



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS

Tema:

SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS
INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN
SISTEMAS, ELECTRÓNICA E INDUSTRIAL.

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: Omar Patricio Fonseca Acosta

TUTOR: Ing. Mg. Carlos Israel Nuñez Miranda

Ambato - Ecuador

marzo - 2023

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL, desarrollado bajo la modalidad Proyecto de Investigación por el señor Omar Patricio Fonseca Acosta, 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, marzo 2023

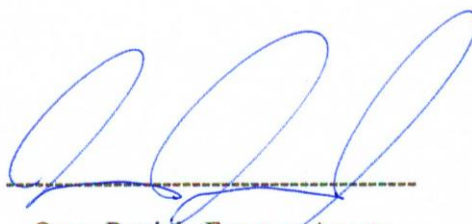
Ing. Mg. Carlos Israel Núñez Miranda

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL, 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, marzo 2023.



Omar Patricio Fonseca Acosta

C.C. 1805028915

AUTOR

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, marzo 2023.



Omar Patricio Fonseca Acosta

C.C. 1805028915

AUTOR

APROBACION DEL TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Omar Patricio Fonseca Acosta, 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 SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL 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 Presidente del Tribunal.

Ambato, marzo 2023.

Ing. Mg. Pilar Urrutia
PRESIDENTE DEL TRIBUNAL

Ing. Mg. Rubén Eduardo
Nogales Portero
PROFESOR CALIFICADOR

Ing. Mg. Hernán Fabricio
Naranjo Ávalos
PROFESOR CALIFICADOR

DEDICATORIA

Este trabajo de investigación está dedicado a mi madre Beatriz Acosta y a mi padre Luis Fonseca por su apoyo incondicional, por ser el motor en mi vida y las personas que nunca dejaron de creer en mí, sin ellos nada de esto sería posible.

AGADRECIMIENTO

A mi madre y mi padre, por ser las personas más importantes en mi vida, quienes nunca dejaron de apoyarme.

A mis amigos que me brindaron su apoyo y me motivaron a cumplir con mis objetivos.

A la UTA y FISEI, por la formación académica que recibí para que me convierta en un profesional.

A mi tutor el Ing. Carlos Nuñez que me guio durante el desarrollo del proyecto.

A cada maestro que hizo parte de este proceso integral de formación.

Omar Patricio Fonseca Acosta

ÍNDICE

APROBACIÓN DEL TUTOR.....	II
AUTORÍA.....	III
DERECHOS DE AUTOR.....	IV
APROBACION DEL TRIBUNAL DE GRADO.....	V
DEDICATORIA	II
AGADRECIMIENTO.....	III
RESUMEN EJECUTIVO	XIX
ABSTRACT.....	XX
INTRODUCCIÓN	XXI
1 MARCO TEORICO.....	1
1.1 Tema de Investigación	1
1.2. Antecedentes investigativos	1
1.2.1 Contextualización del problema	3
1.2.2 Justificación.....	4
1.2.3 Delimitación	5
1.2.3.1 De contenidos	5
1.2.3.2 Espacial.....	6
1.2.3.3 Temporal.....	6
1.2.4 Fundamentación teórica.....	6
1.3. Objetivos	30
1.3.1 Objetivo general	30
1.3.2 Objetivos específicos.....	30
2 METODOLOGIA	32
2.1. Materiales.....	32
2.1.1 ENTREVISTA	32
2.1.2 ENCUESTA	33

2.2. Métodos.....	34
2.2.1 Modalidad de Investigación.....	34
2.2.2 Población y Muestra.....	34
2.2.3 Recolección de información.....	35
2.2.4 Procesamiento y análisis de datos.....	43
3 RESULTADOS Y DISCUSIÓN.....	44
3.1. Análisis y discusión de resultados.....	44
3.1.1 Comparativa de metodologías ágiles.....	44
3.1.2 Metodología seleccionada.....	45
3.1.3 Comparativa de frameworks de desarrollo PWA.....	45
3.1.4 Framework de desarrollo seleccionado.....	47
3.1.5 Determinación de la tecnología Back-End.....	48
3.1.6 Arquitectura de la aplicación.....	48
3.2. Desarrollo de la propuesta.....	48
3.2.1 Fase I: Planificación.....	48
3.2.2 Fase II: Diseño del proyecto.....	79
3.2.2.1 Metáfora de la aplicación.....	79
3.2.2.2 Iteración I.....	80
3.2.2.3 Iteración II.....	82
3.2.2.4 Iteración III.....	86
3.2.2.5 Iteración IV.....	88
3.2.2.6 Iteración V.....	95
3.2.2.7 Iteración VI y VII.....	99
3.2.3 Fase III: Codificación.....	117
3.2.3.1 Métodos del backend Node.js.....	117
3.2.3.2 Métodos del Frontend Angular.....	166
3.2.4 Fase IV: Pruebas.....	214
4 CONCLUSIONES Y RECOMENDACIONES.....	222
4.1. Conclusiones.....	222
4.2 Recomendaciones.....	222
BIBLIOGRAFÍA.....	223

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1: PWA	11
ILUSTRACIÓN 2: ARQUITECTURA DE UNA PWA	13
ILUSTRACIÓN 3: FUNCIONAMIENTO DE UNA PWA	14
ILUSTRACIÓN 4: ARQUITECTURA DE UN WEB FRAMEWORK.....	16
ILUSTRACIÓN 5: ARQUITECTURA DE ANGULAR	22
ILUSTRACIÓN 6: RESULTADO DE ENCUESTAS - PREGUNTA 1.....	40
ILUSTRACIÓN 7: RESULTADO DE ENCUESTAS - PREGUNTA 2.....	40
ILUSTRACIÓN 8: RESULTADO DE ENCUESTAS - PREGUNTA 3.....	41
ILUSTRACIÓN 9: RESULTADO DE ENCUESTAS - PREGUNTA 4.....	41
ILUSTRACIÓN 10: RESULTADO DE ENCUESTAS - PREGUNTA 5.....	42
ILUSTRACIÓN 11: RESULTADO DE ENCUESTAS - PREGUNTA 6.....	42
ILUSTRACIÓN 12: ARQUITECTURA CLIENTE-SERVIDOR	48
ILUSTRACIÓN 13: MÓDULOS ESTABLECIDOS	49
ILUSTRACIÓN 14: INICIO CSEI	53
ILUSTRACIÓN 15: REGISTRO Y VALIDACIÓN	53
ILUSTRACIÓN 16: FACTURACIÓN	53
ILUSTRACIÓN 17: FIN CSEI	53
ILUSTRACIÓN 18. INICIALIZACIÓN DEL PROYECTO	82
ILUSTRACIÓN 19. ESTRUCTURA DEL PROYECTO	82
ILUSTRACIÓN 20. ESTRUCTURA DE LA BASE DE DATOS	81
ILUSTRACIÓN 21. DISEÑO DE INTERFAZ MÓDULO INFORMATIVO	82
ILUSTRACIÓN 22. DISEÑO DE INTERFAZ MÓDULO REGISTRO 1	83
ILUSTRACIÓN 23. DISEÑO DE INTERFAZ MÓDULO REGISTRO 2	84
ILUSTRACIÓN 24. DISEÑO DE INTERFAZ MÓDULO REGISTRO 3	84
ILUSTRACIÓN 25. DISEÑO DE INTERFAZ MÓDULO VALIDACIÓN	85
ILUSTRACIÓN 26. DISEÑO DE INTERFAZ MÓDULO FACTURAS	85
ILUSTRACIÓN 27. DISEÑO DE INTERFAZ MÓDULO CERTIFICADOS	86
ILUSTRACIÓN 28. DISEÑO DE INTERFAZ MÓDULO CONFIGURACIÓN	87
ILUSTRACIÓN 29. INICIO DE SESIÓN.....	88
ILUSTRACIÓN 30. PÁGINA DE INICIO.....	89
ILUSTRACIÓN 31. PÁGINA DE SOBRE EL EVENTO	89
ILUSTRACIÓN 32. PÁGINA DE CONFERENCISTAS	90
ILUSTRACIÓN 33. PÁGINA DE COMITÉ.....	90
ILUSTRACIÓN 34. PÁGINA DE PROGRAMA	91
ILUSTRACIÓN 35. PÁGINA DE ENVIÓ DE TRABAJOS	91
ILUSTRACIÓN 36. PÁGINA DE CONTACTO	92
ILUSTRACIÓN 37. PÁGINA DE EDICIONES ANTERIORES	92

ILUSTRACIÓN 38. INGRESO AL MÓDULO DE REGISTRO.....	93
ILUSTRACIÓN 39. FORMULARIO DE REGISTRO	94
ILUSTRACIÓN 40. SELECCIÓN DE MÉTODO DE PAGO	94
ILUSTRACIÓN 41. FINALIZAR LA INSCRIPCIÓN	95
ILUSTRACIÓN 42. INGRESO MÓDULO DE VALIDACIÓN	96
ILUSTRACIÓN 43. FORMULARIO DE VALIDACIÓN.....	97
ILUSTRACIÓN 44. INGRESO MÓDULO DE FACTURAS.....	97
ILUSTRACIÓN 45. MÓDULO DE FACTURAS.....	98
ILUSTRACIÓN 46. INGRESO MÓDULO DE CERTIFICADOS.....	98
ILUSTRACIÓN 47. MÓDULO DE CERTIFICADOS	99
ILUSTRACIÓN 48. SUBMÓDULO CONTROL DE INSCRIPCIONES	100
ILUSTRACIÓN 49. SUBMÓDULO DE REPORTES	100
ILUSTRACIÓN 50. REPORTE GENERAL.....	101
ILUSTRACIÓN 51. REPORTES FILTRADOS	101
ILUSTRACIÓN 52. REPORTES PAPER	101
ILUSTRACIÓN 53. SUBMÓDULO DE CERTIFICADOS ASISTENCIA Y PARTICIPACIÓN ..	102
ILUSTRACIÓN 54. CERTIFICADO DE ASISTENCIA	102
ILUSTRACIÓN 55. CERTIFICADO DE PARTICIPACIÓN	103
ILUSTRACIÓN 56. SUBMÓDULO DE CERTIFICADOS DE AUTORES.....	103
ILUSTRACIÓN 57. CERTIFICADO AUTORES.....	103
ILUSTRACIÓN 58. SUBMÓDULO DE CERTIFICADOS DE EXPOSITORES.....	104
ILUSTRACIÓN 59. CERTIFICADOS EXPOSITORES	104
ILUSTRACIÓN 60.SUBMÓDULO ADMINISTRACIÓN DEL COMITÉ	105
ILUSTRACIÓN 61. SUBMÓDULO GESTIÓN DE USUARIOS.....	105
ILUSTRACIÓN 62. SUBMÓDULO DE CONFIGURACIÓN – OPCIONES DE INSCRIPCIÓN .	105
ILUSTRACIÓN 63. SUBMÓDULO DE CONFIGURACIÓN – PATROCINADORES	106
ILUSTRACIÓN 64. SUBMÓDULO DE CONFIGURACIÓN – INFORMACIÓN DEL REGISTRO	106
ILUSTRACIÓN 65. SUBMÓDULO DE CONFIGURACIÓN – EDICIONES.....	107
ILUSTRACIÓN 66. AGREGAR NUEVA EDICIÓN.....	107
ILUSTRACIÓN 67. SUBMÓDULO DE CONFIGURACIÓN – REVISTAS.....	108
ILUSTRACIÓN 68. AGREGAR REVISTA	108
ILUSTRACIÓN 69. SUBMÓDULO DE CONFIGURACIÓN – CARRERAS.....	108
ILUSTRACIÓN 70. SUBMÓDULO DE CONFIGURACIÓN – SEMESTRES	109
ILUSTRACIÓN 71. SUBMÓDULO DE CONFIGURACIÓN – TIPO DE INSCRIPCIÓN	109
ILUSTRACIÓN 72. SUBMÓDULO DE CONFIGURACIÓN – IDIOMAS.....	110
ILUSTRACIÓN 73. SUBMÓDULO DE CONFIGURACIÓN – CONFERENCISTAS	110
ILUSTRACIÓN 74. SUBMÓDULO DE CONFIGURACIÓN – INFORMACIÓN DEL CONGRESO	111
ILUSTRACIÓN 75. SUBMÓDULO DE CONFIGURACIÓN – IMÁGENES DE PORTADA	111

