRepository extends JpaRepository<Player, Long> {

    Player getPlayerByIdentification(String identification);

    @Query("SELECT p.id, p.firstName, p.lastName FROM Player p WHERE
p.club.id = :idClub")
    List<Object[]> findByClub(@Param("idClub") Long idClub);

}

```

- **Creación de una clase DTO**

Es recomendable el uso de una clase DTO, debido a que facilita al desarrollador manejar los datos necesarios únicamente para determinado proceso.

En el proceso para insertar los datos de un nuevo club se crea la clase *ClubInsertDTO*. java y se maneja la respectiva data según el proceso requerido.

```

package com.ldpt.LigaDeportiva.dto;
import java.util.Date;
import javax.validation.constraints.NotEmpty;
import javax.validation.constraints.NotNull;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class ClubInsertDTO {

```

```

Long id;
@NotNull(message = "name can't be null")
@NotEmpty(message = "name can't be empty")
String name;
@NotNull(message = "foundationDate can't be null")
@NotEmpty(message = "foundationDate can't be empty")
Date foundationDate;
String ministerialAgreement;
@NotNull(message = "phone can't be null")
@NotEmpty(message = "phone can't be empty")
String phone;
String mail;
@NotNull(message = "president can't be null")
@NotEmpty(message = "president can't be empty")
String president;
String vicepresident;
String secretary;
String treasurer;
@NotNull(message = "delegate1 can't be null")
@NotEmpty(message = "delegate2 can't be empty")
String delegate1;
String delegate2;
@NotNull(message = "logo can't be null")
@NotEmpty(message = "logo can't be empty")
String logo;
@NotNull(message = "league can't be null")
Long league;
String message;
}

```

Mediante la clase DTO, se establece una data de acuerdo al proceso de inserción de Clubes, con los datos que son requeridos obligatoriamente en la base de datos y desde

el frontend pueden ser controlados con el manejo de mensajes de error al enviar un dato vacío o nulo.

- **Creación de una interfaz en la capa operations (Clubes)**

En la interfaz llamada *ClubOperations.java* se declaran todas las funciones a implementar y para el caso determinado en la inserción de clubes se utiliza la clase DTO en lugar de las entidades, limitando así, el proceso requerido.

```
package com.ldpt.LigaDeportiva.operations;
import java.util.List;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.BindingResult;
import com.ldpt.LigaDeportiva.dto.ClubInsertDTO;
import com.ldpt.LigaDeportiva.model.Club;

public interface ClubOperations {
    public ResponseEntity<List<Club>> getClubAll();
    public ResponseEntity<ClubInsertDTO> insertClub(ClubInsertDTO club,
BindingResult bindingResult);
}
```

- **Creación de servicios (Categorías)**

En la clase *CategoryService* se implementan los servicios con la lógica de negocio de acuerdo a los requerimientos.

```
package com.ldpt.LigaDeportiva.service;
import java.util.List;
import java.util.stream.Collectors;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpHeaders;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Service;
```

```

import org.springframework.validation.BindingResult;

import com.ldpt.LigaDeportiva.dto.CategoryInsertDTO;
import com.ldpt.LigaDeportiva.model.Category;
import com.ldpt.LigaDeportiva.operations.CategoryOperations;
import com.ldpt.LigaDeportiva.repository.CategoryRepository;

@Service
public class CategoryService implements CategoryOperations {

    @Autowired
    CategoryRepository categoryRepository;

    @Override
    public ResponseEntity<List<Category>> getCategoriesAll() {
        List<Category> categoryList = categoryRepository.findAll();
        return new ResponseEntity<List<Category>>(categoryList, new
HttpHeaders(), HttpStatus.OK);
    }

    @Override
    public ResponseEntity<CategoryInsertDTO>
insertCategory(CategoryInsertDTO category, BindingResult bindingResult) {
        category.setMessagge("ok");
        List<String> errors = null;
        if (bindingResult.hasErrors()) {
            errors = bindingResult.getAllErrors().stream().map(e ->
e.getCode() + "." + e.getDefaultMessage())
                .collect(Collectors.toList());
            category.setMessagge(errors.toString());
            return new ResponseEntity<CategoryInsertDTO>(category,
new HttpHeaders(), HttpStatus.PRECONDITION_REQUIRED);
        }
        Category newCategory = new Category();
        newCategory.setActive(true);

```

```

        newCategory.setName(category.getName());
        newCategory.setSerie(category.getSerie());
        newCategory.setDescription(category.getDescription());
        categoryRepository.saveAndFlush(newCategory);
        category.setMessagge("ok");
        return new ResponseEntity<CategoryInsertDTO>(category, new
HttpHeaders(), HttpStatus.OK);
    };
}

```

- **Creación de un controlador en la capa de presentación (Categorías)**

En la clase *CategoryController* se declaran constantes que almacenan la dirección y el nombre de las API's que son mapeados con las respectivas anotaciones a los métodos que recibirán el encabezado con el token de acceso y el cuerpo con los datos en formato json, como parámetros desde el Frontend.

```

package com.ldpt.LigaDeportiva.controller;
import java.util.List;
import javax.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestHeader;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;
import com.ldpt.LigaDeportiva.dto.CategoryUpdateDTO;
import com.ldpt.LigaDeportiva.dto.DefaultResponse;
import com.ldpt.LigaDeportiva.model.Category;

```

```

import com.ldpt.LigaDeportiva.operations.CategoryOperations;

@RestController
@RequestMapping("/api")
@CrossOrigin
public class CategoryController {
    @Autowired
    CategoryOperations categoryOperations;

    public final static String CATEGORY_LIST_URI = "/auth/categoryList";
    public final static String CATEGORY_INSERT_URI = "/auth/categoryInsert";
    public static final String CATEGORY_UPDATE_URI = "/auth/updateCategory";
    public static final String CATEGORY_DELETE_URI = "/auth/deleteCategory";

    @GetMapping(CATEGORY_LIST_URI)
    public ResponseEntity<List<Category>> getCategoriesList(
        @RequestHeader(value = "Authorization") String
authorizationToken) {
        return categoryOperations.getCategoriesAll();
    };

    @PostMapping(CATEGORY_INSERT_URI)
    public ResponseEntity<CategoryUpdateDTO> insertCategory(
        @RequestHeader(value = "Authorization") String
authorizationToken,
        @Valid @RequestBody CategoryUpdateDTO
categoryUpdateDTO, BindingResult bindingResult) {
        return categoryOperations.insertCategory(categoryUpdateDTO,
bindingResult);
    }

    @PostMapping(CATEGORY_UPDATE_URI)
    @ResponseBody
    public ResponseEntity<DefaultResponse> updateCategory(@Valid
@RequestBody CategoryUpdateDTO categoryUpdateDTO,

```

```

        BindingResult bindingResult, @RequestHeader(value =
"Authorization") String authorizationToken) {
            return categoryOperations.updateCategory(categoryUpdateDTO,
bindingResult);
        }

        @PostMapping(CATEGORY_DELETE_URI)
        @ResponseBody
        public ResponseEntity<DefaultResponse> deleteVideo(@Valid
@RequestBody CategoryUpdateDTO categoryUpdateDTO,
            BindingResult bindingResult, @RequestHeader(value =
"Authorization") String authorizationToken) {
            return categoryOperations.deleteCategory(categoryUpdateDTO,
bindingResult);
        }
    }
}

```

- **Configuración para el consumo servicios con Axios**

Vue facilita al desarrollador una librería para del consumo de servicios llamada *Axios* y para su importación, en el archivo main.js se declaran la siguiente sentencia:

```

import axios from 'axios'
import VueAxios from 'vue-axios'

```

En el mismo archivo se configura el uso de la librería y como parte de dicha configuración se declara la dirección url donde se alojan los servicios web.

```

Vue.use(VueAxios, axios)
axios.defaults.baseURL = 'http://localhost:8095/api/auth';
window.axios = axios
Vue.config.productionTip = false

```



## Consumo de servicios con Axios

Se recomienda crear acciones en la tienda de Vue para el consumo de servicios, de esta manera se crea una estructura adecuada y con mayor organización, sin embargo, de acuerdo con el desarrollador se puede acceder a las API desde el script de un componente, cuando las acciones no pretenden ser reutilizadas.

- **Observador de estado para detectar token**

Anteriormente se utilizó un observador para detectar si existe un usuario autenticado, en esta ocasión agrega la sentencia para obtener el *tokenId* del usuario.

```
auth.onAuthStateChanged((user) => {
  if (user) {
    store.dispatch('detectUser', { email: user.email, uid: user.uid })
    user.getIdToken().then(res => {
      store.dispatch('detectToken', res)
    });
  } else {
    store.dispatch('detectUser', null)
  }
  new Vue({
    router,
    store,
    vuetify,
    render: h => h(App)
  }).$mount('#app')
})
```

En la tienda se crea una acción para obtener el token detectado.

```
detectToken({ commit }, token) {
  if (token !== null) {
    commit('setToken', token)
  } else {
```

```
    commit('setToken', null)
  }
},
```

Mediante una mutación se cambia el estado de la variable token.

```
setToken(state, token) {
  state.token = token
},
```

- **Uso de componente tipo tabla de Vuetify**

Para manejar los datos y los procedimientos que requiere el usuario administrador se utiliza el componente llamado *v-data-table* y determinadas propiedades para un adecuado funcionamiento.

La estructura de los componentes *v-data-table* carga datos en sus encabezados y en los ítems de acuerdo a las propiedades establecidas, además se agregan botones de acción por cada ítem cargado y un campo para realizar búsquedas inteligentes donde lo requiera el administrador.

Las propiedades utilizadas comúnmente en los componentes *v-tabla-table* utilizados en el presente proyecto son las siguientes:

**headers:** que contiene los encabezados de la tabla.

**search:** para búsqueda inteligente en la table.

**items:** la propiedad obtiene los datos almacenados desde arreglo con formato json.

**sort-by:** propiedad para ordenar la tabla de acuerdo con el valor enviado.

**hide-default-footer:** elimna las paginación selectiva de la página.

**show-select:** mostrar ítems para realizar acciones de selección.

- **Tabla Categorías**

```

<v-data-table :headers="headers" :search="search" :items="championship"
sort-by="id">
  <template v-slot:top>
    <v-toolbar color="indigo" dark>
      <v-btn outlined @click="dialog = true">
        <v-icon>mdi-plus-circle</v-icon>
      </v-btn>
      <v-spacer></v-spacer>
      <v-text-field
        v-model="search"
        append-icon="mdi-magnify"
        label="Buscar"
        hide-details
      ></v-text-field>
    </v-toolbar>
  </template>
  <template v-slot:item.action="{ item }">
    <v-icon class="mr-3" @click="editItem(item)">mdi-pencil</v-icon>
    <v-icon class="mr-3" @click="deleteItem(item)">mdi-delete</v-icon>
    <v-icon>mdi-eye</v-icon>
  </template>
  <template v-slot:no-data>
    <v-btn text color="indigo" @click="championshipList">Cargando...</v-btn>
  </template>
</v-data-table>

```

El arreglo que contiene los headers se declaran en la data del componente:

```

headers: [
  {
    sortable: false,
    value: "action"
  }
]

```

```

    },
    {
      sortable: false,
      text: "NOMBRE",
      value: "name"
    },
    {
      sortable: false,
      text: "AÑO",
      value: "year"
    },
    {
      sortable: false,
      text: "DESCRIPCIÓN",
      value: "description"
    }
  ]

```

- **Método para consumir servicio web Lista de Torneos**

El método consume el servicio web utilizando Axios, en el encabezado se envía el token de acceso que es interceptado desde Spring tras petición del servicio.

```

championshipList() {
  this.loading = true;
  setTimeout(() => {
    this.axios
      .get("/championshipList", {
        headers: {
          Authorization: this.token
        }
      })
      .then(res => {
        this.loading = false;

```

```

    this.championship = res.data;
  })
  .catch(err => {
    this.loading = false;
    switch (err.response.status) {
      case 401:
        alert("Su sesión a caducado");
        this.logout();
        router.push({ name: "login" });
        break;
      case 412:
        break;
      case 428:
        break;
    }
  });
}, 2000);
},

```

- **Uso de Vuelidate (Formulario Inicio de Sesión)**

El uso de la librería *Vuelidate* permite personalizar las validaciones agregando las características necesarias.

La instalación de *Vuelidate* se ejecuta utilizando el gestor de paquetes npm desde el terminal y ubicándose en la carpeta contenedora del proyecto:

```
~ npm install vuelidate
```

Dentro del script de la vista que contiene determinado formulario a validar, se importan las diferentes variables necesarias.

```
import { required, email, minLength } from "vuelidate/lib/validators";
```

En el script se declara una propiedad llamada *validations* para establecer las características de validación.

```
validations: {  
  email: { required, email },  
  password: { required, minLength: minLength(6) }  
}
```

Se crean componentes para personalizar los mensajes de error y se muestran mediante el uso de la directiva *v-if*.

```
v-if="!$v.email.email" Mensaje: *Email incorrecto.
```

```
v-if="!$v.password.minLength" Mensaje: *Usuario o contraseña incorrectos
```

- **Manejo de errores de Firebase Auth**

Una ventaja de usar el servicio de autenticación con Firebase es que de acuerdo con las acciones que se tomen en el formulario de *Login*, se devuelven los errores y son capturados en el *catch*, sin embargo, estos mensajes suelen ser en inglés al ser propios de Firebase.

El error determinado desde Firebase, de acuerdo con la acción en el intento de ingresar al sistema, se guarda en una variable al ejecutar el método *Login* y se la obtiene en el componente que contiene el formulario *Login* a través de una propiedad computada.

```
...mapState(["error"])
```

De acuerdo con el error, se personaliza el mensaje en pantalla a través de la directiva *v-if*.