ILUSTRACIÓN 76. SUBMÓDULO DE CONFIGURACIÓN – GALERÍA DEL LUGAR	112
ILUSTRACIÓN 77. SUBMÓDULO DE CONFIGURACIÓN – GALERÍA INFORMATIVA.....	112
ILUSTRACIÓN 78. SUBMÓDULO DE CONFIGURACIÓN – PRESENTACIÓN	113
ILUSTRACIÓN 79. SUBMÓDULO DE CONFIGURACIÓN – LUGAR DEL EVENTO.....	113
ILUSTRACIÓN 80. SUBMÓDULO DE CONFIGURACIÓN – INFORMACIÓN TURÍSTICA ...	114
ILUSTRACIÓN 81. SUBMÓDULO DE CONFIGURACIÓN – PROGRAMA	114
ILUSTRACIÓN 82. SUBMÓDULO DE CONFIGURACIÓN – ENVIÓ DE TRABAJOS	114
ILUSTRACIÓN 83. SUBMÓDULO DE CONFIGURACIÓN – TEMARIO.....	115
ILUSTRACIÓN 84. SUBMÓDULO DE CONFIGURACIÓN – INICIO.....	115
ILUSTRACIÓN 85. SUBMÓDULO DE CONFIGURACIÓN – GALERÍA DEL INICIO	116
ILUSTRACIÓN 86. SUBMÓDULO DE CONFIGURACIÓN – CRONOGRAMA	116
ILUSTRACIÓN 87. BACKEND - LOGUIN	117
ILUSTRACIÓN 88. BACKEND – CHECK DE TOKEN.....	117
ILUSTRACIÓN 89. BACKEND – GUARDAR ARCHIVOS.....	118
ILUSTRACIÓN 90. BACKEND – CONSULTA LISTADO DE INSCRIPCIONES	118
ILUSTRACIÓN 91. BACKEND – CONSULTA INSCRIPCIÓN SEGÚN CEDULA.....	118
ILUSTRACIÓN 92. BACKEND – ACTUALIZAR INSCRIPCIÓN.....	119
ILUSTRACIÓN 93. BACKEND – LISTAR LAS CARRERAS.....	119
ILUSTRACIÓN 94. BACKEND – OBTENER CARRERA SEGÚN ID.....	119
ILUSTRACIÓN 95. BACKEND – ACTUALIZAR CARRERA	120
ILUSTRACIÓN 96. BACKEND – ACTUALIZAR CARRERA.....	120
ILUSTRACIÓN 97. BACKEND – ELIMINAR CARRERA.....	120
ILUSTRACIÓN 98. BACKEND– LISTAR COMITÉ.....	120
ILUSTRACIÓN 99. BACKEND – OBTENER COMITÉ SEGÚN ID	121
ILUSTRACIÓN 100. BACKEND – AGREGAR COMITÉ.....	121
ILUSTRACIÓN 101. BACKEND – ACTUALIZAR COMITÉ	121
ILUSTRACIÓN 102. BACKEND – ELIMINAR COMITÉ	121
ILUSTRACIÓN 103. BACKEND – LISTAR EDICIONES	122
ILUSTRACIÓN 104. BACKEND – CREAR COMITÉ.....	122
ILUSTRACIÓN 105. BACKEND – ACTUALIZAR COMITÉ	122
ILUSTRACIÓN 106. BACKEND – LISTAR ENVÍOS DE TRABAJOS	123
ILUSTRACIÓN 107. BACKEND – OBTENER ENVIÓ DE TRABAJO SEGÚN ID.....	123
ILUSTRACIÓN 108. BACKEND – CREAR ENVÍO DE TRABAJO	123
ILUSTRACIÓN 109. BACKEND – ACTUALIZAR ENVÍO DE TRABAJOS	124
ILUSTRACIÓN 110. BACKEND – ELIMINAR ENVÍO DE TRABAJOS	124
ILUSTRACIÓN 111. BACKEND – LISTAR FECHAS IMPORTANTES	124
ILUSTRACIÓN 112. BACKEND – OBTENER FECHA IMPORTANTE SEGÚN ID.....	125
ILUSTRACIÓN 113. BACKEND – CREAR FECHA IMPORTANTE	125
ILUSTRACIÓN 114. BACKEND – ACTUALIZAR FECHAS IMPORTANTES.....	125
ILUSTRACIÓN 115. BACKEND – ELIMINAR FECHA IMPORTANTE.....	126

ILUSTRACIÓN 116. BACKEND – LISTAR FORMATOS ENVÍO DE TRABAJOS	126
ILUSTRACIÓN 117. BACKEND – OBTENER FORMATO ENVÍO DE TRABAJO SEGÚN ID	126
ILUSTRACIÓN 118. BACKEND – CREAR FORMATO ENVÍO DE TRABAJO	127
ILUSTRACIÓN 119. BACKEND – ACTUALIZAR FORMATOS ENVÍO DE TRABAJOS	127
ILUSTRACIÓN 120. BACKEND – ELIMINAR FORMATOS ENVÍO DE TRABAJOS	127
ILUSTRACIÓN 121. BACKEND – LISTAR INSCRIPCIONES.....	128
ILUSTRACIÓN 122. BACKEND – OBTENER EXPOSITOR PAPER SEGÚN ID	128
ILUSTRACIÓN 123. BACKEND – ACTUALIZAR INSCRIPCIÓN.....	128
ILUSTRACIÓN 124. BACKEND – LISTAR INFORMACIÓN TURÍSTICA	128
ILUSTRACIÓN 125. BACKEND – OBTENER INFORMACIÓN TURÍSTICA SEGÚN ID.....	129
ILUSTRACIÓN 126. BACKEND – CREAR GALERÍA INFORMACIÓN TURÍSTICA.....	129
ILUSTRACIÓN 127. BACKEND – ACTUALIZAR GALERÍA INFORMACIÓN TURÍSTICA ..	129
ILUSTRACIÓN 128. BACKEND – ELIMINAR GALERÍA INFORMACIÓN TURÍSTICA	130
ILUSTRACIÓN 129. BACKEND – LISTAR GALERÍA LUGAR DEL EVENTO	130
ILUSTRACIÓN 130. BACKEND – CREAR GALERÍA LUGAR DEL EVENTO	130
ILUSTRACIÓN 131. BACKEND – ACTUALIZAR GALERÍA LUGAR DEL EVENTO	131
ILUSTRACIÓN 132. BACKEND – ELIMINAR GALERÍA LUGAR DEL EVENTO	131
ILUSTRACIÓN 133. BACKEND – LISTAR TIPO DE DOCUMENTO DE IDENTIFICACIÓN .	131
ILUSTRACIÓN 134. BACKEND – OBTENER TIPO DE DOCUMENTO DE IDENTIFICACIÓN SEGÚN ID.....	132
ILUSTRACIÓN 135. BACKEND – CREAR TIPO DE DOCUMENTO DE IDENTIFICACIÓN ..	132
ILUSTRACIÓN 136. BACKEND – ACTUALIZAR TIPO DE DOCUMENTO DE IDENTIFICACIÓN	132
ILUSTRACIÓN 137. BACKEND – ELIMINAR TIPO DE DOCUMENTO DE IDENTIFICACIÓN	133
ILUSTRACIÓN 138. BACKEND – LISTAR LOS IDIOMAS.....	133
ILUSTRACIÓN 139. BACKEND – OBTENER IDIOMA SEGÚN ID	133
ILUSTRACIÓN 140. BACKEND – CREAR IDIOMA	133
ILUSTRACIÓN 141. BACKEND – ACTUALIZAR IDIOMA.....	134
ILUSTRACIÓN 142. BACKEND – ELIMINAR IDIOMAS.....	134
ILUSTRACIÓN 143. BACKEND – LISTAR IMÁGENES DE PORTADA.....	134
ILUSTRACIÓN 144. BACKEND – OBTENER IMAGEN DE PORTADA SEGÚN ID	134
ILUSTRACIÓN 145. BACKEND – CREAR IMAGEN DE PORTADA	135
ILUSTRACIÓN 146. BACKEND – ACTUALIZAR IMAGEN DE PORTADA.....	135
ILUSTRACIÓN 147. BACKEND – ELIMINAR IMAGEN DE PORTADA.....	135
ILUSTRACIÓN 148. BACKEND – LISTAR INFORMACIÓN DEL CONGRESO	135
ILUSTRACIÓN 149. BACKEND – ACTUALIZAR INFORMACIÓN DEL CONGRESO.....	136
ILUSTRACIÓN 150. BACKEND – LISTAR INFORMACIÓN TURÍSTICA	136
ILUSTRACIÓN 151. BACKEND – OBTENER INFORMACIÓN TURÍSTICA SEGÚN ID.....	136
ILUSTRACIÓN 152. BACKEND – CREAR INFORMACIÓN TURÍSTICA.....	137