```
v-if="error === 'auth/user-not-found' Mensaje: *Usuario o contraseña incorrectos
```

```
v-if="error === 'auth/wrong-password' Mensaje: *Usuario o contraseña incorrectos<
```

```
v-if="error === 'auth/invalid-email' Mensaje: *Usuario o contraseña incorrectos
```

v-if="error === 'auth/too-many-requests' **Mensaje:** Demasiados intentos de inicio de sesión fallidos. Por favor, inténtelo más tarde.

## **Implementación de la Aplicación Web Progresiva**

Un requisito necesario para el correcto funcionamiento de la PWA es el alojamiento en un protocolo seguro y para cumplirlo se opta por el uso del servicio de hosting en Firebase donde se implementará el Frontend del producto final.

*Firebase Hosting* tiene una baja complejidad en su configuración que se establece de la siguiente manera:

Primero, es necesario instalar de manera global las herramientas de Firebase a través de la terminal con el siguiente comando:

```
~npm install -g firebase-tools
```

El uso del término `-g` sirve para una instalación global de las herramientas. Es importante indicar que dicha instalación se debe realizar una sola vez en el ordenador y a futuro únicamente actualizar los paquetes.

Luego de finalizar con la instalación de las herramientas de Firebase, es primordial ubicarse en la carpeta del proyecto, o a su vez abrir una terminal desde Visual Studio Code con el proyecto abierto y ejecutar el siguiente comando para autenticarse a través de Google:

```
~firebase login
```

Para inicializar el proyecto de Vue se utiliza el siguiente comando:

```
~firebase init
```

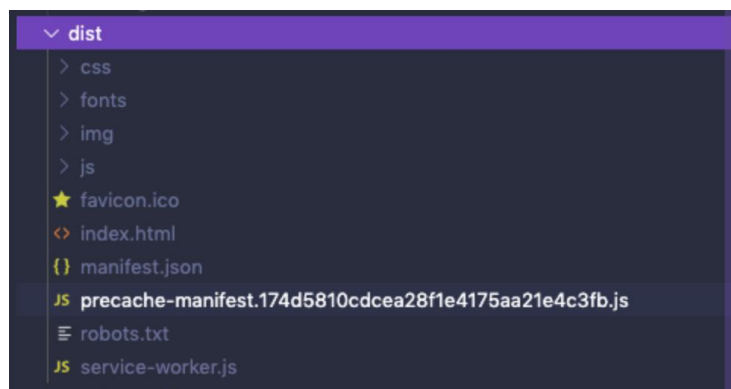
Al ejecutar este comando se requiere seleccionar los servicios utilizados de Firebase que para el proyecto actual son Firestore, Hosting y Storage. Seguidamente seleccionar el proyecto de a subir. Luego pregunta por el nombre para los archivos que contienen la configuración de los servicios de Firebase y para este caso se mantienen por defecto.

Después preguntara por el directorio publico, que será el directorio llamado “*dist*”, donde se compilara el proyecto. finalmente pregunta si utiliza un index.html al cual se escribe el termino “*y*”.

Una vez completada la inicialización, se procede a la compilación del proyecto con el siguiente comando:

***~npm run build***

Al ejecutar el comando se compila el proyecto en el directorio llamado “*dist*” como indica la Fig.



*Figura 56. Directorio público “dist” compilado*

*Elaborado por: Santiago Yaguargos*

Finalmente se implementa la aplicación en el hosting con el comando:

***~firebase deploy***

A través de la consola de firebase en el proyecto Liga Totoras en el menú Hosting, se obtiene la url, que esta compuesta por el id del proyecto seguido de firebaseapp.com.

La aplicación es implementada exitosamente y se puede acceder a través de:

<https://ligadeportivatotoras.web.app> o <https://ligadeportivatotoras.firebaseio.com>

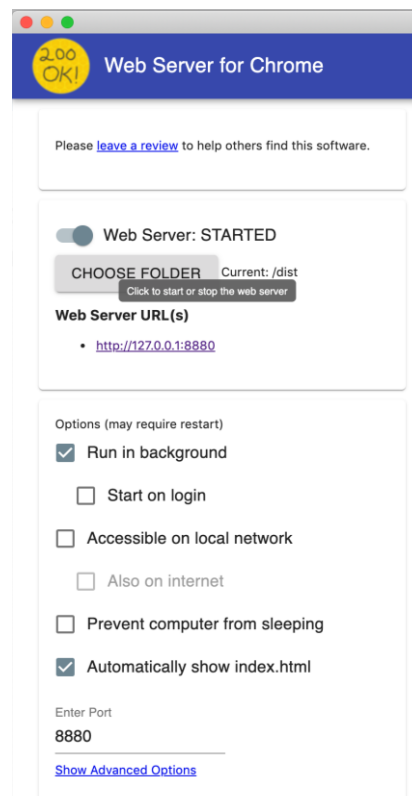
## **Constucción de la Aplicación Web Progresiva**



Cuando el usuario final ingrese al aplicativo, deberá sentir una cómoda accesibilidad con un bajo nivel de complejidad, por lo que el diseño de la aplicación esta basado en propiedades responsivas para que se adapte a cualquier dispositivo. El aplicativo tendrá un alto impacto en su diseño para su configuración como PWA.

Una vez finalizado el desarrollo de la aplicación e implementada en el hosting https de Firebase, se procede con la configuración de los archivos necesarios para la correcta funcionalidad de la Aplicación Web Progresiva.

El uso de la herramienta Web Server For Chrome facilita al desarrollador un servidor local con el protocolo “<http://127.0.0.1:8880>” de la carpeta con el proyecto compilado, como indica la Fig.



*Figura 57. Web Server for Chrome  
Elaborado por: Santiago Yaguargos*

- **Archivo registerServiceWorker.js**

La configuración establecida por defecto al instalar el complemento PWA desde el administrador de proyectos, permite registrar el *ServiceWorker* en el navegador donde se utilice la aplicación.

```
import { register } from 'register-service-worker'
if (process.env.NODE_ENV === 'production') {
  register(`${process.env.BASE_URL}service-worker.js`, {
    ready() {
      console.log(
        'App is being served from cache by a service worker.\n' +
        'For more details, visit https://goo.gl/AFskqB'
      )
    },
    registered() {
      console.log('Service worker has been registered.')
    },
    cached() {
      console.log('Content has been cached for offline use.')
    },
    updatefound() {
      console.log('New content is downloading.')
    },
    updated() {
      console.log('New content is available; please refresh.')
    },
    offline() {
      console.log('No internet connection found. App is running in offline mode.')
    },
    error(error) {
      console.error('Error during service worker registration:', error)
    }
  })
}
```

```
}
```

- **Archivo manifest.json**

El archivo de manifiesto se declaran las características de diseño que tomará la PWA al ser instalada en un dispositivo. Entre las características se establecen los tamaños de imagen adecuados para que, al instalar la aplicación se visualice correctamente el ícono.

```
{  
  "name": "LIGA DEPORTIVA PARROQUIAL TOTORAS",  
  "short_name": "LDPT",  
  "icons": [  
    {  
      "src": "./img/icons/android-chrome-192x192.png",  
      "sizes": "192x192",  
      "type": "image/png"  
    },  
    {  
      "src": "./img/icons/android-chrome-512x512.png",  
      "sizes": "512x512",  
      "type": "image/png"  
    }  
  ],  
  "start_url": "./",  
  "display": "fullscreen",  
  "background_color": "#012e67",  
  "theme_color": "#012e67"  
}
```

Estas configuraciones se verifican desde *Google Chrome* y las *Herramientas del programador* en la sección *Application* y el ítem *Manifest*, como indica la Fig. 58

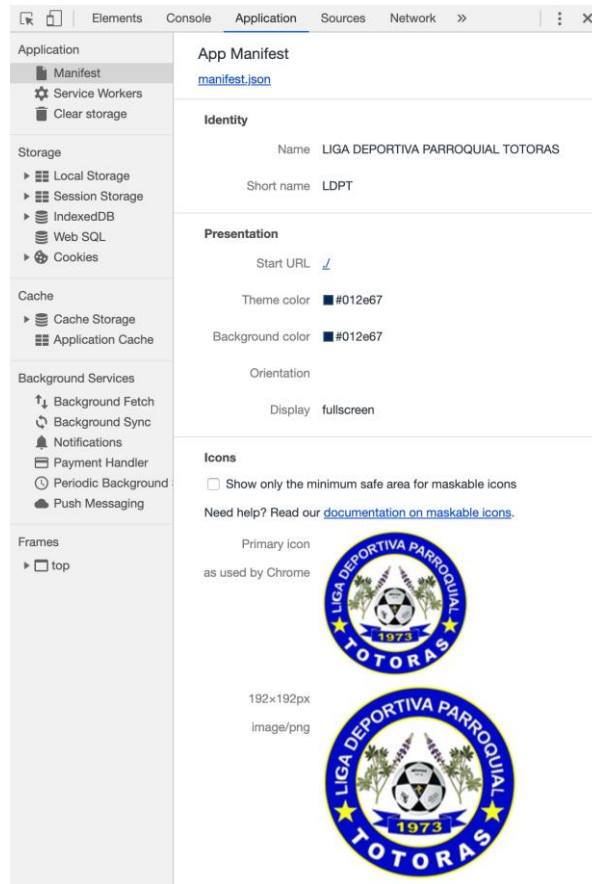


Figura 58. Manifest desde Google Chrome  
Elaborado por: Santiago Yaguargos

- **Crear archivo vue.config.js**

En el proyecto de Vue se crea el archivo llamado vue.config.js, donde se declaran las configuraciones necesarias para el funcionamiento de la PWA mediante *workbox* (librería utilizada para trabajar con estrategias de caché en Vue).

```

module.exports = {
  pwa: {
    name: "LIGA DEPORTIVA PARROQUIAL TOTORAS",
    workboxPluginMode: "InjectManifest",
    workboxOptions: {
      swSrc: "public/service-worker.js"
    }
  }
};

```

- **Configuración del service Worker**

En el *ServiceWorker* se establece la configuración del *workbox* para obtener información del archivo de manifiesto pre almacenado en caché

```
self.__precacheManifest = [].concat(self.__precacheManifest || []);
workbox.setConfig({
  debug: true
});
workbox.precaching.suppressWarnings();
workbox.precaching.precacheAndRoute(self.__precacheManifest, {});
```

Al finalizar la configuración inicial de *workbox*, se compila nuevamente el proyecto y se verifica la correcta instalación del *ServiceWorker* desde Google Chrome y las *Herramientas del programador* en la sección *Application* y el item *Service Workers*, como indica la Fig.59

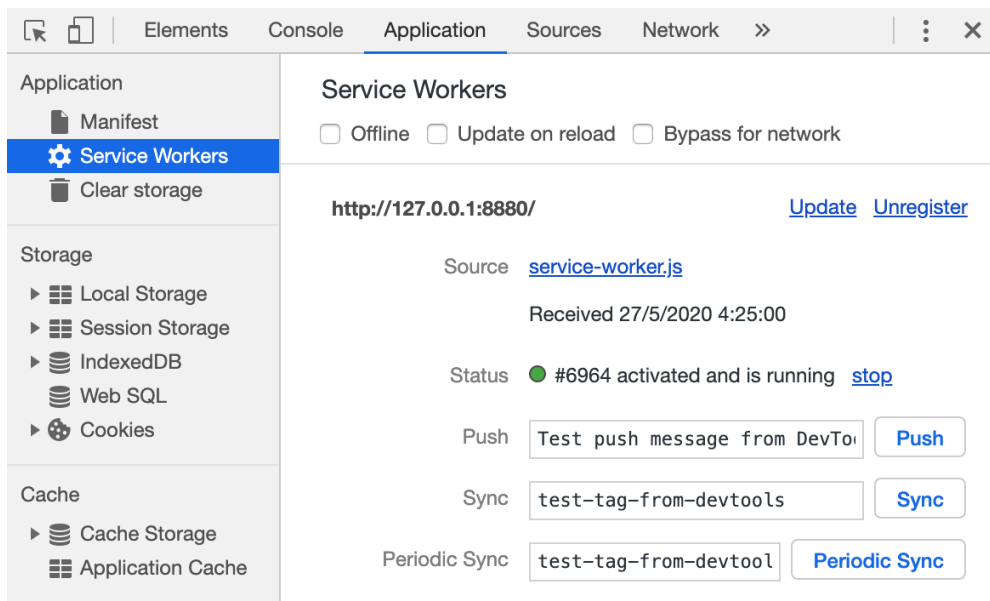


Figura 59. Manifiesto desde Google Chrome

Elaborado por: Santiago Yaguargos

Se crea un archivo que contiene el pre caché almacenado y en su estructura se almacena la ubicación de los archivos a guardar.

```
self.__precacheManifest = [  
  {  
    "revision": "1197567fa75b595b1353b03c648c9f05",  
    "url": "/img/logoLDPT.1197567f.png"  
  },  
  {  
    ...  
  },  
];
```

- **Estrategias de cache**

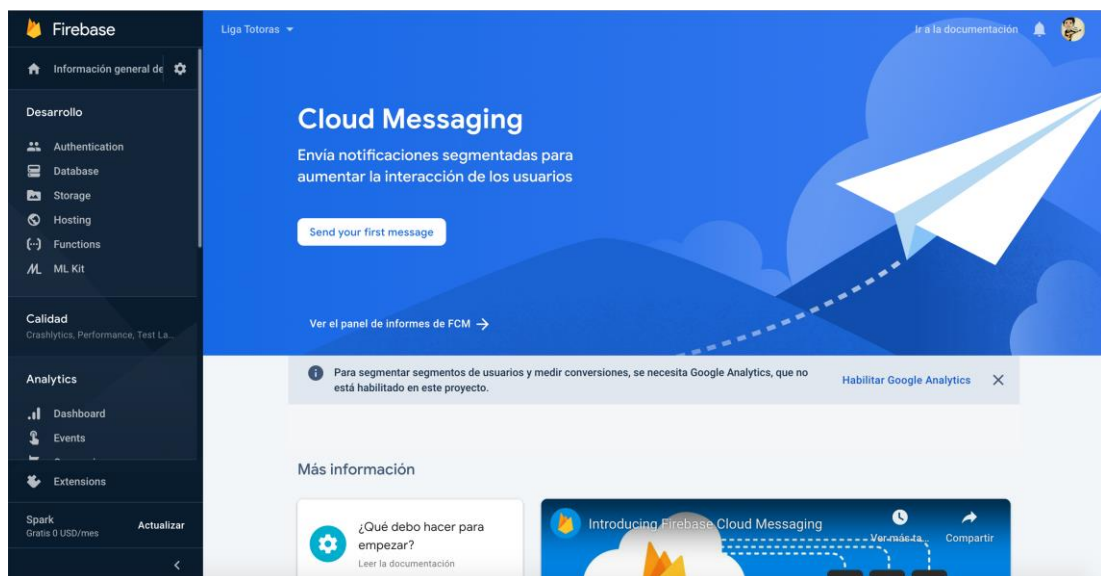
Para las peticiones realizadas a los servicios implementados y desarrollados con Spring, se utiliza la estrategia *networkFirt*. En el registro de la ruta, se establece la dirección de la API, reemplazando el campo “url”

```
workbox.routing.registerRoute(  
  new RegExp("url"),  
  workbox.strategies.networkFist({  
    cacheName: "listPlayer",  
    plugins:[  
      new workbox.expiration.Plugin({  
        maxEntries: 30  
      })  
    ],  
    method: "GET",  
    cacheableResponse: { statuses: [0, 200]}  
  }  
)  
)
```

- **Notificaciones Push en la PWA**

Es importante manejar un aspecto previo al desarrollo de Notificaciones Push, ya que el usuario final recibirá una alerta inicial donde será quien decida dar los permisos para recibir notificaciones en su dispositivo y dependerá del usuario que esta función se habilite

Firestore mediante el servicio llamado *Cloud Messaging*, facilita el desarrollo de Notificaciones Push. Desde la consola de Firestore, acceder al servicio *Cloud Messaging*, como indica la Fig. 60



*Figura 60. Cloud Messaging en la consola de Firestore*

*Elaborado por: Santiago Yaguargos*

En el archivo main se configura las sentencias para el mensaje inicial que se envía al navegador para aceptar o denegar notificaciones push.