ILUSTRACIÓN 153. BACKEND – ACTUALIZAR INFORMACIÓN TURÍSTICA	137
ILUSTRACIÓN 154. BACKEND – ELIMINAR INFORMACIÓN TURÍSTICA	137
ILUSTRACIÓN 155. BACKEND – LISTAR INFORMACIÓN DEL INICIO	138
ILUSTRACIÓN 156. BACKEND – OBTENER INFORMACIÓN DEL INICIO SEGÚN ID	138
ILUSTRACIÓN 157. BACKEND – ACTUALIZAR INFORMACIÓN DEL INICIO.....	138
ILUSTRACIÓN 158. BACKEND – ELIMINAR INFORMACIÓN DE INICIO	138
ILUSTRACIÓN 159. BACKEND – LISTAR GALERÍA DE INICIO	139
ILUSTRACIÓN 160. BACKEND – OBTENER GALERÍA DE INICIO SEGÚN ID	139
ILUSTRACIÓN 161. BACKEND – CREAR GALERÍA DE INICIO.....	139
ILUSTRACIÓN 162. BACKEND – ACTUALIZAR GALERÍA DE INICIO.....	139
ILUSTRACIÓN 163. BACKEND – ELIMINAR GALERÍA DE INICIO.....	140
ILUSTRACIÓN 164. BACKEND – LISTAR INSCRIPCIONES.....	140
ILUSTRACIÓN 165. BACKEND – OBTENER INSCRIPCIÓN SEGÚN ID	140
ILUSTRACIÓN 166. BACKEND – CREAR INSCRIPCIÓN	140
ILUSTRACIÓN 167. BACKEND – ACTUALIZAR INSCRIPCIÓN.....	141
ILUSTRACIÓN 168. BACKEND – ELIMINAR INSCRIPCIÓN.....	141
ILUSTRACIÓN 169. BACKEND – LISTAR CONFERENCISTAS	141
ILUSTRACIÓN 170. BACKEND – CREAR CONFERENCISTA	141
ILUSTRACIÓN 171. BACKEND – ACTUALIZAR CONFERENCISTAS	142
ILUSTRACIÓN 172. BACKEND – ELIMINAR CONFERENCISTAS	142
ILUSTRACIÓN 173. BACKEND – LISTAR LUGARES DEL EVENTO	142
ILUSTRACIÓN 174. BACKEND – OBTENER LUGAR DEL EVENTO SEGÚN ID.....	143
ILUSTRACIÓN 175. BACKEND – CREAR LUGAR DEL EVENTO.....	143
ILUSTRACIÓN 176. BACKEND – ACTUALIZAR LUGAR DEL EVENTO	143
ILUSTRACIÓN 177. BACKEND – ELIMINAR LUGAR DEL EVENTO	143
ILUSTRACIÓN 178. BACKEND – ENVIAR MAIL A LOS INSCRITOS	144
ILUSTRACIÓN 179. BACKEND – ENVIAR MAIL CREACIÓN CERTIFICADOS	144
ILUSTRACIÓN 180. BACKEND – LISTAR TIPOS DE PAGO	144
ILUSTRACIÓN 181. BACKEND – OBTENER TIPO DE PAGO SEGÚN ID	145
ILUSTRACIÓN 182. BACKEND – CREAR TIPO DE PAGO.....	145
ILUSTRACIÓN 183. BACKEND – ACTUALIZAR TIPO DE PAGO.....	145
ILUSTRACIÓN 184. BACKEND – ELIMINAR TIPO DE PAGO.....	145
ILUSTRACIÓN 185. BACKEND – LISTAR PAPER.....	146
ILUSTRACIÓN 186. BACKEND – OBTENER PAPER SEGÚN ID	146
ILUSTRACIÓN 187. BACKEND – CREAR PAPER	146
ILUSTRACIÓN 188. BACKEND – ACTUALIZAR PAPER	146
ILUSTRACIÓN 189. BACKEND – ELIMINAR PAPER	147
ILUSTRACIÓN 190. BACKEND – LISTAR PARALELOS	147
ILUSTRACIÓN 191. BACKEND – OBTENER PARALELO SEGÚN ID.....	147
ILUSTRACIÓN 192. BACKEND – CREAR PARALELO	147

ILUSTRACIÓN 193. BACKEND – ACTUALIZAR PARALELO	148
ILUSTRACIÓN 194. BACKEND – ELIMINAR PARALELO	148
ILUSTRACIÓN 195. BACKEND – LISTAR PATROCINADORES	148
ILUSTRACIÓN 196. BACKEND – OBTENER PATROCINADOR SEGÚN ID	148
ILUSTRACIÓN 197. BACKEND – CREAR PARALELO	149
ILUSTRACIÓN 198. BACKEND – ACTUALIZAR PARALELO	149
ILUSTRACIÓN 199. BACKEND – ELIMINAR PARALELO	149
ILUSTRACIÓN 200. BACKEND – LISTAR PERSONAS	149
ILUSTRACIÓN 201. BACKEND – OBTENER PERSONA SEGÚN CI	150
ILUSTRACIÓN 202. BACKEND – CREAR PERSONA	150
ILUSTRACIÓN 203. BACKEND – ACTUALIZAR PERSONA	150
ILUSTRACIÓN 204. BACKEND – ELIMINAR PERSONA	150
ILUSTRACIÓN 205. BACKEND – LISTAR PRESENTACIÓN	151
ILUSTRACIÓN 206. BACKEND – OBTENER PRESENTACIÓN SEGÚN ID	151
ILUSTRACIÓN 207. BACKEND – CREAR PRESENTACIÓN	151
ILUSTRACIÓN 208. BACKEND – ACTUALIZAR PRESENTACIÓN	152
ILUSTRACIÓN 209. BACKEND – ACTUALIZAR PRESENTACIÓN	152
ILUSTRACIÓN 210. BACKEND – ELIMINAR PRESENTACIÓN	152
ILUSTRACIÓN 211. BACKEND – LISTAR PROGRAMA	152
ILUSTRACIÓN 212. BACKEND – CREAR PROGRAMA	153
ILUSTRACIÓN 213. BACKEND – ACTUALIZAR PROGRAMA	153
ILUSTRACIÓN 214. BACKEND – ELIMINAR PROGRAMA	153
ILUSTRACIÓN 215. BACKEND – LISTAR DETALLES DEL PROGRAMA	153
ILUSTRACIÓN 216. BACKEND – OBTENER DETALLE DEL PROGRAMA SEGÚN ID	154
ILUSTRACIÓN 217. BACKEND – CREAR DETALLE DEL PROGRAMA	154
ILUSTRACIÓN 218. BACKEND – ACTUALIZAR DETALLE DEL PROGRAMA	154
ILUSTRACIÓN 219. BACKEND – ELIMINAR DETALLE DEL PROGRAMA	154
ILUSTRACIÓN 220. BACKEND – LISTAR DÍAS DEL PROGRAMA	155
ILUSTRACIÓN 221. BACKEND – OBTENER DÍA DE PROGRAMA SEGÚN ID	155
ILUSTRACIÓN 222. BACKEND – CREAR DÍA DE PROGRAMA	155
ILUSTRACIÓN 223. BACKEND – ACTUALIZAR DÍA DEL PROGRAMA	155
ILUSTRACIÓN 224. BACKEND – ELIMINAR DÍA DEL PROGRAMA	156
ILUSTRACIÓN 225. BACKEND – LISTAR JORNADA DEL PROGRAMA	156
ILUSTRACIÓN 226. BACKEND – OBTENER JORNADA DEL PROGRAMA	156
ILUSTRACIÓN 227. BACKEND – CREAR JORNADA DEL PROGRAMA	156
ILUSTRACIÓN 228. BACKEND – ACTUALIZAR JORNADA DEL PROGRAMA	157
ILUSTRACIÓN 229. BACKEND – ELIMINAR JORNADA DEL PROGRAMA	157
ILUSTRACIÓN 230. BACKEND – LISTAR INFORMACIÓN DEL REGISTRO	157
ILUSTRACIÓN 231. BACKEND – OBTENER INFORMACIÓN DEL REGISTRO SEGÚN ID	157
ILUSTRACIÓN 232. BACKEND – CREAR INFORMACIÓN DEL REGISTRO	158

ILUSTRACIÓN 233. BACKEND – ACTUALIZAR INFORMACIÓN DEL CONGRESO.....	158
ILUSTRACIÓN 234. BACKEND – ELIMINAR INFORMACIÓN DEL CONGRESO	158
ILUSTRACIÓN 235. BACKEND – LISTAR SEMESTRES.....	158
ILUSTRACIÓN 236. BACKEND – OBTENER SEMESTRE SEGÚN ID	159
ILUSTRACIÓN 237. BACKEND – CREAR SEMESTRE	159
ILUSTRACIÓN 238. BACKEND – ACTUALIZAR SEMESTRE	159
ILUSTRACIÓN 239. BACKEND – ELIMINAR SEMESTRE	159
ILUSTRACIÓN 240. BACKEND – LISTAR TEMARIOS	160
ILUSTRACIÓN 241. BACKEND – OBTENER TEMARIO SEGÚN ID	160
ILUSTRACIÓN 242. BACKEND – CREAR TEMARIO.....	160
ILUSTRACIÓN 243. BACKEND – ACTUALIZAR TEMARIO.....	160
ILUSTRACIÓN 244. BACKEND – ELIMINAR TEMARIO.....	161
ILUSTRACIÓN 245. BACKEND – LISTAR TEMAS DEL TEMARIO	161
ILUSTRACIÓN 246. BACKEND – OBTENER TEMA DEL TEMARIO SEGÚN ID	161
ILUSTRACIÓN 247. BACKEND – CREAR TEMA DEL TEMARIO.....	161
ILUSTRACIÓN 248. BACKEND – ACTUALIZAR TEMA DEL TEMARIO.....	162
ILUSTRACIÓN 249. BACKEND – ELIMINAR TEMA DEL TEMARIO.....	162
ILUSTRACIÓN 250. BACKEND – LISTAR TIPO DE COMITÉ.....	162
ILUSTRACIÓN 251. BACKEND – OBTENER TIPO DE INSCRIPCIÓN.....	162
ILUSTRACIÓN 252. BACKEND – CREAR TIPO DE INSCRIPCIÓN.....	163
ILUSTRACIÓN 253. BACKEND – ACTUALIZAR TIPO DE INSCRIPCIÓN.....	163
ILUSTRACIÓN 254. BACKEND – ELIMINAR TIPO DE INSCRIPCIÓN.....	163
ILUSTRACIÓN 255. BACKEND – LISTAR TIPO DE PAPER.....	163
ILUSTRACIÓN 256. BACKEND – OBTENER TIPO DE PAPER SEGÚN ID.....	164
ILUSTRACIÓN 257. BACKEND – CREAR TIPO DE PAPER	164
ILUSTRACIÓN 258. BACKEND – ACTUALIZAR TIPO DE PAPER	164
ILUSTRACIÓN 259. BACKEND – LISTAR ELIMINAR TIPO DE PAPER	164
ILUSTRACIÓN 260. BACKEND – LISTAR TIPO DE USUARIO ADMINISTRADOR	165
ILUSTRACIÓN 261. BACKEND – OBTENER TIPO DE USUARIO ADMINISTRADOR SEGÚN ID.....	165
ILUSTRACIÓN 262. BACKEND – CREAR USUARIO	165
ILUSTRACIÓN 263. BACKEND – ACTUALIZAR USUARIO	166
ILUSTRACIÓN 264. BACKEND – ELIMINAR USUARIO	166
ILUSTRACIÓN 265. FRONTEND – OBTENER DETALLE DEL PROGRAMA.....	166
ILUSTRACIÓN 266. FRONTEND – OBTENER DÍAS DEL PROGRAMA	167
ILUSTRACIÓN 267. FRONTEND – OBTENER JORNADA DEL PROGRAMA	167
ILUSTRACIÓN 268. FRONTEND – OBTENER GALERÍA DE INICIO.....	167
ILUSTRACIÓN 269. FRONTEND – OBTENER INFORMACIÓN DEL INICIO.....	167
ILUSTRACIÓN 270. FRONTEND – OBTENER TEMARIOS	168
ILUSTRACIÓN 271. FRONTEND – OBTENER TEMA DE LOS TEMARIOS	168

ILUSTRACIÓN 272. FRONTEND – OBTENER INFORMACIÓN ENVIÓ DE TRABAJOS	168
ILUSTRACIÓN 273. FRONTEND – OBTENER INFORMACIÓN FECHAS ENVÍO DE TRABAJOS	169
ILUSTRACIÓN 274. FRONTEND – OBTENER INFORMACIÓN DEL PROGRAMA.....	169
ILUSTRACIÓN 275. FRONTEND – OBTENER INFORMACIÓN TURÍSTICA	169
ILUSTRACIÓN 276. FRONTEND – OBTENER INFORMACIÓN SOBRE EL EVENTO	170
ILUSTRACIÓN 277. FRONTEND – OBTENER COMITÉ.....	170
ILUSTRACIÓN 278. FRONTEND – OBTENER EDICIONES	170
ILUSTRACIÓN 279. FRONTEND – OBTENER INVESTIGADORES.....	171
ILUSTRACIÓN 280. FRONTEND – ENVIAR EMAIL DE CONTACTO.....	171
ILUSTRACIÓN 281. FRONTEND – CONFIRMAR ENVÍO DE EMAIL	171
ILUSTRACIÓN 282. FRONTEND – CAMBIAR EXPOSITOR.....	172
ILUSTRACIÓN 283. FRONTEND – HABILITAR PANTALLA DE REGISTRO	172
ILUSTRACIÓN 284. FRONTEND – CARGAR TIPO DE INSCRIPCIÓN	172
ILUSTRACIÓN 285. FRONTEND – CARGAR TIPOS DE PAGOS	173
ILUSTRACIÓN 286. FRONTEND – CARGAR IDIOMAS.....	173
ILUSTRACIÓN 287. FRONTEND – CARGAR TIPOS DE PAPER.....	173
ILUSTRACIÓN 288. FRONTEND – CARGAR PARALELOS.....	173
ILUSTRACIÓN 289. FRONTEND – CARGAR SEMESTRES	174
ILUSTRACIÓN 290. FRONTEND – CARGAR CARRERAS.....	174
ILUSTRACIÓN 291. FRONTEND – CARGAR TIPO DE IDENTIFICACIÓN	174
ILUSTRACIÓN 292. FRONTEND – CAMBIAR TIPO DE IDENTIFICACIÓN	174
ILUSTRACIÓN 293. FRONTEND – CAMBIAR TIPO DE DOCUMENTO DE IDENTIDAD	175
ILUSTRACIÓN 294. FRONTEND – CAMBIAR SEMESTRES	175
ILUSTRACIÓN 295. FRONTEND – CAMBIAR CARRERA.....	175
ILUSTRACIÓN 296. FRONTEND – CONFIRMAR LENGUAJE DEL CORREO	175
ILUSTRACIÓN 297. FRONTEND – CAMBIAR TIPO DE INSCRIPCIÓN.....	176
ILUSTRACIÓN 298. FRONTEND – CAMBIAR DE FORMULARIO	176
ILUSTRACIÓN 299. FRONTEND – VERIFICAR LOS DATOS DEL REGISTRO	177
ILUSTRACIÓN 300. FRONTEND – ELIMINAR AUTOR DEL ARRAY	177
ILUSTRACIÓN 301. FRONTEND – CARGAR DATOS DE LOS INSCRITOS 1	178
ILUSTRACIÓN 302. FRONTEND – CARGAR DATOS DE LOS INSCRITOS 2	178
ILUSTRACIÓN 303. FRONTEND – VERIFICAR SI EL PARTICIPANTE YA SE ENCUENTRA INSCRITO.....	179
ILUSTRACIÓN 304. FRONTEND – VALIDAR CEDULA	179
ILUSTRACIÓN 305. FRONTEND – VALIDAR EMAIL.....	180
ILUSTRACIÓN 306. FRONTEND – GUARDAR PERSONA	180
ILUSTRACIÓN 307. FRONTEND – GUARDAR INSCRIPCIÓN	181
ILUSTRACIÓN 308. FRONTEND – HABILITAR PÁGINA DE VALIDACIÓN	181
ILUSTRACIÓN 309. FRONTEND – BUSCAR INSCRIPCIÓN	182

ILUSTRACIÓN 310. FRONTEND – MENSAJE GENERAL REGISTRO	182
ILUSTRACIÓN 311. FRONTEND – RENOMBRAR ARCHIVO.....	182
ILUSTRACIÓN 312. FRONTEND – SELECCIONAR IMAGEN.....	183
ILUSTRACIÓN 313. FRONTEND – ENVIAR DATOS.....	183
ILUSTRACIÓN 314. FRONTEND – VERIFICAR SI EXISTE UN COMPROBANTE SUBIDO .	184
ILUSTRACIÓN 315. FRONTEND – OBTENER INSCRIPCIÓN.....	184
ILUSTRACIÓN 316. FRONTEND – OBTENER CERTIFICADOS DE AUTORES.....	184
ILUSTRACIÓN 317. FRONTEND – OBTENER CERTIFICADOS DE EXPOSITORES.....	185
ILUSTRACIÓN 318. FRONTEND – OBTENER FACTURA SEGÚN CEDULA.....	185
ILUSTRACIÓN 319. FRONTEND – VISUALIZAR FACTURA.....	185
ILUSTRACIÓN 320. FRONTEND – GENERAR REPORTE PAPERS	186
ILUSTRACIÓN 321. FRONTEND – FILTRO INSCRITOS.....	186
ILUSTRACIÓN 322. FRONTEND – FILTRO COMPROBANTES.....	187
ILUSTRACIÓN 323. FRONTEND – FILTRO TIPO DE PAGO	187
ILUSTRACIÓN 324. FRONTEND – FILTRO TIPO DE CARRERA	187
ILUSTRACIÓN 325. FRONTEND – FILTRO TIPO DE SEMESTRE.....	188
ILUSTRACIÓN 326. FRONTEND – FILTRO TIPO DE PARALELO	188
ILUSTRACIÓN 327. FRONTEND – CAMBIAR ESTADO DE LOS INSCRITOS.....	188
ILUSTRACIÓN 328. FRONTEND – SELECCIONAR ARCHIVO.....	189
ILUSTRACIÓN 329. FRONTEND – GUARDAR FACTURA	189
ILUSTRACIÓN 330. FRONTEND – ABRIR MODAL FACTURA.....	189
ILUSTRACIÓN 331. FRONTEND – SUBIR FACTURA POR LOTES.....	190
ILUSTRACIÓN 332. FRONTEND – GUARDAR PATROCINADORES.....	190
ILUSTRACIÓN 333. FRONTEND – ACTUALIZAR ESTADO DEL PATROCINADOR	190
ILUSTRACIÓN 334. FRONTEND – EDITAR DATOS PATROCINADORES.....	191
ILUSTRACIÓN 335. FRONTEND – GUARDAR INFORMACIÓN DEL REGISTRO	191
ILUSTRACIÓN 336. FRONTEND – ACTUALIZAR ESTADO REGISTRO.....	191
ILUSTRACIÓN 337. FRONTEND – EDITAR DATOS DEL REGISTRO	192
ILUSTRACIÓN 338. FRONTEND – GUARDAR EDICIÓN	192
ILUSTRACIÓN 339. FRONTEND – GUARDAR PAPER	193
ILUSTRACIÓN 340. FRONTEND – ACTUALIZAR ESTADO DEL SEMESTRE	193
ILUSTRACIÓN 341. FRONTEND – ACTUALIZAR ESTADO DE LA CARRERA.....	193
ILUSTRACIÓN 342. FRONTEND – ACTUALIZAR ESTADO TIPO DE INSCRIPCIÓN	194
ILUSTRACIÓN 343. FRONTEND – GUARDAR TIPO DE INSCRIPCIÓN	194
ILUSTRACIÓN 344. FRONTEND – ACTUALIZAR ESTADO DEL IDIOMA.....	194
ILUSTRACIÓN 345. FRONTEND – GUARDAR ESTADO DEL IDIOMA	195
ILUSTRACIÓN 346. FRONTEND – GUARDAR CARRERA.....	195
ILUSTRACIÓN 347. FRONTEND – GUARDAR COMITÉ.....	196
ILUSTRACIÓN 348. FRONTEND – MODIFICAR INSCRIPCIÓN	196
ILUSTRACIÓN 349. FRONTEND – GUARDAR INVESTIGADOR.....	197

ILUSTRACIÓN 350. FRONTEND – OBTENER PDF	198
ILUSTRACIÓN 351. FRONTEND – GUARDAR INFORMACIÓN CONGRESO	198
ILUSTRACIÓN 352. FRONTEND – GUARDAR IMAGEN DE PORTADA.....	199
ILUSTRACIÓN 353. FRONTEND – CONTAR TIPOS DE INSCRITOS	199
ILUSTRACIÓN 354. FRONTEND – GUARDAR GALERÍA LUGAR	200
ILUSTRACIÓN 355. FRONTEND – GUARDAR INFORMACIÓN DE LA GALERÍA	200
ILUSTRACIÓN 356. FRONTEND – GUARDAR INFORMACIÓN DE LA PRESENTACIÓN ...	201
ILUSTRACIÓN 357. FRONTEND – GUARDAR LUGAR DEL EVENTO	201
ILUSTRACIÓN 358. FRONTEND – GUARDAR INFORMACIÓN TURÍSTICA.....	202
ILUSTRACIÓN 359. FRONTEND – GUARDAR PROGRAMA.....	202
ILUSTRACIÓN 360. FRONTEND – GUARDAR ENVIÓ DE TRABAJOS.....	203
ILUSTRACIÓN 361. FRONTEND – GUARDAR FORMATOS DE ENVÍO DE TRABAJOS	203
ILUSTRACIÓN 362. FRONTEND – GUARDAR FECHAS DE ENVÍOS DE TRABAJO	204
ILUSTRACIÓN 363. FRONTEND – CAMBIAR ESTADO DEL COMITÉ.....	204
ILUSTRACIÓN 364. FRONTEND – GUARDAR TEMARIO	205
ILUSTRACIÓN 365. FRONTEND – GUARDAR TEMAS DEL TEMARIO	205
ILUSTRACIÓN 366. FRONTEND – GUARDAR DETALLES DEL CRONOGRAMA	206
ILUSTRACIÓN 367. FRONTEND – GUARDAR DÍA DEL CRONOGRAMA	206
ILUSTRACIÓN 368. FRONTEND – GUARDAR GALERÍA DEL INICIO	207
ILUSTRACIÓN 369. FRONTEND – ENVIAR CORREO DE FORMA INDIVIDUAL	207
ILUSTRACIÓN 370. FRONTEND – GUARDAR NUEVO USUARIO	208
ILUSTRACIÓN 371. FRONTEND – EDITAR USUARIO	208
ILUSTRACIÓN 372. FRONTEND – ENVIAR CORREO POR LOTES	209
ILUSTRACIÓN 373. FRONTEND – GENERAR CERTIFICADO DE PARTICIPACIÓN.....	209
ILUSTRACIÓN 374. FRONTEND – GENERAR CERTIFICADOS DE PARTICIPACIÓN POR LOTES.....	210
ILUSTRACIÓN 375. FRONTEND – GENERAR CERTIFICADO DE ASISTENCIA.....	210
ILUSTRACIÓN 376. FRONTEND – GENERAR CERTIFICADOS DE ASISTENCIA POR LOTES	211
ILUSTRACIÓN 377. FRONTEND – GENERAR CERTIFICADO COMITÉ.....	211
ILUSTRACIÓN 378. FRONTEND – GENERAR CERTIFICADO EXPOSITOR	212
ILUSTRACIÓN 379. FRONTEND – GENERAR CERTIFICADO DE EXPOSITOR	212
ILUSTRACIÓN 380. FRONTEND – GENERAR CERTIFICADO DE AUTOR	212
ILUSTRACIÓN 381. FRONTEND – GENERAR CERTIFICADOS DE AUTORES POR LOTES	213
ILUSTRACIÓN 382. FRONTEND – GENERAR CERTIFICADOS DE COMITÉ POR LOTES ..	213
ILUSTRACIÓN 383: INICIO CSEI 2022	220
ILUSTRACIÓN 384: REGISTRO Y VALIDACIÓN CSEI 2022	220
ILUSTRACIÓN 385: FACTURACIÓN CSEI 2022	221
ILUSTRACIÓN 386: FIN CSEI 2022 (PROCESOS ADMINISTRATIVOS).....	221
ILUSTRACIÓN 387: FIN CSEI 2022 (PROCESOS PARTICIPANTES).....	221