```
const messaging = firebase.messaging();
navigator.serviceWorker.register('service-worker.js')
.then((registration) => {
  messaging.useServiceWorker(registration);
//Solicitar permiso de notificaciones
messaging.requestPermission (). then (() => {
  console.log ('Permiso de notificación otorgado');
```

```
// Obtener token
messaging.getToken(). then ((token) => {
  console.log (token)
})
}). catch ((err) => {
  console.log ('No se puede obtener permiso para notificar.', err);
});
```

La configuración para el diseño de notificaciones se realiza en el archivo service worker.

```
messaging.setBackgroundMessageHandler(function (payload) {
  const notificationTitle = 'Nueva Noticia';
  const notificationOptions = {
    body: 'Revisar noticias de LDPT.',
    icon: '/ldpt.png'
  };
  return self.registration.showNotification(notificationTitle,
    notificationOptions);
});
```

- **Fase 5. Pruebas**

Es esencial realizar pruebas de funcionamiento luego de implementar la PWA para determinar errores que pueden alterar el uso del aplicativo y debido al enfoque que tiene hacia un usuario administrador y sus visitantes se a determinado el empleo de pruebas de caja negra que están centradas en los requisitos funcionales del software sin tomar en cuenta la estructura del código ni escenarios internos del software.

Para aplicar estas pruebas, es primordial basarse únicamente a los requerimientos de software y sus funcionalidades, es decir determinar la aceptación a las historias de usuario levantadas antes de construir el aplicativo.



## Pruebas de aceptación - PWA

| Prueba de Aceptación   |                                |
|--|--------------------------------|
| <b>Número: 1</b>   | <b>Historia de Usuario: #2</b> |
| <b>Nombre:</b> Visualizar información de LDPT  |                                |
| <b>Descripción:</b><br>El usuario final podrá acceder a la información básica acerca de la institución como son: historia breve, directorio y organigrama deportivo.   |                                |
| <b>Condiciones de ejecución:</b><br>Abrir la PWA y seleccionar el ítem Nosotros en el Menú.  |                                |
| <b>Entrada:</b><br>Los datos informativos deben estar almacenados en el hosting de Firebase.   |                                |
| <b>Resultado esperado:</b><br>El usuario final accede a información relevante de la institución, que son datos importantes para conocimiento público del personal directa e indirectamente involucrado a Liga Deportiva Parroquial Totoras |                                |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |                                |

*Tabla 58. Prueba de aceptación - Información de LDPT  
Elaborado por: Santiago Yaguargos*

| Prueba de Aceptación   |                                |
|--|--------------------------------|
| <b>Número: 2</b>   | <b>Historia de Usuario: #3</b> |
| <b>Nombre:</b> Información de Clubes Filiales  |                                |
| <b>Descripción:</b><br>El usuario final podrá acceder a la información de los Clubes Filiales a Liga Deportiva Parroquial Totoras de manera interactiva. |                                |
| <b>Condiciones de ejecución:</b><br>Abrir la PWA y seleccionar el logo del club del que el usuario desea obtener información.                            |                                |
| <b>Entrada:</b><br>Los datos del club deben estar almacenados en la base de datos para obtener la información requerida.                                 |                                |
| <b>Resultado esperado:</b>   |                                |

|  |
|--|
| El usuario final accede a información relevante de cada club como son: logo, datos del presidente, año de fundación, un enlace a su red social y finalmente los jugadores habilitados en las diferentes categorías en la temporada actual. |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |

*Tabla 59. Prueba de aceptación - Información de Clubes Filiales  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                |
|---|--------------------------------|
| <b>Número: 3</b>  | <b>Historia de Usuario: #4</b> |
| <b>Nombre:</b> Descargar documentación  |                                |
| <b>Descripción:</b><br>El usuario final podrá acceder a enlaces de descarga sobre documentación necesaria como estatutos, reglamentos, campeonatos, formatos de oficios, transparencia y actas. |                                |
| <b>Condiciones de ejecución:</b><br>Abrir la PWA y seleccionar el ítem Normativa o Actas Oficiales en el Menú.  |                                |
| <b>Entrada:</b><br>El usuario final selecciona el documento necesario.  |                                |
| <b>Resultado esperado:</b><br>Se descarga el respectivo documento.  |                                |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                |

*Tabla 60. Prueba de aceptación - Descargar documentación de LDPT  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                |
|--|--------------------------------|
| <b>Número: 4</b>   | <b>Historia de Usuario: #5</b> |
| <b>Nombre:</b> Inicio de Sesión  |                                |
| <b>Descripción:</b><br>El usuario administrador podrá acceder al aplicativo web mediante una correcta autenticación. |                                |
| <b>Condiciones de ejecución:</b>   |                                |

|  |
|--|
| Abrir la PWA y seleccionar el botón con el ícono de LOGIN  |
| <p><b>Entrada:</b></p> <p>Si el administrador se autentica correctamente, ingresa al sistema, caso contrario se muestran mensajes de alerta.</p> |
| <p><b>Resultado esperado:</b></p> <p>Se mostrará la pantalla de inicio en modo administrador.</p>  |
| <p><b>Evaluación de la prueba:</b></p> <p>Prueba satisfactoria</p>   |

*Tabla 61. Prueba de aceptación - Inicio de Sesión*

*Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                |
|---|--------------------------------|
| <b>Número: 5</b>  | <b>Historia de Usuario: #6</b> |
| <b>Nombre:</b> Ingresar noticias  |                                |
| <b>Descripción:</b>   |                                |
| El administrador podrá ingresar noticias acerca de Liga Deportiva Parroquial Totoras.   |                                |
| <b>Condiciones de ejecución:</b>  |                                |
| El administrador selecciona el ítem Noticias en el Menú de Opciones y se despliega el formulario para ingresar seguido de las últimas noticias. |                                |
| <b>Entrada:</b>   |                                |
| El administrador deberá ingresar los datos en los campos requeridos para ingresar una nueva noticia.  |                                |
| <b>Resultado esperado:</b>  |                                |
| Se mostrará la noticia ingresada en el listado de últimas noticias.   |                                |
| <b>Evaluación de la prueba:</b>   |                                |
| Prueba satisfactoria  |                                |

*Tabla 62. Prueba de aceptación - Ingresar Noticias*

*Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>        |                                |
|------------------------------------|--------------------------------|
| <b>Número: 6</b>                   | <b>Historia de Usuario: #7</b> |
| <b>Nombre:</b> Actualizar noticias |                                |

|   |
|---|
| <b>Descripción:</b>   |
| El administrador podrá actualizar los datos de las noticias acerca de Liga Deportiva Parroquial Totoras.  |
| <b>Condiciones de ejecución:</b>  |
| El administrador selecciona el ítem Noticias en el Menú de Opciones y se despliega el formulario para ingresar seguido de las últimas noticias. |
| <b>Entrada:</b>   |
| El administrador podrá editar los datos de las noticias existentes en el listado.   |
| <b>Resultado esperado:</b>  |
| Se mostrará la noticia actualizada en el listado de últimas noticias.   |
| <b>Evaluación de la prueba:</b>   |
| Prueba satisfactoria  |

*Tabla 63. Prueba de aceptación - Actualizar Noticias  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                |
|--|--------------------------------|
| <b>Número: 7</b>   | <b>Historia de Usuario: #8</b> |
| <b>Nombre:</b> Eliminar noticias   |                                |
| <b>Descripción:</b>  |                                |
| El administrador podrá eliminar noticias acerca de Liga Deportiva Parroquial Totoras.                            |                                |
| <b>Condiciones de ejecución:</b>   |                                |
| En el ítem Noticias desde el Menú de Opciones del administrador, se despliegan las noticias ordenadas por fecha. |                                |
| <b>Entrada:</b>  |                                |
| El administrador podrá eliminar de una en una las noticias existentes en el listado mediante en botón ELIMINAR.  |                                |
| <b>Resultado esperado:</b>   |                                |
| Se mostrará la lista actualizada sin las noticias eliminadas.  |                                |
| <b>Evaluación de la prueba:</b>  |                                |
| Prueba satisfactoria   |                                |

*Tabla 64. Prueba de aceptación - Eliminar Noticias  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                |
|---|--------------------------------|
| <b>Número: 8</b>  | <b>Historia de Usuario: #9</b> |
| <b>Nombre:</b> Visualizar noticias  |                                |
| <b>Descripción:</b><br>El usuario final requiere acceder a un modulo donde se visualicen noticias publicadas de acontecimientos institucionales de mayor importancia. |                                |
| <b>Condiciones de ejecución:</b><br>Abrir la PWA y seleccionar Noticias desde la pantalla inicial o desde el Menú.  |                                |
| <b>Entrada:</b><br>El usuario final accede al módulo de noticias  |                                |
| <b>Resultado esperado:</b><br>Se mostrará la lista de noticias paginadas y con la opción para leer más en cada noticia para ser visualizada en una ventana emergente  |                                |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                |

*Tabla 65. Prueba de aceptación - Visualizar Noticias  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 9</b>  | <b>Historia de Usuario: #10</b> |
| <b>Nombre:</b> Ingresar Clubes  |                                 |
| <b>Descripción:</b><br>El administrador puede ingresar los datos para un nuevo Club o también conocida como Institución Deportiva filial a Liga Deportiva Parroquial Totoras.                 |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de clubes. En la tabla seleccionar el botón con el ícono AGREGAR. |                                 |