ÍNDICE DE TABLAS

TABLA 1: VENTAJAS Y DESVENTAJAS DE UNA PWA	15
TABLA 2: CARACTERÍSTICAS DE ANGULAR	20
TABLA 30: COMPARATIVA DE FRAMEWORKS DE DESARROLLO PWA.....	45
TABLA 41: DEFINICIÓN DE LOS ROLES	50
TABLA 5: CUADRO DE ACTIVIDADES Y PROCESOS DEL DESARROLLO DEL CSEI.	50
TABLA 62. PLANTILLA DE HISTORIAS DE USUARIO.....	54
TABLA 7. HISTORIA DE USUARIO - ESTRUCTURAR EL PROYECTO	55
TABLA 8. HISTORIA DE USUARIO - DISEÑO DE LA BASE DE DATOS	55
TABLA 9. HISTORIA DE USUARIO - DISEÑO DE INTERFAZ MÓDULO INFORMATIVO	56
TABLA 10. HISTORIA DE USUARIO - DISEÑO DE INTERFAZ MÓDULO DE REGISTRO.....	56
TABLA 11. HISTORIA DE USUARIO - DISEÑO DE INTERFAZ MÓDULO DE VALIDACIÓN	57
TABLA 12. HISTORIA DE USUARIO - DISEÑO DE INTERFAZ MÓDULO DE FACTURAS ...	57
TABLA 13. HISTORIA DE USUARIO - DISEÑO DE INTERFAZ MÓDULO DE CERTIFICADOS	58
TABLA 14. HISTORIA DE USUARIO - DISEÑO DE INTERFAZ MÓDULO DE CONFIGURACIÓN	58
TABLA 15. HISTORIA DE USUARIO - INICIO DE SESIÓN	59
TABLA 16. HISTORIA DE USUARIO - DESARROLLO MÓDULO INFORMATIVO.	59
TABLA 17. HISTORIA DE USUARIO - DESARROLLO MÓDULO DE REGISTRO	60
TABLA 18. HISTORIA DE USUARIO - DESARROLLO MÓDULO DE VALIDACIÓN.....	60
TABLA 19. HISTORIA DE USUARIO - DESARROLLO MÓDULO DE FACTURAS	61
TABLA 20. HISTORIA DE USUARIO - DESARROLLO MÓDULO DE CERTIFICADOS	61
TABLA 21. HISTORIA DE USUARIO - DESARROLLO MÓDULO DE CONFIGURACIÓN.....	62
TABLA 22. ESTIMACIÓN DE HISTORIAS DE USUARIO	63
TABLA 23. PLANTILLA DE TAREAS DE HISTORIAS DE USUARIO	64
TABLA 24. DESCRIPCIÓN DE LAS TAREAS	64
TABLA 25. TAREA - ESTRUCTURAR EL PROYECTO	64
TABLA 26. TAREA - DISEÑO DE LA BASE DE DATOS	65
TABLA 27. TAREA - DISEÑO INICIO	65
TABLA 28. TAREA - DISEÑO PÁGINA SOBRE EL EVENTO.....	65
TABLA 29. TAREA - DISEÑO PÁGINA CONFERENCISTAS.....	66
TABLA 30. TAREA - DISEÑO PÁGINA COMITÉ	66
TABLA 31. TAREA - DISEÑO PÁGINA PROGRAMA.....	66
TABLA 32. TAREA - DISEÑO PÁGINA ENVIÓ DE TRABAJOS.....	66
TABLA 33. DISEÑO PÁGINA CONTACTO	67
TABLA 34. TAREA - DISEÑO PÁGINA EDICIONES ANTERIORES.....	67
TABLA 35. TAREA - DISEÑO INTERFAZ MÓDULO DE REGISTRO	67
TABLA 36. TAREA - DISEÑO INTERFAZ MÓDULO DE VALIDACIÓN.....	68

TABLA 37. TAREA - DISEÑO INTERFAZ MÓDULO DE FACTURAS.....	68
TABLA 38. TAREA - DISEÑO INTERFAZ MÓDULO DE CERTIFICADOS	68
TABLA 39. TAREA - DISEÑO INTERFAZ CONTROL DE INSCRIPCIONES.....	69
TABLA 40. TAREA - DISEÑO INTERFAZ MÓDULO DE REPORTES.....	69
TABLA 41. TAREA - DISEÑO DE INTERFAZ MÓDULO DE GENERACIÓN DE CERTIFICADOS.....	70
TABLA 42. TAREA - DISEÑO DE INTERFAZ MÓDULO DE CONFIGURACIÓN	70
TABLA 43. TAREA - DISEÑO DE INTERFAZ MÓDULO DE COMITÉ	70
TABLA 44. TAREA - DISEÑO DE INTERFAZ MÓDULO DE USUARIOS	71
TABLA 45. TAREA - INICIO DE SESIÓN FRONT-END.....	71
TABLA 46. TAREA - INICIO DE SESIÓN BACK-END.....	71
TABLA 47. TAREA - DESARROLLO PÁGINA INFORMATIVA INICIO.....	72
TABLA 48. TAREA - DESARROLLO PÁGINA INFORMATIVA SOBRE EL EVENTO	72
TABLA 49. TAREA - DESARROLLO PÁGINA INFORMATIVA COMITÉ.....	72
TABLA 50. TAREA - DESARROLLO DE PÁGINA INFORMATIVA PROGRAMA	73
TABLA 51. TAREA - DESARROLLO DE PÁGINA INFORMATIVA ENVIÓ DE TRABAJOS ...	73
TABLA 52. TAREA - DESARROLLO PÁGINA INFORMATIVA CONTACTO	74
TABLA 53. TAREA - DESARROLLO PÁGINA INFORMATIVA EDICIONES ANTERIORES ..	74
TABLA 54. TAREA - DESARROLLO DE MODULO REGISTRO.....	74
TABLA 55. TAREA - DESARROLLO MÓDULO VALIDACIÓN	75
TABLA 56. TAREA - DESARROLLO MÓDULO FACTURAS	75
TABLA 57. TAREA - DESARROLLO MÓDULO CERTIFICADOS.....	75
TABLA 58. TAREA - DESARROLLO SUBMÓDULO CONTROL DE INSCRIPCIONES	76
TABLA 59. TAREA - DESARROLLO SUBMÓDULO REPORTES.....	76
TABLA 60. TAREA - DESARROLLO SUBMÓDULO CERTIFICADOS	76
TABLA 61. TAREA - DESARROLLO SUBMÓDULO COMITÉ	77
TABLA 62. TAREA - DESARROLLO SUBMÓDULO GESTIÓN USUARIOS.....	77
TABLA 63. TAREA - DESARROLLO SUBMÓDULO CONFIGURACIÓN SISTEMA.....	78
TABLA 64. PLANIFICACIÓN DE ENTREGA	78
TABLA 65. ESTIMACIÓN DE LA PRIMERA ITERACIÓN	80
TABLA 66. ESTIMACIÓN DE LA SEGUNDA ITERACIÓN	82
TABLA 67. ESTIMACIÓN DE LA TERCERA ITERACIÓN.....	86
TABLA 68. ESTIMACIÓN DE LA CUARTA ITERACIÓN.....	88
TABLA 69. ESTIMACIÓN DE LA QUINTA ITERACIÓN.....	95
TABLA 70. ESTIMACIÓN DE LA SEXTA Y SÉPTIMA ITERACIÓN	99
TABLA 71. PRUEBA DE ACEPTACIÓN 001	214
TABLA 72. PRUEBA DE ACEPTACIÓN 002	214
TABLA 73. PRUEBA DE ACEPTACIÓN 003	215
TABLA 74. PRUEBA DE ACEPTACIÓN 004	215
TABLA 75. PRUEBA DE ACEPTACIÓN 005	216

TABLA 76. PRUEBA DE ACEPTACIÓN 006	217
TABLA 77. PRUEBA DE ACEPTACIÓN 007	217
TABLA 78: CUADRO DE ACTIVIDADES Y PROCESOS DEL CSEI CON LA IMPLEMENTACIÓN DEL SITIO WEB.....	218

RESUMEN EJECUTIVO

El presente proyecto de investigación que lleva por tema “SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL”. Tiene por objetivo Desarrollar un sitio web para la administración de congresos internacionales en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

Para este proyecto se aplicó una entrevista a los organizadores y una encuesta a usuarios afines al sistema que revelaron necesidades de estos con el sitio web. Con los resultados obtenidos, se ha determinado todos los requerimientos necesarios para correcto desarrollo e implementación del sitio web enfocado a congresos internacionales que funcione con calidad y seguridad para el uso de la universidad.

El sitio web fue desarrollado utilizando el framework Angular que se adaptó de mejor manera a las necesidades del Sitio Web y que en definitiva es la opción con mejores características tanto en superioridad compatibilidad y seguridad para el desarrollo e implementación del sitio web, para congresos donde los usuarios van a tener facilidad de acceso a toda la información del evento y registro en este.

La metodología seleccionada para el desarrollo de este sitio web fue XP (Xtreme Programming), esta metodología ágil se centra en los datos más importantes evitando en la medida de lo posible la documentación innecesaria, de esta forma se puede avanzar rápidamente con el desarrollo del proyecto.

Palabras clave: Sitio web, administración de congresos internacionales, Angular, CSEI, XP.