| <b>Entrada:</b><br>El administrador deberá ingresar los datos en los campos requeridos para ingresar un nuevo club.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrará el club ingresado en la tabla de clubes   |                                 |
| <b>Evaluación de la prueba:</b>   |                                 |

Prueba satisfactoria

*Tabla 66. Prueba de aceptación - Ingresar Clubes  
Elaborado por: Santiago Yaguargos*

| Prueba de Aceptación   |                                 |
|--|---------------------------------|
| <b>Número: 10</b>  | <b>Historia de Usuario: #11</b> |
| <b>Nombre:</b> Actualizar Clubes   |                                 |
| <b>Descripción:</b><br>El administrador puede actualizar los datos de un Club o también conocido como Institución Deportiva filial a LDPT.   |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de clubes. En la tabla seleccionar el botón con el ícono EDITAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá editar los datos de los clubes existentes en la tabla.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrará los datos del club actualizados en la tabla de clubes.   |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |                                 |

*Tabla 67. Prueba de aceptación - Actualizar Clubes  
Elaborado por: Santiago Yaguargos*

| Prueba de Aceptación   |                                 |
|--|---------------------------------|
| <b>Número: 11</b>  | <b>Historia de Usuario: #12</b> |
| <b>Nombre:</b> Eliminar clubes   |                                 |
| <b>Descripción:</b><br>El administrador puede eliminar un Club o también conocido como Institución Deportiva filial a LDPT.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de clubes. En la tabla seleccionar el botón con el ícono ELIMINAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá eliminar un club existente en la tabla.  |                                 |

|   |
|---|
| <b>Resultado esperado:</b><br>Se mostrará la tabla actualizada sin los clubes eliminados. |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria                                   |

*Tabla 68. Prueba de aceptación - Eliminar Clubes  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 12</b>   | <b>Historia de Usuario: #13</b> |
| <b>Nombre:</b> Visualizar clubes  |                                 |
| <b>Descripción:</b><br>El administrador podrá visualizar el detalle de un club filial de Liga Deportiva Parroquial Totoras seleccionado.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de clubes. En la tabla seleccionar el botón con el ícono VER. |                                 |
| <b>Entrada:</b><br>El administrador podrá seleccionar un club existente en la tabla.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrarán los datos del club filial a Liga Deportiva Parroquial Totoras seleccionado de la tabla.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 69. Prueba de aceptación - Visualizar Clubes  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 13</b>   | <b>Historia de Usuario: #14</b> |
| <b>Nombre:</b> Buscar clubes  |                                 |
| <b>Descripción:</b><br>El administrador podrá buscar por el nombre de un club filial de Liga Deportiva Parroquial Totoras seleccionado. |                                 |
| <b>Condiciones de ejecución:</b>  |                                 |

|   |
|---|
| El administrador selecciona el ítem Club en el Menú de Opciones, se despliega la tabla de clubes y en el encabezado el campo de búsqueda. |
| <b>Entrada:</b><br>El administrador digita el nombre de un club.  |
| <b>Resultado esperado:</b><br>Se filtrará los datos del club filial a Liga Deportiva Parroquial Totoras a buscar de la tabla.             |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |

*Tabla 70. Prueba de aceptación - Buscar Clubes  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 14</b>  | <b>Historia de Usuario: #15</b> |
| <b>Nombre:</b> Ingresar torneos  |                                 |
| <b>Descripción:</b><br>El administrador puede ingresar un torneo también conocido como campeonato o temporada.   |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de torneos. En la tabla seleccionar el botón con el ícono AGREGAR. |                                 |
| <b>Entrada:</b><br>El administrador deberá ingresar los datos en los campos requeridos para ingresar un nuevo torneo.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrará el club ingresado en la tabla de torneos.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |                                 |

*Tabla 71. Prueba de aceptación - Ingresar torneos  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b> |                                 |
|-----------------------------|---------------------------------|
| <b>Número: 15</b>           | <b>Historia de Usuario: #16</b> |



|   |
|---|
| <b>Nombre:</b> Actualizar torneos   |
| <b>Descripción:</b><br>El administrador puede actualizar los datos de un torneo existente, también conocido como campeonato o temporada.  |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de torneos. En la tabla seleccionar el botón con el ícono EDITAR. |
| <b>Entrada:</b><br>El administrador podrá editar los datos de los torneos existentes en la tabla.   |
| <b>Resultado esperado:</b><br>Se mostrará los datos del torneo actualizados en la tabla.  |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |

*Tabla 72. Prueba de aceptación - Actualizar torneos  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 16</b>   | <b>Historia de Usuario: #17</b> |
| <b>Nombre:</b> Eliminar torneos   |                                 |
| <b>Descripción:</b><br>El administrador puede eliminar un torneo existente, también conocido como campeonato o temporada.   |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de torneos. En la tabla seleccionar el botón con el ícono ELIMINAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá eliminar un torneo existente en la tabla.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrará la tabla actualizada sin los torneos eliminados.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 73. Prueba de aceptación - Eliminar torneos  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 17</b>  | <b>Historia de Usuario: #18</b> |
| <b>Nombre:</b> Visualizar torneos  |                                 |
| <b>Descripción:</b><br>El administrador puede visualizar un torneo existente, también conocido como campeonato o temporada.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de torneos. En la tabla seleccionar el botón con el ícono VER. |                                 |
| <b>Entrada:</b><br>El administrador podrá visualizar un torneo existente en la tabla.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrarán los datos de un torneo seleccionado de la tabla.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |                                 |

*Tabla 74. Prueba de aceptación - Visualizar torneos  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 18</b>  | <b>Historia de Usuario: #19</b> |
| <b>Nombre:</b> Buscar clubes   |                                 |
| <b>Descripción:</b><br>El administrador puede buscar un torneo en el listado por el año.   |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones, se despliega la tabla de torneos y en el encabezado el campo de búsqueda. |                                 |
| <b>Entrada:</b><br>El administrador digita el nombre de un torneo.   |                                 |
| <b>Resultado esperado:</b><br>Se filtrará los datos del torneo a buscar de la tabla.   |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |                                 |

*Tabla 75. Prueba de aceptación - Buscar torneos*

Elaborado por: Santiago Yaguargos

| Prueba de Aceptación  |                                 |
|---|---------------------------------|
| <b>Número: 19</b>   | <b>Historia de Usuario: #20</b> |
| <b>Nombre:</b> Ingresar categorías  |                                 |
| <b>Descripción:</b><br>El administrador podrá ingresar categorías a las que pertenecerán los diferentes equipos.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Club en el Menú de Opciones y se despliega la tabla de categorías. En la tabla seleccionar el botón con el ícono AGREGAR. |                                 |
| <b>Entrada:</b><br>El administrador deberá ingresar los datos en los campos requeridos para ingresar una nueva categoría.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrará la categoría ingresada en la tabla de torneos.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

Tabla 76. Prueba de aceptación - Ingresar categorías  
Elaborado por: Santiago Yaguargos

| Prueba de Aceptación  |                                 |
|---|---------------------------------|
| <b>Número: 20</b>   | <b>Historia de Usuario: #21</b> |
| <b>Nombre:</b> Actualizar categorías  |                                 |
| <b>Descripción:</b><br>El administrador puede actualizar los datos de una categoría existente.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Categorías en el Menú de Opciones y se despliega la tabla de datos. En la tabla seleccionar el botón con el ícono EDITAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá editar los datos de las categorías existentes en la tabla.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrará los datos de las categorías actualizadas en la tabla.   |                                 |

**Evaluación de la prueba:**

Prueba satisfactoria

*Tabla 77. Prueba de aceptación - Actualizar categorías  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 21</b>  | <b>Historia de Usuario: #22</b> |
| <b>Nombre:</b> Eliminar categorías   |                                 |
| <b>Descripción:</b><br>El administrador podrá actualizar datos de las determinadas categorías a las que pertenecen los diferentes equipos.   |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Categorías en el Menú de Opciones y se despliega la tabla. En la tabla seleccionar el botón con el ícono ELIMINAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá eliminar una categoría existente en la tabla.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrará la tabla actualizada sin las categorías eliminados.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |                                 |

*Tabla 78. Prueba de aceptación - Eliminar categorías  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 22</b>  | <b>Historia de Usuario: #23</b> |
| <b>Nombre:</b> Visualizar categorías   |                                 |
| <b>Descripción:</b><br>El administrador podrá visualizar las categorías existentes.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Categorías en el Menú de Opciones y se despliega la tabla respectiva. En la tabla seleccionar el botón con el ícono VER. |                                 |
| <b>Entrada:</b><br>El administrador podrá visualizar una categoría existente en la tabla.  |                                 |

|   |
|---|
| <b>Resultado esperado:</b><br>Se mostrarán los datos de una categoría seleccionado de la tabla. |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |

*Tabla 79. Prueba de aceptación - Visualizar categorías  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 23</b>   | <b>Historia de Usuario: #24</b> |
| <b>Nombre:</b> Ingresar jugadores   |                                 |
| <b>Descripción:</b><br>El administrador podrá ingresar datos de nuevos jugadores, estos tendrán que pertenecer a un Club o también conocida como Institución Deportiva. |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Jugadores en el Menú de Opciones. En la tabla, seleccionar el botón con el ícono AGREGAR.       |                                 |
| <b>Entrada:</b><br>El administrador deberá ingresar los datos en los campos requeridos para ingresar un nuevo jugador.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrará el jugador ingresado en la tabla de torneos.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 80. Prueba de aceptación - Ingresar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 24</b>  | <b>Historia de Usuario: #25</b> |
| <b>Nombre:</b> Actualizar jugadores  |                                 |
| <b>Descripción:</b><br>El administrador puede actualizar los datos de un jugador registrado. |                                 |
| <b>Condiciones de ejecución:</b>   |                                 |

|  |
|--|
| El administrador selecciona el ítem Jugadores en el Menú de Opciones. En la tabla, seleccionar el botón con el ícono EDITAR. |
| <b>Entrada:</b><br>El administrador podrá editar los datos de los jugadores existentes en la tabla.                          |
| <b>Resultado esperado:</b><br>Se mostrará los datos de los jugadores actualizados en la tabla.                               |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria  |

*Tabla 81. Prueba de aceptación - Actualizar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 25</b>   | <b>Historia de Usuario: #26</b> |
| <b>Nombre:</b> Eliminar jugadores   |                                 |
| <b>Descripción:</b><br>El administrador podrá eliminar jugadores registrados.   |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Jugadores en el Menú de Opciones. En la tabla seleccionar el botón con el ícono ELIMINAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá eliminar un jugador existente en la tabla.  |                                 |
| <b>Resultado esperado:</b><br>Se mostrará la tabla actualizada sin los jugadores eliminados.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 82. Prueba de aceptación - Eliminar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 26</b>  | <b>Historia de Usuario: #27</b> |
| <b>Nombre:</b> Visualizar jugadores  |                                 |
| <b>Descripción:</b><br>El administrador podrá visualizar los jugadores existentes. |                                 |

|  |
|--|
| <b>Condiciones de ejecución:</b>   |
| El administrador selecciona el ítem Jugadores en el Menú de Opciones. En la tabla seleccionar el botón con el ícono VER. |
| <b>Entrada:</b>  |
| El administrador podrá visualizar un jugador existente en la tabla.  |
| <b>Resultado esperado:</b>   |
| Se mostrarán los datos de un jugador seleccionado de la tabla.   |
| <b>Evaluación de la prueba:</b>  |
| Prueba satisfactoria   |

*Tabla 83. Prueba de aceptación - Visualizar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 27</b>   | <b>Historia de Usuario: #28</b> |
| <b>Nombre:</b> Buscar jugadores   |                                 |
| <b>Descripción:</b>   |                                 |
| El administrador podrá realizar búsquedas de los jugadores registrados ingresando cédula o apellidos.   |                                 |
| <b>Condiciones de ejecución:</b>  |                                 |
| El administrador selecciona el ítem Jugadores en el Menú de Opciones, se despliega la tabla de jugadores y en el encabezado el campo de búsqueda. |                                 |
| <b>Entrada:</b>   |                                 |
| El administrador digita el número de cédula o los apellidos de un jugador.  |                                 |
| <b>Resultado esperado:</b>  |                                 |
| Se filtrará los datos de jugador o jugadores a buscar de la tabla.  |                                 |
| <b>Evaluación de la prueba:</b>   |                                 |
| Prueba satisfactoria  |                                 |

*Tabla 84. Prueba de aceptación - Buscar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b> |                                 |
|-----------------------------|---------------------------------|
| <b>Número: 28</b>           | <b>Historia de Usuario: #29</b> |

|   |
|---|
| <b>Nombre:</b> Ingresar equipos   |
| <b>Descripción:</b><br>El administrador podrá ingresar nuevos equipos en su respectiva categoría y perteneciente a un Club o también llamada Institución Deportiva. |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Equipos en el Menú de Opciones. En la tabla, seleccionar el botón con el ícono AGREGAR.     |
| <b>Entrada:</b><br>El administrador deberá ingresar los datos en los campos requeridos para ingresar un nuevo equipo.   |
| <b>Resultado esperado:</b><br>Se mostrará el equipo ingresado en la tabla.  |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |

*Tabla 85. Prueba de aceptación - Ingresar equipos  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 29</b>   | <b>Historia de Usuario: #30</b> |
| <b>Nombre:</b> Actualizar equipos   |                                 |
| <b>Descripción:</b><br>El administrador podrá actualizar datos de los equipos registrados en su respectiva categoría y perteneciente a un Club o también llamada Institución Deportiva. |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Equipos en el Menú de Opciones. En la tabla desplegada, seleccionar el botón con el ícono EDITAR.               |                                 |
| <b>Entrada:</b><br>El administrador podrá editar los datos de los equipos existentes en la tabla.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrará los datos de los equipos actualizados en la tabla.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 86. Prueba de aceptación - Actualizar equipos*



Elaborado por: Santiago Yaguargos

| Prueba de Aceptación  |                                 |
|---|---------------------------------|
| <b>Número: 30</b>   | <b>Historia de Usuario: #31</b> |
| <b>Nombre:</b> Eliminar equipos   |                                 |
| <b>Descripción:</b><br>El administrador podrá eliminar equipos registrados y pertenecientes a un Club o también llamada Institución Deportiva.                              |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Equipos en el Menú de Opciones. En la tabla desplegada, seleccionar el botón con el ícono ELIMINAR. |                                 |
| <b>Entrada:</b><br>El administrador podrá eliminar un equipo existente en la tabla.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrará la tabla actualizada sin los equipos eliminados.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

Tabla 87. Prueba de aceptación - Eliminar equipos  
Elaborado por: Santiago Yaguargos

| Prueba de Aceptación   |                                 |
|--|---------------------------------|
| <b>Número: 31</b>  | <b>Historia de Usuario: #32</b> |
| <b>Nombre:</b> Visualizar equipos  |                                 |
| <b>Descripción:</b><br>El administrador podrá visualizar el listado de equipos en su respectiva categoría y perteneciente a un Club o también llamada Institución Deportiva. |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Jugadores en el Menú de Opciones. En la tabla seleccionar el botón con el ícono VER.                 |                                 |
| <b>Entrada:</b><br>El administrador podrá visualizar un jugador existente en la tabla.   |                                 |
| <b>Resultado esperado:</b><br>Se mostrarán los datos de un jugador seleccionado de la tabla.   |                                 |

|   |
|---|
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria |
|---|

*Tabla 88. Prueba de aceptación - Visualizar equipos  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 32</b>   | <b>Historia de Usuario: #33</b> |
| <b>Nombre:</b> Buscar equipos   |                                 |
| <b>Descripción:</b><br>El administrador podrá realizar búsquedas de equipos por su nombre, categoría o club.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Equipos en el Menú de Opciones, se despliega la tabla de equipos y en el encabezado el campo de búsqueda. |                                 |
| <b>Entrada:</b><br>El administrador digita el nombre, categoría o club de un equipo.  |                                 |
| <b>Resultado esperado:</b><br>Se filtrará los datos del equipo o equipos a buscar de la tabla.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 89. Prueba de aceptación - Buscar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 33</b>   | <b>Historia de Usuario: #34</b> |
| <b>Nombre:</b> Habilitar equipos  |                                 |
| <b>Descripción:</b><br>El administrador podrá habilitar equipos que participaran en un torneo existente.  |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Habilitar Equipos en el Menú de Opciones, se despliega una lista con los equipos. |                                 |
| <b>Entrada:</b><br>El administrador selecciona los equipos a habilitar.   |                                 |

|  |
|--|
| <b>Resultado esperado:</b><br>Se mostrará una nueva lista con los equipos habilitados. |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria                                |

*Tabla 90. Prueba de aceptación - Habilitar equipos*

*Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 34</b>   | <b>Historia de Usuario: #35</b> |
| <b>Nombre:</b> Inhabilitar equipos  |                                 |
| <b>Descripción:</b><br>El administrador podrá inhabilitar a los equipos que no participaran en un determinado torneo.                                     |                                 |
| <b>Condiciones de ejecución:</b><br>El administrador selecciona el ítem Habilitar Equipos en el Menú de Opciones, se despliega una lista con los equipos. |                                 |