ABSTRACT

This research project has as its theme "WEB SITE FOR THE ADMINISTRATION OF INTERNATIONAL CONFERENCES IN THE FACULTY OF SYSTEMS, ELECTRONICS AND INDUSTRIAL ENGINEERING". Its objective is to develop a website for the administration of international congresses in the Faculty of Systems, Electronic and Industrial Engineering. from the Technical University of Ambato.

For this project, an interview was applied to the organizers and a survey of users related to the system that revealed their needs with the website. With the results obtained, all the necessary requirements have been determined for the correct development and implementation of the website focused on international congresses that works with quality and security for the use of the university.

The website was developed using the Angular framework that was best adapted to the needs of the Website and which is the option with the best features in terms of superiority, compatibility, and security for the development and implementation of the website, for congresses where Users will have easy access to all event information and registration in it.

The methodology selected for the development of this website was XP (Xtreme Programming), this agile methodology focuses on the most important data, avoiding unnecessary documentation as far as possible, in this way you can quickly advance with the development of the project.

Keywords: Website, international conference administration, engineering, Angular, CSEI, XP.

INTRODUCCIÓN

El presente trabajo de investigación denominado “SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL.” Está dividido en los siguientes capítulos.

CAPITULO I: “MARCO TEORICO”, se presenta la problemática del CSEI, justificación del proyecto, contextualización del problema a nivel maso, meso y micro de trabajos relacionados con la problemática a satisfacer, además de la fundamentación teórica necesaria para el desarrollo del presente proyecto.

CAPITULO II: “METODOLOGIA”, en este capítulo se presentan los materiales a utilizarse, el procesamiento y análisis de datos y se definen las etapas de desarrollo del proyecto.

CAPITULO III: “RESULTADOS Y DISCUIONES”, se presentan los resultados obtenidos, se detalla el desarrollo del proyecto de investigación basado en la metodología de desarrollo.

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

CAPÍTULO I

1 MARCO TEORICO

1.1 Tema de Investigación

SITIO WEB PARA LA ADMINISTRACIÓN DE CONGRESOS INTERNACIONALES EN LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL.

1.2. Antecedentes investigativos

Las herramientas y tecnologías de desarrollo de sistemas web, han sido estudiadas y analizadas a lo largo de los años, desde la aparición de los primeros lenguajes de desarrollo web hasta los tiempos actuales, los avances tecnológicos y aparición de nuevas tecnologías obligan a desarrolladores web a estar en un constante aprendizaje.

En [1], investigaron las tendencias tecnológicas, lenguajes de programación, herramientas y plataformas para el desarrollo Web determinando lo siguiente:

“El mejor lenguaje para el desarrollo de aplicaciones será aquel que mejor encaje en las preferencias de cada usuario o de la empresa, y sirva mejor al fin que se persigue. Por lo general, el costo, portabilidad, tecnología y seguridad son los cuatro aspectos más importantes que permiten definir que lenguaje es el indicado para el desarrollo de aplicaciones Web”.

En [2], realizaron una comparación entre Angular, React y Vue.js concluyendo que “Las decisiones a favor y en contra de una determinada tecnología dependen en gran medida en el caso de uso y otras circunstancias variables: tamaños de proyectos, trabajadores, conocimientos, experiencias previas y términos como estos afectan este proceso”, además resumió sus extensas comparaciones en casos de uso relevantes presentando el siguiente conjunto de recomendaciones:

- Alta apreciación de TypeScript ⇒ Angular
- Enfatizar la orientación y la estructura en todos los proyectos ⇒ Angular
- Viniendo de un fondo de programación orientada a objetos ⇒ Angular
- Alta importancia de la flexibilidad ⇒ React o Vue

- Gran escala de aplicaciones ⇒ Los tres
- Proceso de aprendizaje inicial superficial ⇒ Vue
- Énfasis en el uso de las tecnologías más nuevas y populares ⇒ Vue
- Gran ecosistema ⇒ Reaccionar
- Separación de preocupaciones en un archivo ⇒ Vue
- Diseñadores requeridos para trabajar con código HTML ⇒ Angular o Vue
- Fuerte enfoque en el uso de JavaScript ⇒ React

En [3], evaluaron la facilidad de aprendizaje y usabilidad de tres frameworks JavaScript concluyendo:

“En este trabajo se ha analizado la usabilidad de tres frameworks JavaScript, conocidos y utilizados en el mundo del desarrollo web, donde el que mejor resultado ha obtenido ha sido Angular, seguido de Ember y de Backbone en último lugar. Se ha mostrado que la usabilidad, un factor muchas veces infravalorado en el área de desarrollo, podría tener un papel importante en la adopción de un framework JavaScript”.

El autor de [4], señala que “Desde el punto de vista de la metodología, se prefieren las ágiles, siendo los principales modelos el Extreme Programming (XP) y Scrum. Sumado al interés por aquellas que permiten flexibilidad, tiempo y costos menos elevados”, de este antecedente se toma en cuenta la preferencia del autor por metodologías ágiles para el desarrollo de aplicaciones web por sus características.

César Quispe Calsín en [5], resalta que “La metodología de desarrollo de software XP, elegida para la presente investigación, ha permitido determinar los requisitos necesarios para la implementación del sistema de tramite documentario y gestión administrativa.” siendo el levantamiento de requerimientos un componente esencial a la hora de desarrollar un proyecto se toma en cuenta la preferencia del autor por la metodología ágil XP para el desarrollo de su sistema web.

Según Arauz en su trabajo [6], concluyen que:

- La herramienta conocida como aplicativo web brinda la oportunidad de introducir procesos de administración de un alto nivel, de una manera sistemática colaborando con el control organizacional e institucional, ingresando desde cualquier dispositivo tecnológico podremos realizar procesos, sin estar presente ayudando de forma eficaz y brindando nuevas alternativas.
- La implementación del aplicativo web sirvió como herramienta de agilizar los procesos de las asociaciones con un mayor control de producción en cada una de sus metodologías, y a su vez adquirieron un mayor crecimiento de producción agrícola en el mercador competitivo.

En la facultad de ingeniería en Sistemas, Electrónica e Industrial, se han desarrollado diferentes proyectos investigativos con la finalidad de satisfacer necesidades de esta, Vallejo Cueva en [7] implemento una solución a un problema interno de la facultad, concluyendo que si software ayudó a facilitar la gestión y seguimiento del proceso de titulación de la facultad, siendo de gran aporte para la facultad y resolviendo un problema real de esta.

1.2.1 Contextualización del problema

El uso de la web como medio de consulta, difusión y administración de información ha experimentado un exponencial crecimiento. Esto ha provocado que el desarrollo de software web incremente su popularidad y se desarrollen soluciones web cada vez más complejas, visualmente modernas y atractivas que ofrecen diferentes herramientas y funcionalidades a sus usuarios. Para resaltar entre los usuarios existe la necesidad de adaptarse a las nuevas tendencias de desarrollo web.

Los avances tecnológicos provocan una constante necesidad de adaptarse a cambios, lenguajes como JavaScript son de los más extendidos y utilizados actualmente para el desarrollo de sistemas web. La mayoría de las tecnologías web incluyen funcionalidades a nivel de frontend y backend con lenguajes como JavaScript o

TypeScript. Los beneficios de estos lenguajes de programación se han visto replicados en aplicaciones web y móviles con tecnologías como Angular, IONIC y React native.

La Facultad de Ingeniería en Sistemas, Electrónica e Industrial ha venido desarrollando el evento denominado “Congreso de Ciencias de la Computación, Electrónica e Industrial, CSEI” [8], que en sus tres últimas ediciones ha contado con un sitio web para la difusión de información, mismo que presenta páginas estáticas, las que al no contar con un panel de administración, se tiene que modificar directamente desde el código del sitio, además sitio web integra un módulo de inscripción de participantes que no permite detallar la información de pago y datos de facturación de los usuarios, lo cual genera retrasos en el proceso de validación. El administrador cuenta con un módulo que carece de herramientas para la modificación de registros, provocando que este tenga que acceder a la base de datos para modificar la información, dicho módulo a su vez no permite la carga de facturas y generación de reportes. Existe la necesidad de mejorar, innovar y adaptarse a nuevos requerimientos de manejo y difusión de información.

Un sistema web correctamente desarrollado, es una herramienta que mejora la forma en la cual se realiza una actividad, razón por la cual es importante el desarrollo de un nuevo sitio web para la automatización de procesos en la administración de congresos internacionales.

1.2.2 Justificación

La Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, desde 2013 bajo el nombre de “Jornadas Internacionales de Investigación Científica” y desde 2019 con el nombre “Congreso de Ciencia de la Computación, Electrónica e Industrial CSEI”, ha venido desarrollando un espacio de difusión científica. Los miembros del comité organizador CSEI requieren de un nuevo sitio web para difundir la información de ediciones pasadas y futuras del evento científico, así como para mejorar la administración de los procesos que conllevan la realización de un congreso internacional. El desarrollo e implementación de un nuevo sitio web mejorará la administración del evento durante su desarrollo y presentará la

información de este de forma visualmente atractiva a las personas que estén interesadas en el congreso.

Los beneficios de los sistemas web, sus características y prestaciones, hacen que sea una tecnología adaptable a cualquier necesidad. Las innovaciones en tecnologías de desarrollo web dan la pauta para la creación de nuevos y más complejos sistemas, con funcionalidades que cumplan todo tipo de necesidades e interfaces modernas, además de brindar comodidad a los usuarios durante su interacción.

El desarrollo del nuevo sitio web tiene como fin, dotar de herramientas que le permitan al administrador: publicar, editar y eliminar información del congreso actual, así como de ediciones pasadas de este, visualizar el número de usuarios que visitaron el sitio web y cantidad de inscritos, modificar datos de usuarios inscritos directamente desde el sistema sin la necesidad de acceder a la base de datos, validar la inscripción con la visualización detallada del método de pago, cargar facturas de inscritos y generar reportes del número de registrados, inscritos e ingresos generados.

Además, el sitio web permitirá a los usuarios consultar información dinámica del congreso, llevar a cabo su inscripción en una interfaz multi idioma, detallar la información de su forma de pago y datos de facturación, siendo posible también descargar su factura.

Con la información expuesta previamente se determinó que el presente proyecto de investigación es de gran interés para los Organizadores del Congreso Internacional CSEI y la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, ya que con el nuevo sitio web tanto usuarios como administrador se beneficiaran de este y sus prestaciones.

1.2.3 Delimitación

1.2.3.1 De contenidos

Área Académica: Software

Línea de Investigación: Desarrollo de Software.

Sub línea de investigación: Aplicaciones Web.

1.2.3.2 Espacial

Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

1.2.3.3 Temporal

La presente investigación se desarrollará en el período septiembre 2022 – marzo 2023.

1.2.4 Fundamentación teórica

Software

Según el estándar 729 de IEEE, software se define como: “El conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación” [9]. El termino hace referencia a un programa o conjunto de programas, datos, procesos, rutinas y reglas que permiten a un hardware realizar determinadas tareas. Software, es el componente no físico que forma parte de dispositivos como teléfonos inteligentes, electrodomésticos, vehículos, etc. Software también es considerado un producto comercializable.