| <b>Entrada:</b><br>El administrador selecciona los equipos a inhabilitar.   |                                 |
| <b>Resultado esperado:</b><br>Se actualiza la lista con los equipos habilitados.  |                                 |
| <b>Evaluación de la prueba:</b><br>Prueba satisfactoria   |                                 |

*Tabla 91. Prueba de aceptación - Inhabilitar equipos*

*Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 35</b>  | <b>Historia de Usuario: #36</b> |
| <b>Nombre:</b> Visualizar equipos habilitados  |                                 |
| <b>Descripción:</b><br>El usuario final podrá visualizar los equipos que participaran actualmente en un torneo.        |                                 |
| <b>Condiciones de ejecución:</b><br>El usuario final abre la PWA y seleccionar el logo del club a obtener información. |                                 |

|  |
|--|
| <b>Entrada:</b>  |
| Los datos deben estar almacenados en la base de datos para obtener la información requerida. |
| <b>Resultado esperado:</b>   |
| Se visualiza la página del club con los equipos habilitados.                                 |
| <b>Evaluación de la prueba:</b>  |
| Prueba satisfactoria   |

*Tabla 92. Prueba de aceptación - Visualizar equipos habilitados  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>   |                                 |
|---|---------------------------------|
| <b>Número: 36</b>   | <b>Historia de Usuario: #37</b> |
| <b>Nombre:</b> Habilitar jugadores  |                                 |
| <b>Descripción:</b>   |                                 |
| El administrador podrá habilitar jugadores en los equipos que están habilitados en una determinada temporada, ingresando su cédula.                                   |                                 |
| <b>Condiciones de ejecución:</b>  |                                 |
| El administrador selecciona el ítem Habilitar Jugadores en el Menú de Opciones, se despliega una lista con equipos habilitados y un formulario para añadir jugadores. |                                 |
| <b>Entrada:</b>   |                                 |
| El administrador ingresa la cedula del jugador y selecciona HABILITAR.  |                                 |
| <b>Resultado esperado:</b>  |                                 |
| Se mostrará una tabla con los jugadores en los equipos habilitados.   |                                 |
| <b>Evaluación de la prueba:</b>   |                                 |
| Prueba satisfactoria  |                                 |

*Tabla 93. Prueba de aceptación - Habilitar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 37</b>  | <b>Historia de Usuario: #38</b> |
| <b>Nombre:</b> Inhabilitar jugadores   |                                 |
| <b>Descripción:</b>  |                                 |
| El administrador podrá inhabilitar a los jugadores de los equipos habilitados. |                                 |

|   |
|---|
| <b>Condiciones de ejecución:</b>  |
| El administrador selecciona el ítem Habilitar Jugadores en el Menú de Opciones, se despliega una lista con equipos habilitados y un formulario para añadir jugadores. |
| <b>Entrada:</b>   |
| El administrador busca un jugador habilitado y selecciona INHABILITAR.  |
| <b>Resultado esperado:</b>  |
| Se actualiza la tabla con los jugadores habilitados.  |
| <b>Evaluación de la prueba:</b>   |
| Prueba satisfactoria  |

*Tabla 94. Prueba de aceptación - Inhabilitar jugadores  
Elaborado por: Santiago Yaguargos*

| <b>Prueba de Aceptación</b>  |                                 |
|--|---------------------------------|
| <b>Número: 38</b>  | <b>Historia de Usuario: #39</b> |
| <b>Nombre:</b> Visualizar jugadores habilitados  |                                 |
| <b>Descripción:</b>  |                                 |
| El usuario final podrá visualizar los jugadores habilitados que participaran en un determinado equipo según el torneo. |                                 |
| <b>Condiciones de ejecución:</b>   |                                 |
| El usuario final abre la PWA y seleccionar el logo del club a obtener información.                                     |                                 |
| <b>Entrada:</b>  |                                 |
| Los datos deben estar almacenados en la base de datos para obtener la información requerida.                           |                                 |
| <b>Resultado esperado:</b>   |                                 |
| Se visualiza la página del club con los equipos habilitados, seguido de los jugadores habilitados.                     |                                 |
| <b>Evaluación de la prueba:</b>  |                                 |
| Prueba satisfactoria   |                                 |

*Tabla 95. Prueba de aceptación - Visualizar jugadores habilitados  
Elaborado por: Santiago Yaguargos*

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

#### 1.2.1. Conclusiones

- La librería de maquetación Vuetify se adapta correctamente a las necesidades del usuario, manteniendo una estructura seria en su entorno con un alto nivel en la facilidad de uso, que permite a los involucrados directa e indirectamente con la institución, familiarizarse rápidamente con el aplicativo.
- El framework de desarrollo Vue se mantiene en constante actualización y facilita al desarrollador migrar en diferentes versiones debido a la facilidad que brinda en curva de aprendizaje a comparación de otros frameworks populares como Angular y React.
- La tecnología Firebase y el uso de sus servicios, ayudan al desarrollador crear software de alta calidad en menor tiempo, debido a la facilidad en las configuraciones de los módulos de autenticación, firestore, cloud messaging, storage y hosting.
- El uso de Firebase y su base de datos no relacional en el almacenamiento de datos para los procesos de información, es altamente escalable en el aplicativo, es decir, sin importar el crecimiento de colecciones en Firestore, el rendimiento no sufre alteraciones.
- El uso de Firebase y su servicio de autenticación optimizan los tiempos de desarrollo, de esta manera se mantiene la expectativa al habilitar diferentes proveedores, con un alto nivel de seguridad.
- El interceptor creado en Spring caracteriza al sistema con un alto nivel de seguridad, ya que las peticiones a los servicios web que se mantienen alojados en una IP pública, requieren un token generado desde la autenticación de Firebase.
- Workbox facilita la estructura para las estrategias de cache en la construcción de la PWA y las configuraciones de los archivos, entre ellos el service worker.

### 1.2.2. Recomendaciones

- Se recomienda utilizar las diferentes propiedades que brindan los componentes de Vuetify para aumentar detalles en facilidad de uso del aplicativo.
- Se recomienda investigar acerca de las ventajas que conllevan al migrar a una siguiente versión de Vue.
- Se recomienda investigar otros servicios de Firebase que pueden ser implementados en este tipo de aplicativos, debido a la optimización en tiempos de desarrollo y facilidad de adaptación.
- Se recomienda aprovechar la característica de modularidad que ofrece Vue, para la creación de futuros módulos en el aplicativo.
- Se recomienda utilizar Firestore cuando se trate del almacenamiento de datos en nuevos procesos informativos que se requieran automatizar a futuro.
- Se recomienda utilizar Firebase y la configuración de Notificaciones Push en el aplicativo, gracias a las características que brinda la tecnología en el desarrollo de Aplicaciones Web Progresivas.
- Se recomienda investigar con mayor énfasis el uso de la librería workbox para la configuración de las estrategias de cache en las PWA's y de esa manera mejorar la experiencia del usuario al usar este tipo de tecnologías.

## BIBLIOGRAFÍA

- [1] B. Cardenas, «SISTEMA AUTOMATIZADO PARA EL REGISTRO Y SANCIONES DE LOS JUGADORES EN LA LIGA DEPORTIVA PARROQUIAL HUACHI GRANDE,» Ambato, 2012.
- [2] B. Moreno, «SISTEMA AUTOMATIZADO PARA EL CONTROL DE LA FIABILIDAD DE LA INFORMACIÓN EN LA GESTIÓN DE LOS CAMPEONATOS DE FÚTBOL EN LIGA DEPORTIVA CANTONAL PELILEO,» Ambato, 2015.
- [3] N. Miranda y G. Torres, «IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PROGRESIVA (PWA), PARA LA IDENTIFICACIÓN Y NOTIFICACIÓN DE MASCOTAS EXTRAVIADAS USANDO PLACAS CON CÓDIGO DE RESPUESTA RÁPIDA (QR) EN COLLARES,» Guayaquil, 2018.
- [4] R. Correa Delgado, «REGLAMENTO GENERAL A LA LEY DEL DEPORTE, EDUCACIÓN FÍSICA Y RECREACIÓN,» Quito, 2011.
- [5] V. Mejia Heano, «La informatica y su contribucion a la automatizacion de procesos,» *LUPA EMPRESARIAL BUSINESS SCHOOL CEIPA*, p. 1, 2009.
- [6] G. Areitio y A. Areitio, «Información, Informática e Internet: del ordenador personal a la Empresa 2.0,» Visión Libros, 2009.
- [7] F. Luna, JavaScript, Aprende a programar en el lenguaje de la Web, Buenos Aires, 2019.
- [8] Node.js Foundation, «NodeJS,» [En línea]. Available: <https://nodejs.org/es/>.
- [9] Y. Salas, Aprendiendo NodeJS, Kindle Edition, 2018.
- [10] G. Martinez, G. Camacho y D. Biancha, DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO DE APLICACIONES, Pereira: Scientia Et Technica, 2010.
- [11] J. Dongil, DESARROLLA APLICACIONES CON VUEJS, Madrid, 2018.
- [12] G. Figueroa, Aplicando arquitectura single page application para mejorar la experiencia del usuario en un sistema de administración de recursos educativos., 2017.



- [13] Vue.js, «El Framework JavaScript Progresivo,» 2017. [En línea]. Available: <https://es.vuejs.org>.
- [14] Vuetify, LLC, «Material Design Component Framework,» [En línea]. Available: <https://vuetifyjs.com>.
- [15] Vuelidate.js, «Vuelidate 0.75,» [En línea]. Available: <https://vuelidate.js.org/>.
- [16] Upplication, «Progressive Web Apps La nueva manera de entender las Apps,» [En línea]. Available: [https://www.upplication.com/images/pwa/upplication\\_PWA\\_2018\\_es-ES.pdf](https://www.upplication.com/images/pwa/upplication_PWA_2018_es-ES.pdf).
- [17] MDN web docs, «Web App Manifest,» 2020. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/Manifest>.
- [18] Amazon Web Services, Inc., «¿Qué es NoSql?,» 2020. [En línea]. Available: <https://aws.amazon.com/es/nosql/>.
- [19] A. Barrera, «NextU LATAM Inc.,» 2020. [En línea]. Available: <https://www.nextu.com/blog/bases-datos-nosql/>.
- [20] S. Sucino, Aplicación Android para compartir coche basada en la tecnología Firebase, Sevilla, 2019.
- [21] Google Inc., Firebase, San Francisco, 2020.
- [22] The PostgreSQL Global Development Group, PostgreSQL 10.12 Documentation, California, 1996-2020.
- [23] D. Sandoval, SISTEMA DE LOTERÍA DE APUESTAS DEPORTIVAS EN EL FÚTBOL CON SPRING FRAMEWORK PARA LA EMPRESA ALQUIMIASOFT S.A., Ambato, 2020.
- [24] VMare, Inc., Spring Boot 2.2.5, 2020.
- [25] R. Pressman, Ingeniería de Software Un Enfoque Practico.
- [26] M. Gaunt, «Introducción a los Services Workers».
- [27] E. Lanza, Manage Street parking payments using Progressive Web Application (PWA), Cantabria, 2019.

## ANEXOS

**Anexo 1. Encuesta para determinar requerimientos y la factibilidad para el desarrollo una PWA para Liga Deportiva Parroquial Totoras.**

**UNIVERSIDAD TÉCNICA DE AMBATO  
FACULTAD DE TECNOLOGÍAS DE LA INFORMACIÓN,  
TELECOMUNICACIONES E INDUSTRIAL  
CARRERA DE INGENIERÍA EN SISTEMA COMPUTACIONALES E  
INFORMÁTICOS**

**Encuesta dirigida a la población involucrada en los procesos que maneja Liga  
Deportiva Parroquial Totoras**

### **OBJETIVO:**

Determinar los requerimientos y la factibilidad para el desarrollo de una Aplicación Web Progresiva que automatice procesos de gestión e información.

### **Instrucciones**

- Lea detenidamente cada pregunta y responda con toda libertad y sinceridad
- Encierre en círculo la(s) respuesta(s) que usted considere apropiada(s).

## CUESTIONARIO

- 1. ¿Cuántas horas al día utiliza algún servicio de internet (redes sociales, páginas informativas, entre otros)? (1 respuesta)**
  - a) Ninguna
  - b) 1 hora
  - c) 3 horas
  - d) 5 horas
  - e) Más de 5 horas
  
- 2. ¿Qué dispositivos utiliza usualmente para acceder a un servicio con internet? (1 o más respuestas)**
  - a) Teléfono móvil
  - b) Computadora portátil o de escritorio
  - c) Tablet
  - d) Otro. Especifique: \_\_\_\_\_

- 3. ¿Usted considera que se ha llevado una correcta administración en el almacenamiento de datos de los clubes, equipos y jugadores? (1 respuesta)**
- a) Sí
  - b) No
- 4. ¿Qué tan importante es para usted la transparencia de datos y la información que Liga Deportiva Parroquial Totoras brinda, previo y durante un campeonato? (1 respuesta)**
- a) Muy importante
  - b) Importante
  - c) Poco importante
  - d) No es importante
- 5. ¿Desearía tener acceso a la información de los clubes, equipos y jugadores mediante una aplicación en tiempo real a través de su dispositivo móvil? (1 respuesta)**
- a) Si
  - b) No
- 6. ¿Desearía recibir noticias y toda la información que maneja Liga Deportiva Parroquial Totoras a través de notificaciones en su dispositivo móvil? (1 respuesta)**
- a) Si
  - b) No

**Gracias por su colaboración**

## Anexo 2. Manual de instalación de la PWA

El presente manual tiene como objetivo explicar la instalación y el uso de la PWA en los diferentes dispositivos que la tecnología soporta.

De preferencia, en los dispositivos con los Sistemas Operativos Android, Windows y MacOS utilizar Google Chrome, mientras que en los dispositivos con el Sistema Operativo iOS utilizar el navegador Safari.

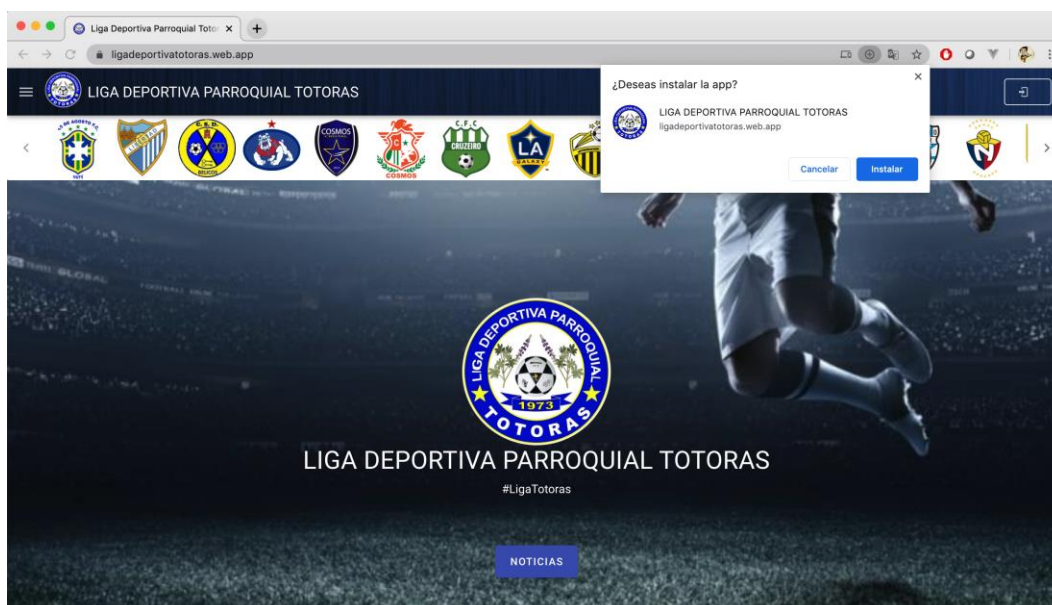
### Desde Google Chrome para ordenadores


#### 1. Acceder a la PWA mediante su url

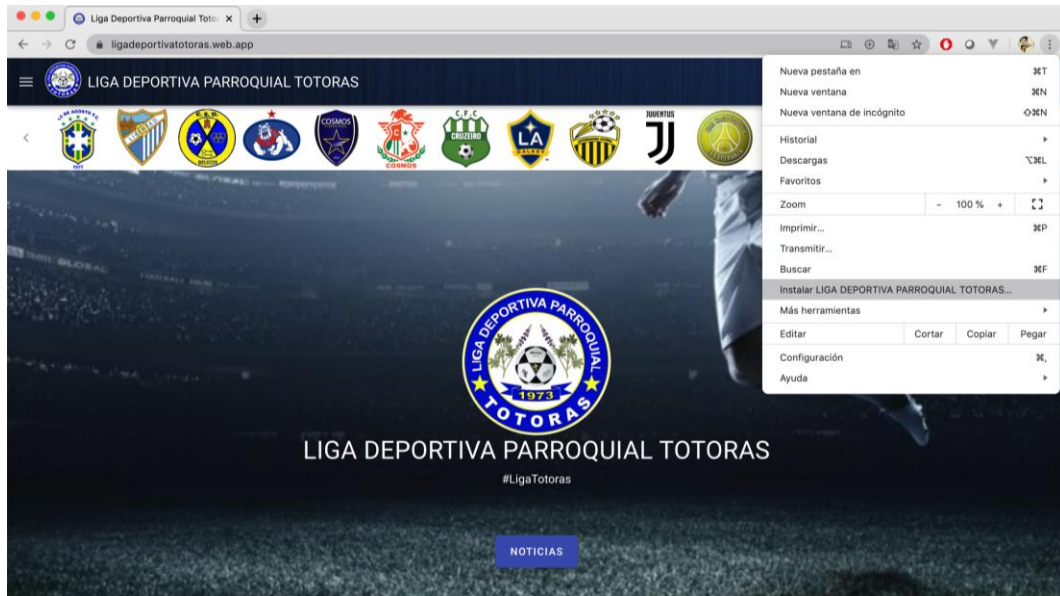
En el navegador desktop acceder a la url: <https://ligadeportivatotoras.web.app>

#### 2. Instalar la PWA

Modo 1: Al seleccionar en el icono + ubicado en la parte derecha de la barra de búsqueda url del navegador, se mostrará una notificación y seguidamente dar click en Instalar.

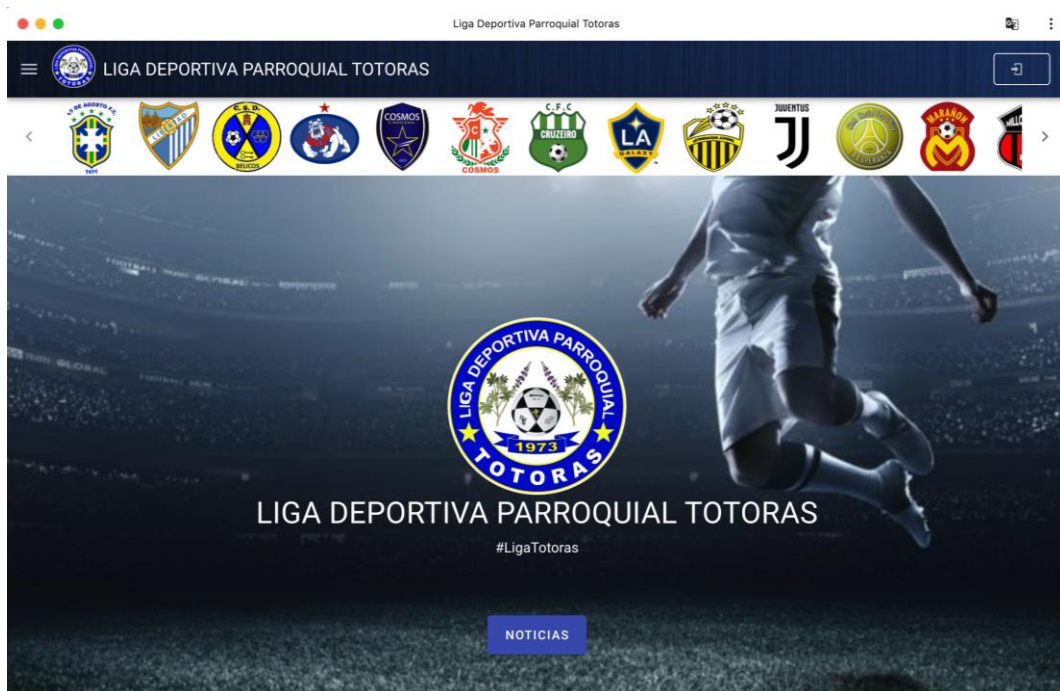


Modo 2: Seleccionar el icono  ubicado en la parte superior derecha del navegador, se mostrarán las opciones y dar click en “Instalar LIGA DEPORTIVA PARROQUIAL TOTORAS”.



### 3. Abrir la PWA

La PWA se instalará en el ordenador y se crea el respectivo icono con el logo. Acceder desde el Dashboard o el Escritorio según sea el Sistema Operativo.




## Desde Google Chrome para dispositivos Android

### 1. Acceder a la PWA mediante su url

En el navegador desktop acceder a la url: <https://ligadeportivatotoras.web.app>

### 2. Instalar la PWA

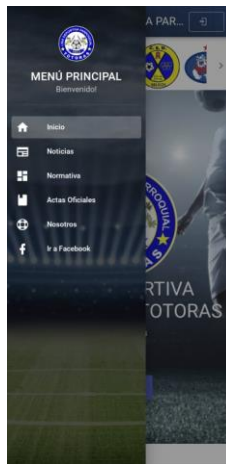
Seleccionar el ícono  ubicado en la parte superior derecha del navegador, se mostrarán las opciones y dar click en “Agregar a la pantalla inicial”.




Se mostrará un mensaje para confirmar la acción. Seleccionar “Agregar”.

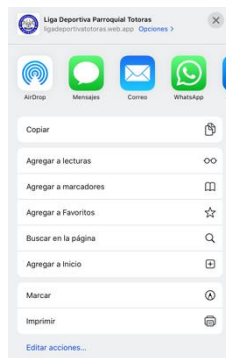


La PWA se instalará en el dispositivo Android y se crea el respectivo icono con el logo. Acceder desde la Pantalla inicial.



### Desde Safari para dispositivos iOS

En el centro de la parte inferior del navegador, seleccionar el botón Compartir , se mostrarán las opciones y seleccionar en “Agregar a Inicio”.



### 3. Renombrar la PWA

Se puede renombrar la PWA o mantener por defecto “LDPT”. Este nombre se refleja desde la Pantalla Inicial.



#### 4. Abrir la PWA

La PWA se instalará en el dispositivo iOS y se crea el respectivo icono con el logo. Acceder desde la Pantalla Inicial.





### **Anexo 3. Manual de uso del sistema PWA de Liga Deportiva Parroquial Totoras Versión 1.0.2. para el Administrador.**

El presente manual tiene como objetivo explicar el uso del sistema al administrador.

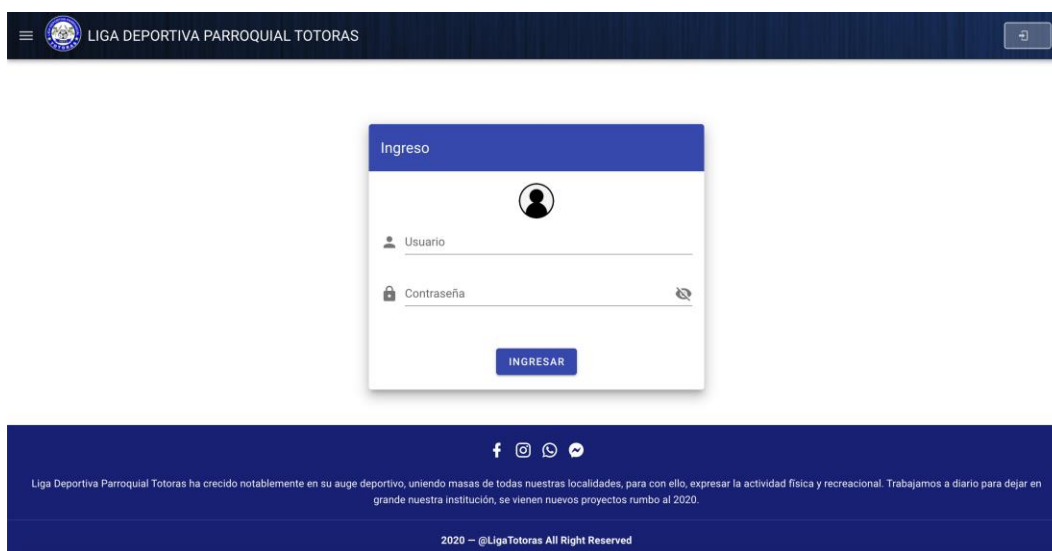
#### **1. Acceder a la PWA mediante su url**

De preferencia, en los dispositivos con los Sistemas Operativos Android, Windows y MacOS utilizar Google Chrome, mientras que en los dispositivos con el Sistema Operativo iOS utilizar el navegador Safari.

Url: <https://ligadeportivatotoras.web.app>

#### **2. Ingresar al sistema**

En la parte superior derecha se encuentra el icono de Login. Al dar clic en el botón se muestra la pantalla con el formulario de ingreso, donde se procede a la autenticación mediante correo y contraseña.



#### **3. Pantalla inicial del Usuario Administrador del Sistema**

En la pantalla inicial del administrador, se encuentra el Menú con los respectivos ítems que permiten acceder a la Administración de Noticias, Clubes, Jugadores, Equipos, Torneos, Categorías y Habilitaciones.

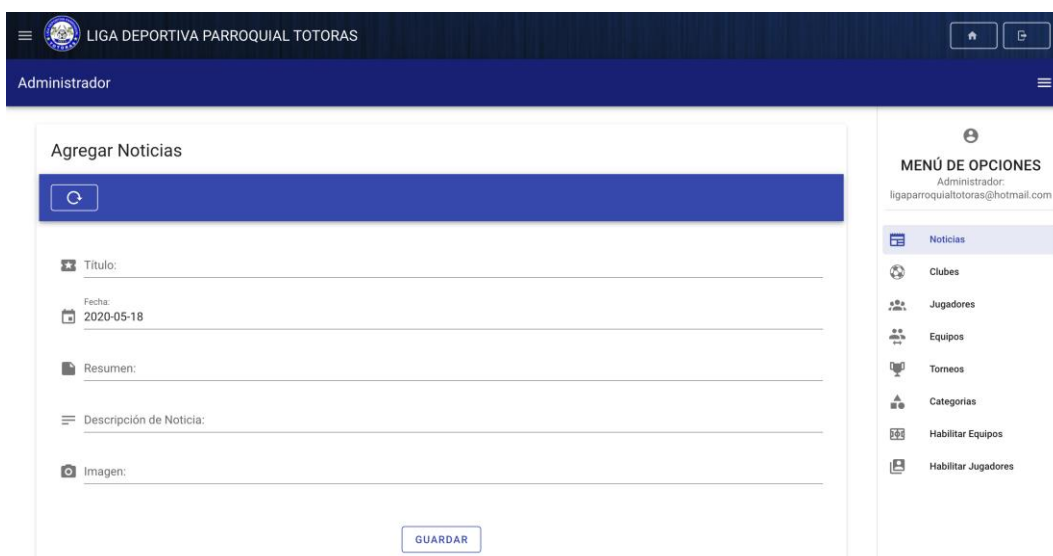


#### 4. Administrar Noticias

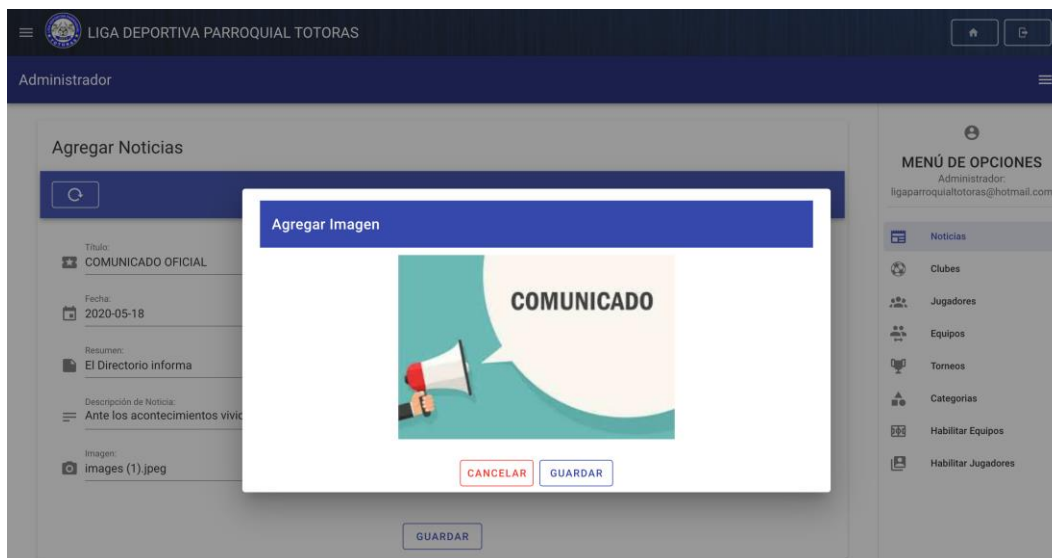
El administrador puede agregar, editar, eliminar y visualizar noticias en tiempo real, por tanto, el usuario que visite la página visualizará la información establecida desde la sección Noticias

- **Agregar Noticia**

Seleccionar el ítem Noticias y llenar todos los campos del formulario Agregar Noticias. La fecha actual se establece por defecto, con opción a cambiarla.



Para agregar una imagen, seleccionar el campo de imagen en el formulario de Ingreso y escoger la imagen requerida desde el dispositivo que esté en uso la aplicación.



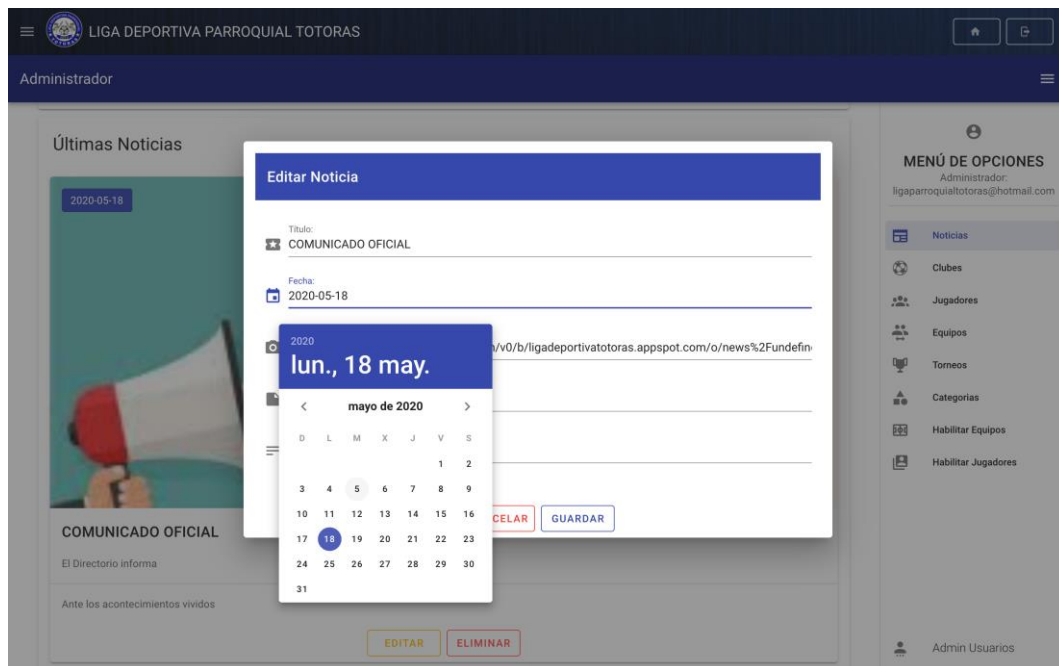
- **Visualizar Noticias**

En la sección Últimas Noticias, ubicada debajo del formulario de ingreso, se encuentran ordenadas las noticias desde la más reciente con los botones para editar o eliminar.



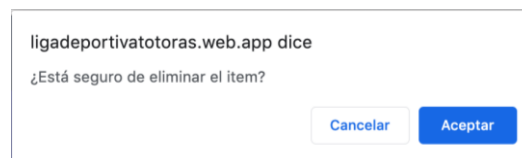
- **Editar Noticia**

Ubicarse en la noticia respectiva y seleccionar el botón EDITAR, seguidamente se abre un cuadro de diálogo con el formulario y los campos precargados para su edición. La información se actualizará automáticamente en la sección Últimas Noticias.



- **Eliminar Noticia**

Ubicarse en la noticia respectiva y seleccionar el botón ELIMINAR, seguidamente se muestra un cuadro de diálogo con un mensaje de confirmación. La información se actualizará automáticamente en la sección Últimas Noticias.



## 5. Administrar Clubes

El administrador puede agregar, editar, eliminar, visualizar y realizar búsquedas de los datos requeridos de los clubes. Seleccionar en el ítem Clubes y se mostrará la tabla respectiva con datos ordenados por nombre del club.

LIGA DEPORTIVA PARROQUIAL TOTORAS

Administrador

### Clubes

Buscar

| LOGO | NOMBRE ↑                | FUNDACIÓN  | PRESIDENTE    | TELÉFONO   |
|------|-------------------------|------------|---------------|------------|
|      | CLUB ATLÉTICO MARAÑÓN   | 1957-08-30 | CARLOS ALDAZ  | 0993872537 |
|      | CLUB DEPORTIVO MIRADOR  | 1983-10-23 | DIEGO HIDALGO | 0984086794 |
|      | CLUB DEPORTIVO NACIONAL | 1960-03-19 | GUIDO SACA    | 0998486379 |


Rows per page: 10 1-3 of 3

MENÚ DE OPCIONES  
 Administrador:  
 ligaparroquialtotoras@hotmail.com

- Noticias
- Clubes
- Jugadores
- Equipos
- Torneos
- Categorías
- Habilitar Equipos
- Habilitar Jugadores

Admin Usuarios

- **Agregar Club**

Seleccionar el botón , se abre un cuadro de diálogo con el formulario. Se deberán llenar los campos obligatorios, caso contrario no se permite guardar los datos de un nuevo club.

**Nuevo Club**

Club o Institución Deportiva:

Acuerdo Ministerial:

Presidente:

Fecha:

Vicepresidente:

Teléfono:

Secretario(a):

Correo:


Tesorero(a):

Delegado 1:


Logo:

Delegado 2:

- **Editar Club**

Al seleccionar el ícono Editar , se abre un cuadro de diálogo con el formulario y los campos precargados, donde se procede a editar los datos pertinentes.

- **Eliminar Club**

Seleccionar el botón , se abre una alerta del navegador para confirmar la acción.

- **Visualizar Clubes**

Seleccionar el botón , para visualizar la información de un respectivo club.

| Clubes  |   |            |               |            |  |
|---|---|------------|---------------|------------|--|
| + Buscar <input type="text" value="Buscar"/>  |   |            |               |            |  |
| LOGO  | NOMBRE ↑  | FUNDACIÓN  | PRESIDENTE    | TELÉFONO   |  |
|    |  CLUB ATLÉTICO MARAÑÓN   | 1957-08-30 | CARLOS ALDAZ  | 0993872537 |  |
|    |  CLUB DEPORTIVO MIRADOR  | 1983-10-23 | DIEGO HIDALGO | 0984086794 |  |
|    |  CLUB DEPORTIVO NACIONAL | 1960-03-19 | GUIDO SACA    | 0998486379 |  |

Rows per page: 10 1-3 of 3 < >

- **Buscar Clubes**

Para realizar una búsqueda inteligente, el administrador debe digitar lo requerido en el campo Buscar, ubicado en el encabezado de la tabla.

Clubes

Buscar MIRA

| LOGO | NOMBRE ↑               | FUNDACIÓN  | PRESIDENTE    | TELÉFONO   |
|------|------------------------|------------|---------------|------------|
|      | CLUB DEPORTIVO MIRADOR | 1983-10-23 | DIEGO HIDALGO | 0984086794 |

Rows per page: 10 1-1 of 1

## 6. Administrar Jugadores

El administrador puede agregar, editar, eliminar, visualizar y realizar búsquedas de los datos requeridos de los jugadores. Seleccionar en el ítem Jugadores y se mostrará la tabla respectiva con datos ordenados por los apellidos del jugador.

LIGA DEPORTIVA PARROQUIAL TOTORAS

Administrador

Jugadores

Buscar

| CÉDULA     | CÓDIGO  | APELLIDOS ↑       | NOMBRES         | DIRECCIÓN      |
|------------|---------|-------------------|-----------------|----------------|
| 1802397293 | LT0398  | HIDALGO NAVARRETE | DIEGO FERNANDO  | BARRIO MIRADOR |
| 1801029001 | LT02938 | MARIN DUARTE      | LUISA FERNANDA  | CUMANDA        |
| 1804348009 | LT09293 | YAGUARGOS CASTRO  | SANTIAGO DAMIAN | BARRIO MIRADOR |


Rows per page: 10 1-3 of 3

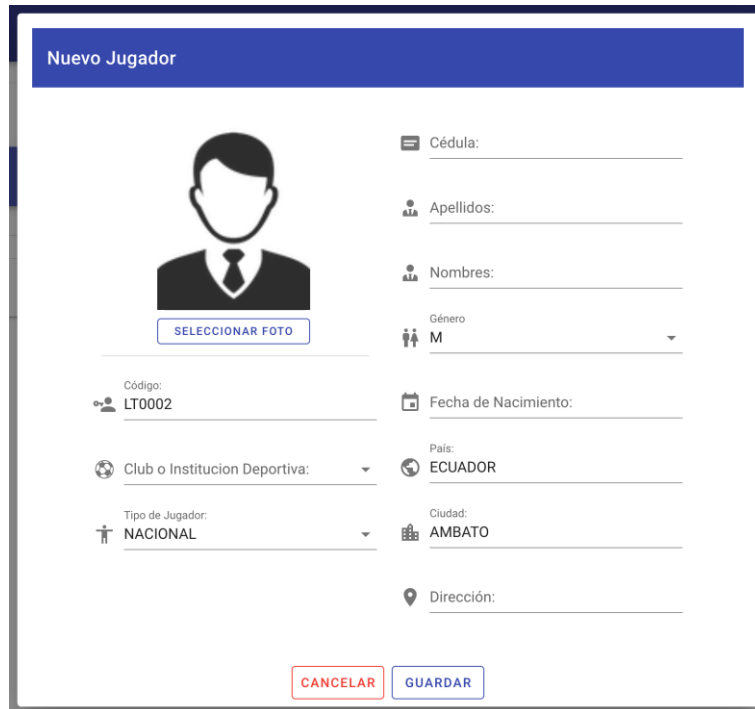
**MENÚ DE OPCIONES**  
 Administrador:  
 ligaparroquialtotoras@hotmail.com

- Noticias
- Clubes
- Jugadores**
- Equipos
- Torneos
- Categorías
- Habilitar Equipos