Software Libre

Según Free Software Foundation, el *Software Libre* Esta basado en la cooperación y la transparencia, respeta la libertad de los usuarios para obtenerlo, copiarlo, estudiarlo, modificarlo y redistribuirlo libremente [10]. Software libre y gratuito no es lo mismo, los productos de software gratuitos permiten su utilización sin coste, pero no otorga a sus usuarios la posibilidad de estudiarlo o modificarlo.

El “software libre” es cuestión de libertad, no de precio [10]. El acceso al código fuente es la característica que más resalta de este tipo de software. Software libre es cualquier software que otorgue estas libertades:

- Ejecutar el programa sea cual sea nuestro propósito.

- Estudiar el funcionamiento del programa y adaptarlo a tus necesidades.
- Redistribuir copias y ayudar.
- Mejorar el programa y luego publicarlo para el bien de toda la comunidad.

Páginas web

Es un documento digital que alberga información, ya sea textual, gráfica, visual o sonora, que está en Internet y es accesible gracias al protocolo HTTP [11]. Se dividen en dos grupos estáticas y dinámicas.

Páginas web estáticas

Las páginas web estáticas se caracterizan por no incorporar una base de datos, presentar información fija y tener tiempos de carga muy cortos, el costo y tiempo de desarrollo es inferior al de páginas dinámicas [12].

Páginas web dinámicas

Las páginas web dinámicas se caracterizan por permitir la interacción con el usuario en tiempo real, incorporar una base de datos para el almacenamiento y manipulación de datos, presentar información, son interactivas y funcionales. Para este tipo de páginas se emplean dos tipos de programaciones Front-end y Back-end [13].

Front-end

Se denomina front-end a la parte visual de un sitio web con la que un usuario tiene interacción, también denominado “lado cliente” [14].

Back-end

Se denomina back-end a la parte lógica y funcional de un sitio web [15]. Es el conjunto de acciones que no son visibles directamente por un usuario, responsable del almacenamiento y manipulación de datos está conformado por tiene tres partes: servidor, aplicación y base de datos"

Sitio Web

Un sitio web es un conjunto de páginas web asociadas a un tema en común, incluye una página de inicio que da la bienvenida a los usuarios, los cuales se puede acceder a través de un nombre de dominio y dirección en Internet específicos [16]. El objetivo de un sitio web es mostrar información y la tarea de los usuarios es consumir esta información.

Sistemas Web

Según Mora en el año 2010 se definió a los sistemas web como: “Aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador” [17].

Según Báez en el año 2013, menciona que: “Son aquellos sistemas alojados en internet o una intranet, con la peculiaridad que brindan funcionalidades potentes y más específicas a diferencia de las páginas web” [18].

Un sistema web a diferencia de un sitio web, es una aplicación software que se desarrolla en un lenguaje soportado por los navegadores web ya sean desktop o móvil, se accede de forma global o local y se tiene funcionalidades con las que los usuarios pueden interactuar [19]. El principal objetivo de un sistema web es permitir que los usuarios realicen tareas.

Servicios web y API

Servicios web

La empresa de International Business Machines IBM [20], menciona que:

“Los servicios web son aplicaciones web que ayudan a mejorar la flexibilidad de los procesos empresariales mediante la integración con aplicaciones que, de lo contrario, no se comunican”.

“Son aplicaciones modulares autocontenidas que puede describir, publicar, localizar e invocar a través de una red”.

Servicios web son el conjunto de aplicaciones o tecnologías que interoperan en una red, Existen dos tipos de servicio web:

Servicios web SOAP

Protocolo simple de acceso a objetos, es un protocolo basado en el lenguaje XML, se creó originalmente para posibilitar la comunicación entre las aplicaciones que se diseñaban con diferentes lenguajes y en distintas plataformas [21].

Servicios web REST

Transferencia de estado representacional, es un conjunto de principios arquitectónicos que se ajustan a las necesidades de las aplicaciones móviles y los servicios web ligeros [22].

La arquitectura REST cumple con las siguientes limitaciones:

- Arquitectura cliente-servidor.
- Sistema sin estado.
- Capacidad de almacenamiento en caché.
- Sistema en capas.
- Disponibilidad del código según se solicite.
- Interfaz uniforme.

REST es un conjunto de pautas que ofrece una implementación flexible, mientras que SOAP es un protocolo con requisitos específicos, como en el caso de la mensajería XML [22].

API

Según RedHat: “Una API o interfaz de programación de aplicaciones es un conjunto de definiciones y protocolos que se usa para diseñar e integrar el software de las aplicaciones, permite que sus productos y servicios se comuniquen con otros” [22]. El propósito de las APIs es la integración, las APIs le permiten ofrecer acceso a los recursos, al mismo tiempo mantener la seguridad y el control.

Apis SOAP y RESTful

Las APIs que están diseñadas bajo los protocolos de SOAP usan XML para formatear sus mensajes y reciben solicitudes a través de HTTP o SMTP. Las APIs web que funcionan con la arquitectura REST se llaman API de RESTful. Las APIs SOAP siguen un protocolo mientras que las APIs REST cumplen con las limitaciones de su arquitectura [22].

Servicios web y api similitudes.

- Se puede acceder a ambos a través de HTTP.
- Son un medio de comunicación entre consumidores y proveedores.

Servicios web y API diferencias.

- Los servicios web, requieren de una red para funcionar. Mientras que una API puede funcionar sin conexión.
- Los servicios web utilizan tres estilos de comunicación: SOAP, REST y XML-RPC. Mientras que una API utiliza cualquier estilo de comunicación.
- Los servicios web son compatibles únicamente con XML. Mientras que una API es compatible con XML y JSON.
- Los servicios web proporciona soporte solo para el protocolo HTTP. Mientras que una API proporciona soporte para el protocolo HTTP y HTTPS.

Un servicio web, es una API la cual es utilizada para que dispositivos electrónicos se comuniquen a través de la web. Una API es una interfaz de programación que no está limitada por a un formato de transferencia de datos y no necesariamente está expuesta en la web. Un servicio web es una API, pero una API no necesariamente es un servicio web.

Aplicaciones Web Progresivas

Para entender que es una PWA previamente se debe tener claro lo que son las aplicaciones web y las aplicaciones nativas.

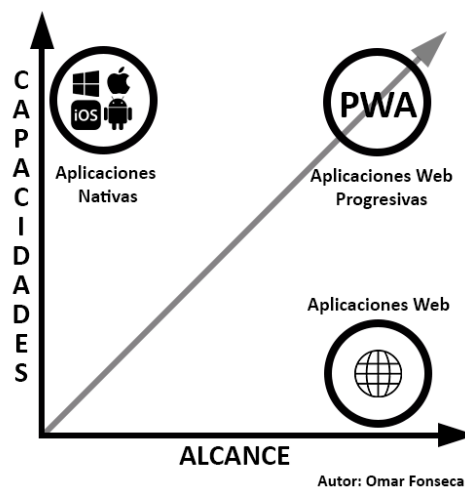
Aplicaciones web. -Son aplicaciones que no dependen de un sistema operativo, funcionan a través de un navegador web.

Aplicaciones Nativas. - Son aplicaciones que están desarrolladas específicamente para funcionar y depender de un sistema operativo.

Teniendo en cuenta estos conceptos, una PWA es una aplicación web que utiliza las últimas tecnologías disponibles en los navegadores para ofrecer en dispositivos móviles una experiencia lo más parecida posible a la de una aplicación nativa [23].

Según Iván Ramírez, “Las aplicaciones web progresivas son una evolución natural de las aplicaciones web que difuminan la barrera entre la web y las aplicaciones, pudiendo realizar tareas que generalmente solo las aplicaciones nativas podían llevar a cabo” [24].

Ilustración 1: PWA



Elaborado por: El investigador

En resumen, las aplicaciones web progresivas son la evolución de las aplicaciones web y las aplicaciones nativas, funcionan como una a una página web, pero también se puede utilizar como las aplicaciones nativas.

Características

Las características de las PWA permiten que se visualicen páginas como aplicaciones nativas, haciendo que el usuario tenga una experiencia similar a que ofrecen las aplicaciones nativas.

Progresiva: Capacidad para adaptarse a las actualizaciones de los navegadores web.

Responsiva: Capacidad para que el diseño de la interfaz se adapte al tamaño de cualquier dispositivo sin importar las dimensiones de su pantalla [25].

Notificaciones Push: Capacidad de enviar notificaciones al usuario ya sea en sistemas móvil o desktop, sin la necesidad de que la pagina este abierta [26].

Acceso offline: Capacidad de seguir funcionando sin acceso a internet, limitando sus funciones hasta retomar la conexión.

Instalación en la pantalla de inicio: Capacidad de ser instalada en la pantalla de inicio emulando ser una aplicación móvil nativa.

Pantallas de splash: Capacidad de mostrar una pantalla inicial al usuario, que hará la funciona de pantalla de carga.

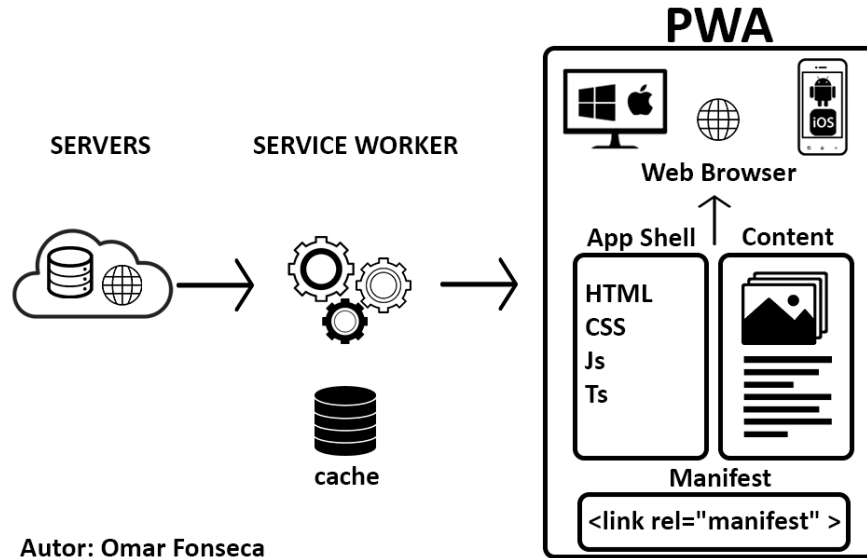
Carga Rápida: Velocidad de carga superior a la de aplicaciones web tradicionales [26].

Segura: Inclusión de un certificado https, para disminuir el riesgo de robo de datos o manipulación de contraseñas [25].

Arquitectura y Componentes

A continuación, se puede observar de forma gráfica el modelo de la arquitectura de una PWA, así como los componentes que la conforman.

Ilustración 2: Arquitectura de una PWA



Elaborado por: El investigador

Service Worker

Considerado como un elemento imprescindible de una PWA, es el motivo principal de las características de las aplicaciones web progresivas [27]. Se trata de uno o varios scripts que funcionan en segundo plano, permiten generar una experiencia fuera de línea descargando en cache los recursos principales para el funcionamiento de la aplicación, también es el responsable del envío de notificaciones push.