- Habilitar Jugadores

Admin Usuarios

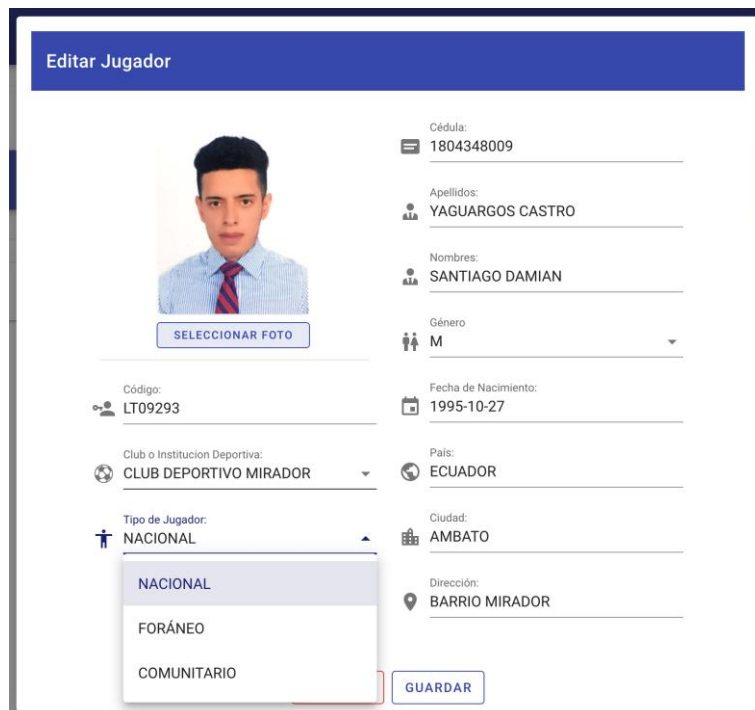
- **Agregar Jugador**

Seleccionar el botón , se abre un cuadro de diálogo con el formulario. Se deberán llenar los campos obligatorios, caso contrario no se permite guardar los datos de un nuevo jugador. El código del jugador se genera automáticamente.




- **Editar Jugador**

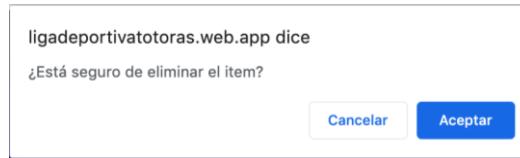
Al seleccionar el ícono Editar ✎, se abre un cuadro de diálogo con el formulario y los campos precargados, donde se procede a editar los datos pertinentes.






- **Eliminar Jugador**

Seleccionar el botón , se abre una alerta del navegador para confirmar la acción.



- **Visualizar Jugadores**


Seleccionar el botón , para visualizar la información de un respectivo jugador.

| Jugadores   |            |         |                   |                 |                |
|---|------------|---------|-------------------|-----------------|----------------|
| + Buscar  |            |         |                   |                 |                |
|   | CÉDULA     | CÓDIGO  | APELLIDOS ↑       | NOMBRES         | DIRECCIÓN      |
|       | 1802397293 | LT0398  | HIDALGO NAVARRETE | DIEGO FERNANDO  | BARRIO MIRADOR |
|       | 1801029001 | LT02938 | MARIN DUARTE      | LUISA FERNANDA  | CUMANDA        |
|    | 1804348009 | LT09293 | YAGUARGOS CASTRO  | SANTIAGO DAMIAN | BARRIO MIRADOR |

Rows per page: 10 1-3 of 3 < >

- **Buscar Jugadores**

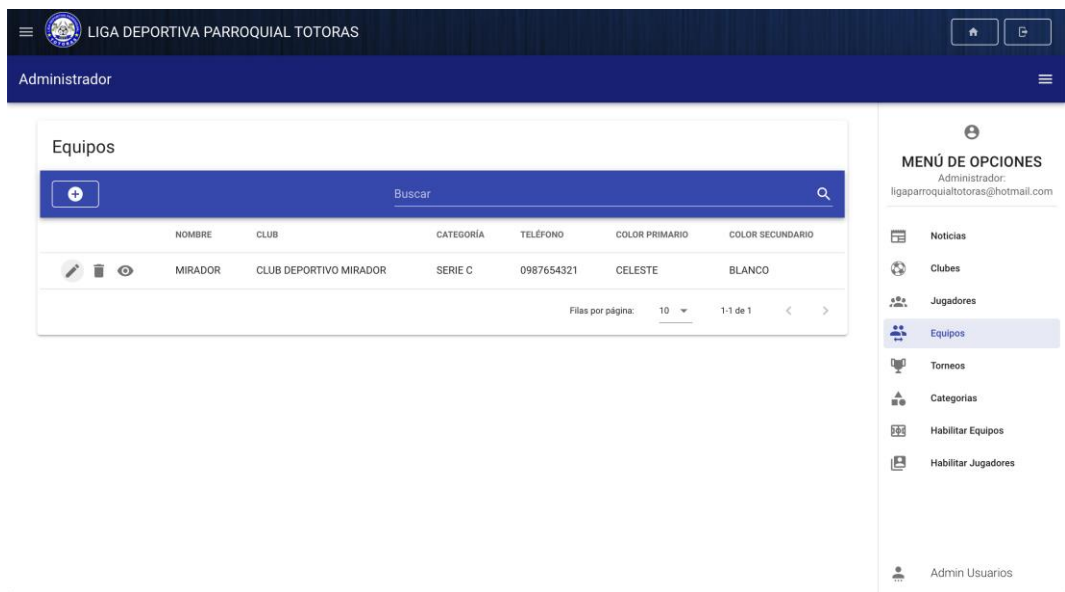
Para realizar una búsqueda inteligente, el administrador debe digitar lo requerido en el campo Buscar, ubicado en el encabezado de la tabla.

| Jugadores   |            |         |                  |                 |                |
|---|------------|---------|------------------|-----------------|----------------|
| + Buscar YAGU   |            |         |                  |                 |                |
|   | CÉDULA     | CÓDIGO  | APELLIDOS ↑      | NOMBRES         | DIRECCIÓN      |
|    | 1804348009 | LT09293 | YAGUARGOS CASTRO | SANTIAGO DAMIAN | BARRIO MIRADOR |


Rows per page: 10 1-1 of 1 < >

## 7. Administrar Equipos


El administrador puede agregar, editar, eliminar, visualizar y realizar búsquedas de los datos requeridos de los equipos. Seleccionar en el ítem Equipos y se mostrará la tabla respectiva con datos ordenados por los nombres del equipo.



- **Agregar Jugador**

Seleccionar el botón , se abre un cuadro de diálogo con el formulario. Se deberán llenar los campos obligatorios, caso contrario no se permite guardar los datos de un nuevo equipo. El equipo pertenece a un club y una categoría.

- **Editar Equipo**

Al seleccionar el ícono Editar , se abre un cuadro de diálogo con el formulario y los campos precargados, donde se procede a editar los datos pertinentes.

**Editar Equipo**

Club o Institución Deportiva:  
 CLUB DEPORTIVO MIRADOR

Categoría:  
 SERIE C

Nombre:  
 MIRADOR

Teléfono:  
 0987654321

Color Principal:  
 CELESTE

Color Secundario:  
 AZUL

**CANCELAR** **GUARDAR**

- **Eliminar Equipo**

Seleccionar el botón , se abre una alerta del navegador para confirmar la acción.

ligadeportivatotoras.web.app dice  
 ¿Está seguro de eliminar el item?

**Cancelar** **Aceptar**

- **Visualizar Jugadores**

Seleccionar el botón , para visualizar la información de un respectivo jugador.

| Equipos   |         |                        |           |            |                |                  |
|---|---------|------------------------|-----------|------------|----------------|------------------|
| <div style="float: right; text-align: right;">           Buscar  </div> |         |                        |           |            |                |                  |
|   | NOMBRE  | CLUB                   | CATEGORÍA | TELÉFONO   | COLOR PRIMARIO | COLOR SECUNDARIO |
|   | MIRADOR | CLUB DEPORTIVO MIRADOR | SERIE C   | 0987654321 | CELESTE        | BLANCO           |

Filas por página: 10 1-1 de 1

- **Buscar Equipos**

Para realizar una búsqueda inteligente, el administrador debe digitar lo requerido en el campo Buscar, ubicado en el encabezado de la tabla.

| Equipos |        |        |             |                         |                |                  |      |      |
|---------|--------|--------|-------------|-------------------------|----------------|------------------|------|------|
|         |        | Buscar |             |                         |                |                  |      |      |
|         |        | NACIO  |             |                         |                |                  |      |      |
|         | NOMBRE | CLUB   | CATEGORÍA   | TELÉFONO                | COLOR PRIMARIO | COLOR SECUNDARIO |      |      |
|         |        |        | NACIONAL    | CLUB DEPORTIVO NACIONAL | SERIE A        | 0983243983       | ROJO | AZUL |
|         |        |        | NACIONAL JR | CLUB DEPORTIVO NACIONAL | SERIE C        | 0987634734       | ROJO | ROJO |

Filas por página: 10 1-2 de 2

## 8. Administrar Torneos

El administrador puede agregar, editar, eliminar, visualizar y realizar búsquedas de los datos requeridos de los torneos. Seleccionar en el ítem Torneos y se mostrará la tabla respectiva con datos ordenados por los nombres del torneo.

The screenshot shows the administrator interface for 'LIGA DEPORTIVA PARROQUIAL TOTORAS'. The main content area displays a table of tournaments:

| Torneos |        |        |                              |      |                    |
|---------|--------|--------|------------------------------|------|--------------------|
|         |        | Buscar |                              |      |                    |
|         |        |        |                              |      |                    |
|         | NOMBRE | AÑO    | DESCRIPCIÓN                  |      |                    |
|         |        |        | CAMPEONATO INTERNO DE FUTBOL | 2020 | Campeonato Amateur |

Filas por página: 10 1-1 de 1

The sidebar menu on the right includes the following options:


- Noticias
- Clubes
- Jugadores
- Equipos
- Torneos** (selected)
- Categorías
- Habilitar Equipos
- Habilitar Jugadores
- Admin Usuarios

- **Agregar Torneo**

Seleccionar el botón , se abre un cuadro de diálogo con el formulario. Se deberán llenar los campos obligatorios, caso contrario no se permite guardar los datos de un nuevo torneo. El año actual se carga por defecto.




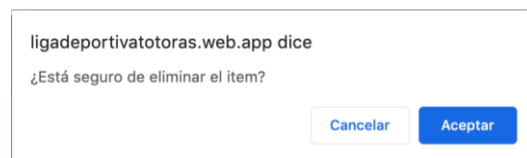
- **Editar Torneo**

Al seleccionar el ícono Editar , se abre un cuadro de diálogo con el formulario y los campos precargados, donde se procede a editar los datos pertinentes.



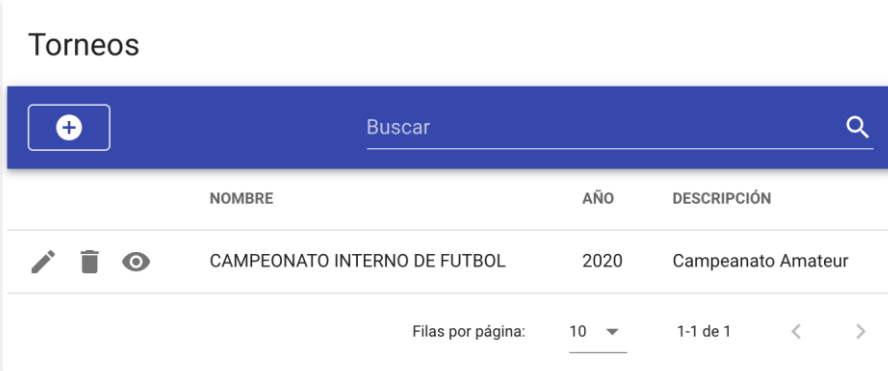
- **Eliminar Torneo**




Seleccionar el botón , se abre una alerta del navegador para confirmar la acción.



- **Visualizar Torneos**

Seleccionar el botón , para visualizar la información de un respectivo torneo.



| Torneos   |                              |      |                    |
|---|------------------------------|------|--------------------|
| Buscar  |                              |      |                    |
|   | NOMBRE                       | AÑO  | DESCRIPCIÓN        |
|    | CAMPEONATO INTERNO DE FUTBOL | 2020 | Campeonato Amateur |

Filas por página: 10 1-1 de 1

- **Buscar Torneo**

Para realizar una búsqueda inteligente, el administrador debe digitar lo requerido en el campo Buscar, ubicado en el encabezado de la tabla.

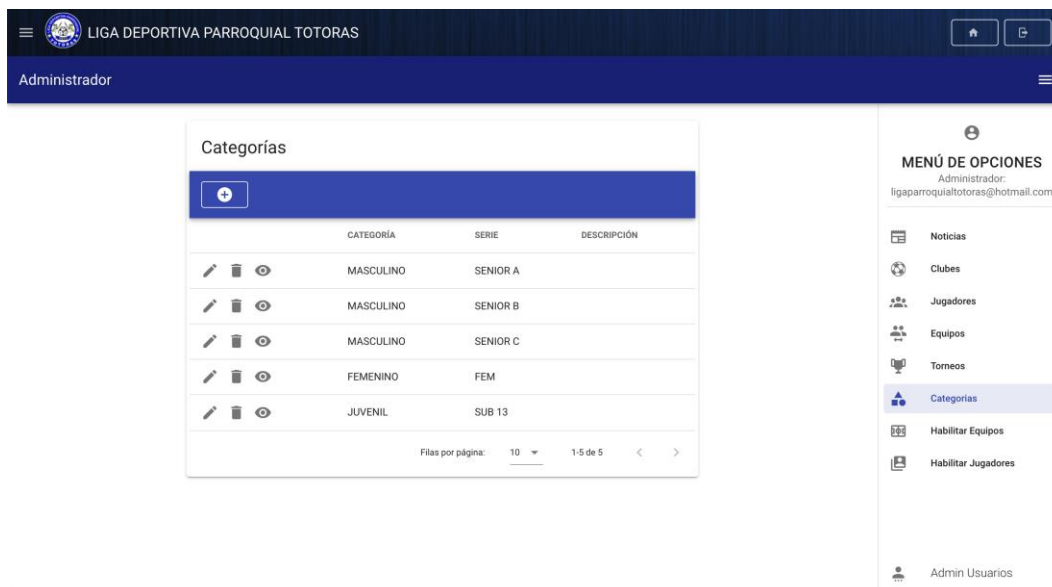


| Torneos   |                              |      |                    |
|---|------------------------------|------|--------------------|
| Buscar<br>CAMPEONA  |                              |      |                    |
|   | NOMBRE                       | AÑO  | DESCRIPCIÓN        |
|    | CAMPEONATO INTERNO DE FUTBOL | 2020 | Campeonato Amateur |


Filas por página: 10 1-1 de 1

## 9. Administrar Categorías


El administrador puede agregar, editar, eliminar y visualizar los datos requeridos de las categorías. Seleccionar en el ítem Categorías y se mostrará la tabla respectiva.




- **Agregar Categoría**

Seleccionar el botón , se abre un cuadro de diálogo con el formulario. Se deberán llenar los campos obligatorios, caso contrario no se permite guardar los datos de una nueva categoría.

- **Editar Categoría**

Al seleccionar el ícono Editar , se abre un cuadro de diálogo con el formulario y los campos precargados, donde se procede a editar los datos pertinentes.

- **Eliminar Categoría**

Seleccionar el botón , se abre una alerta del navegador para confirmar la acción.

- **Visualizar Categorías**

Seleccionar el botón , para visualizar la información de un respectivo torneo.

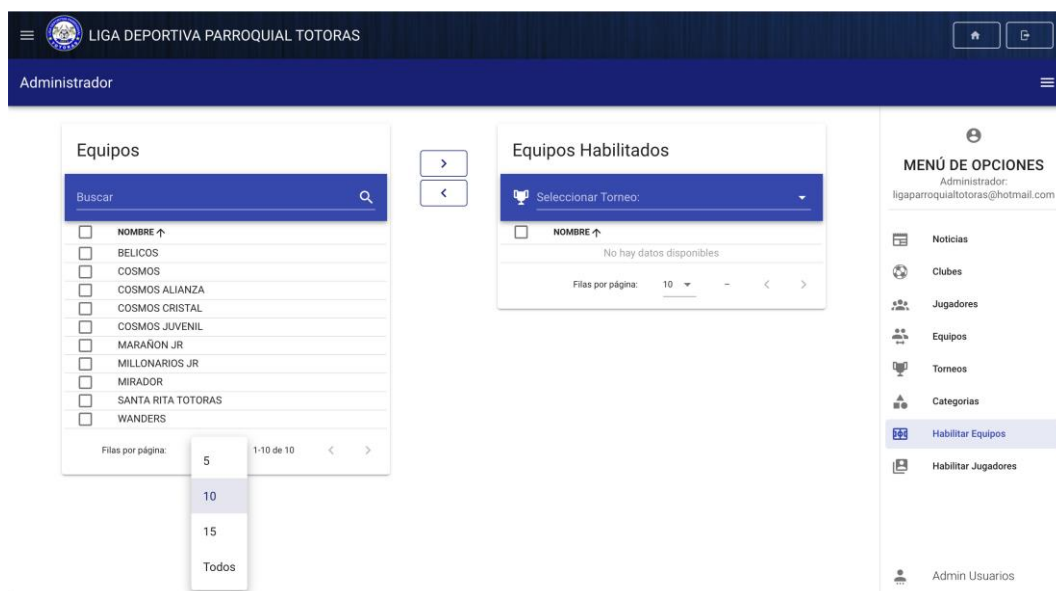
| Categorías  |           |          |             |
|---|-----------|----------|-------------|
| +   |           |          |             |
|   | CATEGORÍA | SERIE    | DESCRIPCIÓN |
|    | MASCULINO | SENIOR A |             |
|    | MASCULINO | SENIOR B |             |
|    | MASCULINO | SENIOR C |             |
|    | FEMENINO  | FEM      |             |
|    | JUVENIL   | SUB 13   |             |



Filas por página: 10 1-5 de 5 < >



## 10. Habilitar e inhabilitar Equipos

El administrador puede habilitar e inhabilitar equipos en o de un determinado torneo respectivamente. Seleccionar en el ítem Habilitar Equipos y se mostrará la vista correspondiente.



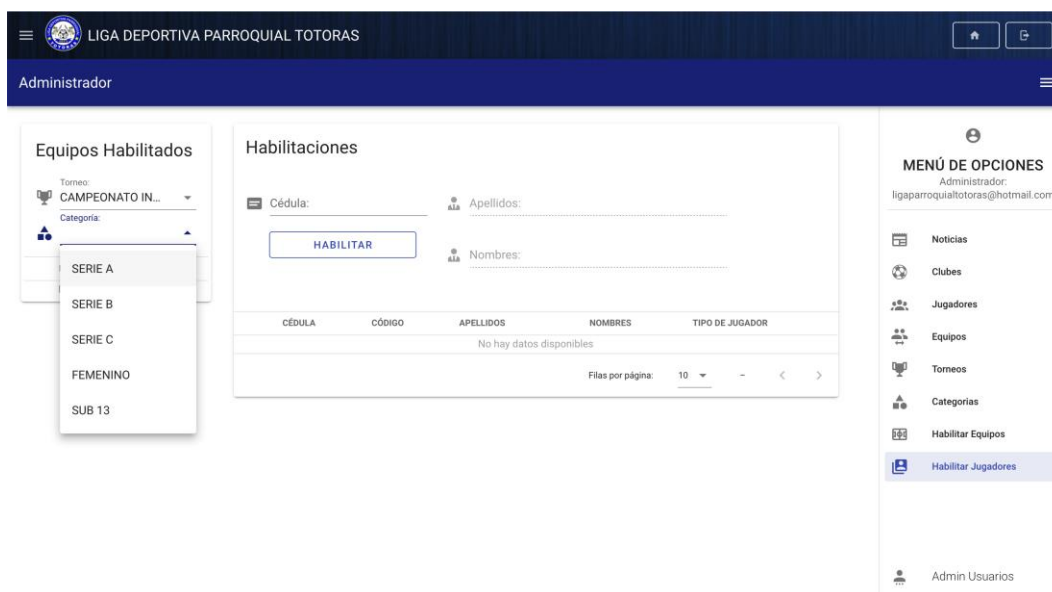
Seleccionar un torneo en la tabla de Equipos Habilitados (por defecto el último torneo), luego seleccionar de la tabla Equipos, el o los equipos para su habilitación. Finalmente, al seleccionar el botón . Para inhabilitar equipos, se realiza el proceso de manera viceversa desde la tabla Equipos Habilitados y seleccionado el botón .



## 11. Habilitar e inhabilitar Jugadores

El administrador puede habilitar e inhabilitar jugadores en o de un determinado equipo habilitado respectivamente. Seleccionar en el ítem Habilitar Jugadores y se mostrará la vista correspondiente.

Seleccionar un torneo en la tabla de Equipos Habilitados (por defecto el último torneo), luego seleccionar la categoría correspondiente donde se filtran los equipos habilitados.




Una vez seleccionado el torneo y la categoría, se cargan los equipos habilitados ordenados por el nombre del equipo. Luego seleccionar el equipo, para habilitar o deshabilitar jugadores.



- **Habilitar Jugadores**

Ingresar el número de cédula del jugador, se cargará automáticamente apellidos, nombres y la respectiva foto. Seleccionar el botón HABILITAR para agregarlo en un equipo. El jugador no puede ser habilitado si no existe equipo seleccionado.

- **Habilitar Jugadores**

En la tabla de jugadores habilitados por cada equipo, seleccionar el ícono , para la respectiva inhabilitación.