App Shell

El Shell de la aplicación contiene todos los archivos Html, Css, Js, Ts, que tienen que cargarse inmediatamente para que la interfaz de usuario funcione, los recursos principales almacenando en cache [28]. Permite que la aplicación funcione sin la necesidad de conexión, funciona en conjunto con el Service Worker y Content para mostrar la información dinámica de la aplicación.

Content

Se refiere a todo el contenido dinámico de la aplicación, es decir, todo aquello que puede cambiar.

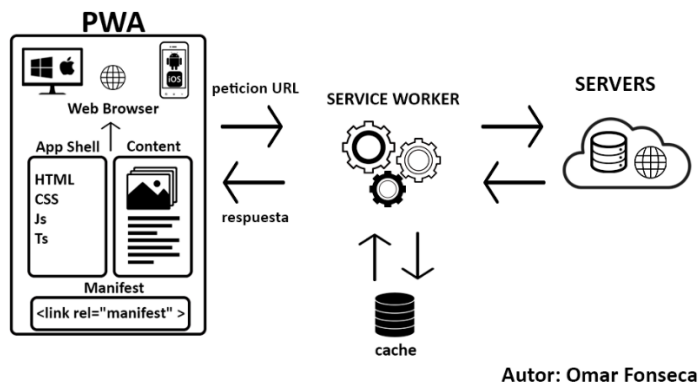
Manifest

Es un archivo de tipo JSON en el cual se definen los aspectos de características visuales y de usabilidad de una PWA [29]. En su contenido se puede encontrar:

- Nombre de la PWA.
- Icono, este se mostrará cuando la PWA sea instalada.
- Colores.
- Orientación.
- Idioma principal.
- Descripción, un breve detalle de la aplicación.
- Página de inicio.
- URL de inicio.
- Display, distintas configuraciones para la pantalla.

Funcionamiento

Ilustración 3: Funcionamiento de una PWA



Autor: Omar Fonseca

Elaborado por: El investigador

El principal componente para el funcionamiento de una PWA es el Service Worker, ya que todo transcurre a través de este. La PWA emite una petición URL, si la solicitud requiere de una conexión a internet el Service Worker se conecta al servidor caso contrario lo hará a cache. Finalmente, el Service Worker envía la respuesta a la PWA.

Ventajas y Desventajas de una PWA

Al ser un punto intermedio entre aplicaciones web tradicionales y aplicaciones nativas las PWA presentan las siguientes ventajas y desventajas:

Tabla 1: Ventajas y Desventajas de una PWA

VENTAJAS	DESVENTAJAS
No requieren de una tienda de aplicaciones para su descarga.	No tienen acceso a todas las funciones de los dispositivos.
Son multiplataforma, cualquier usuario puede agregarla a su pantalla de inicio o escritorio.	No permite crear sitios muy complejos.
Menor coste de desarrollo.	
Actualizaciones automáticas.	

Elaborado por: El investigador

El uso de las PWA seguirá creciendo por sus características y potencial a futuro, existen estudios en los que se ha podido comprobar que la performance de las aplicaciones web progresivas es incluso superior en algunas acciones particulares a las aplicaciones nativas [28]. Las PWA elimina las barreras impuestas por los sistemas operativos permitiendo que los desarrolladores creen soluciones multiplataforma con una única base de código.

Web Framework

Es un esquema o marco de trabajo con una estructura inicial utilizado por desarrolladores para elaborar un proyecto software. También considerado como una

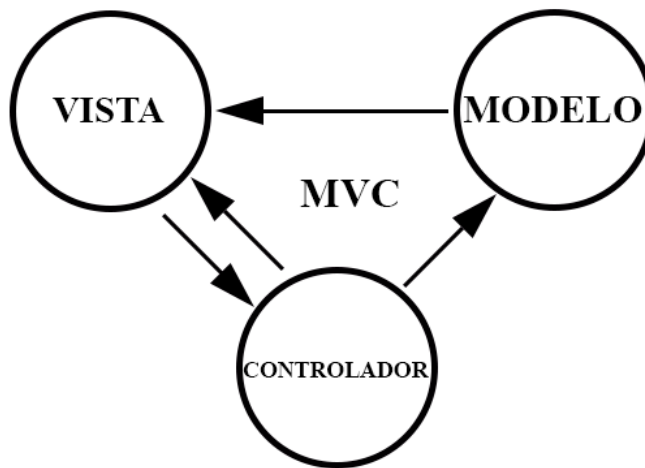
especie de plantilla o conjunto de herramientas y módulos que se pueden utilizar para desarrollar diferentes proyectos.

Framework hace referencia a “una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación” [29].

Arquitectura de un Web Framework

Un web framework tiene una estructura básica que consta de modelo, vista y controlador, también conocida como MVC.

Ilustración 4: Arquitectura de un Web Framework



Autor: Omar Fonseca

Elaborado por: El investigador

El modelo admite el backend y contiene todas las capas de lógica de datos. La Vista admite el frontend se encarga de cómo se ve visualmente la página. El controlador convierte las entradas en comandos [29].

Framework para el desarrollo de Aplicaciones Web Progresivas.

Actualmente existen diferentes frameworks para el desarrollo de aplicaciones web progresivas, entre los que más resaltan están Vue.js, React, Ionic, Flutter y Angular.

Vue.js

“Es un framework progresivo de JavaScript para construir interfaces de usuario. Se basa en HTML, CSS y JavaScript estándar, y proporciona un modelo de programación declarativo y basado en componentes que lo ayuda a desarrollar interfaces de usuario de manera eficiente, ya sea simple o compleja”. Vue.js es un framework de código abierto, que permite desarrollar UI interfaces de usuario y SPA Single Page Application. El desarrollo con esta tecnología está centrado en la capa de Vista es decir el Frontend de una aplicación. “Es sencillo y fácil para entender incluso para desarrolladores no experimentados” [30]. Es una librería que permite agregar comportamientos y funcionalidades a cualquier contexto donde se pueda ejecutar Javascript [31].

React

Es una librería JavaScript de código abierto para crear interfaces de usuario, fue creado en Facebook en el año 2013 para abordar algunos de los desafíos asociados con un sitio web a gran escala basado en datos. Sirve para desarrollar aplicaciones web de una manera más ordenada y con menos código que con la utilización de JavaScript puro o librerías como jQuery [32].

Una de las características principales de este framework es el Virtual DOM, que es una representación del DOM en memoria, utilizado para aumentar el rendimiento de los componentes y aplicaciones Frontend. Su funcionamiento se basa en actualizar los cambios primero en el Virtual DOM, tarea la cual es más rápida y luego compararlo con el DOM del navegador para actualizar únicamente las partes que sean necesarias.

Document Object Model o DOM, es una interfaz de programación para documentos HTML y XML, que permite crear, cambiar, o remover elementos del documento [33].

Ionic

Es un software developer kit de código abierto moderno y multiplataforma para aplicaciones web, iOS y Android que trabaja en conjunto con Angular, React o Vue. Este conjunto de herramientas móviles de código abierto sirve para crear aplicaciones web y aplicaciones nativas multiplataforma de alta calidad. Es una amplia suite de desarrollo de aplicaciones híbridas de código abierto, su app development framework

es solo una parte de su ecosistema, también brindan servicios como herramientas de creación de prototipos, herramientas analíticas y un servicio de notificación automática a través de su plataforma [34].

Este framework requiere de la instalación de Angular, React o Vue, NodeJS, y Apache Cordova para trabajar, la instalación requiere de npm para la administración de paquetes de una forma independiente en sistemas de escritorio.

Entre las características que más resaltan están:

- Permite desarrollar aplicaciones nativas iOS y Android, aplicaciones web progresivas y aplicaciones de escritorio con una única base de código.
- Su diseño es limpio, sencillo y funcional.
- Emplea Capacitor o Cordova para aplicaciones web progresivas.
- Está basado en HTML, CSS y JavaScript.
- Trabaja con Angular, React y Vue.

En resumen, Ionic es un Framework que permite la creación de aplicaciones móviles híbridas basando su construcción en desarrollo web.

Flutter

Según Google “Flutter es un marco de trabajo de código abierto de Google para crear hermosas aplicaciones multiplataforma compiladas de forma nativa a partir de una única base de código” [35].

Este framework tiene tres características principales:

Rapidez

La compilación de Flutter es en código ARM o Intel, así como en JavaScript haciendo que su rendimiento sea rápido en cualquier dispositivo.

Productividad

Con la herramienta Hot Reload, Flutter permite que las actualizaciones de código muestren los cambios de manera casi instantánea sin perder su estado.

La funcionalidad Hot Reload de Flutter ayuda a construir y probar de manera rápida y fácil, interfaces de usuario, añadiendo funcionalidades y corrigiendo errores casi instantáneamente. Hot reload trabaja inyectando ficheros de código fuente actualizados en la Máquina Virtual (VM) Dart en ejecución [35].

Flexibilidad

Permite el control de cada píxel de la interfaz para crear diseños personalizados y adaptables que se vean y se sientan geniales en cualquier pantalla.

Flutter a diferencia de otros frameworks utiliza el lenguaje de programación Dart el cual de igual forma fue creado por Google.

Dart es un lenguaje de programación de propósito general de código abierto. Es puramente orientado a objetos, es un lenguaje con sintaxis de estilo C. Fue desarrollado originalmente por Google en 2011 [36]. Dart está optimizado para construir interfaces de usuario personalizadas y aplicaciones multiplataforma. El código Dart se compila en lenguaje de máquina nativo para la plataforma en la que se está ejecutando la aplicación. Por ejemplo, el código web Dart se compila para JavaScript.

Angular

Es un framework de código abierto creado por Google para el desarrollo de aplicaciones web de una única página también conocidas como SPA o Single Page Aplicación. Angular separa y ordena su frontend y backend evitando que se escriba código repetitivo, permitiendo que la creación de aplicaciones sea rápida y posibilite modificaciones y actualizaciones [37]. Una aplicación web desarrollada con angular inicialmente tendrá una velocidad de carga lenta ya que al ser de tipo SPA la aplicación cargará todo el sitio web y sus recursos la primera vez que se abra, haciendo que la posterior navegación dentro de esta sea muy rápida con transiciones instantáneas.

Angular es un framework Javascript, pero está escrito en TypeScript una extensión del lenguaje [2], El desarrollo en Angular se realiza por medio de TypeScript que ofrece muchas herramientas adicionales al lenguaje base, como el tipado estático o los decoradores. Para el desarrollo de SPA con este framework es importante estar familiarizado con los lenguajes HTML, CSS y JavaScript ya que son la base de Angular.

Características

Tabla 2: Características de Angular

MULTIPLATAFORMA		
PWA	Nativo	Escritorio
Utiliza las capacidades modernas de la plataforma web para ofrecer experiencias similares a las de las aplicaciones. Instalación de alto rendimiento, sin conexión y en cero pasos.	Crea aplicaciones móviles nativas con estrategias de Cordova, Ionic o NativeScript.	Crea aplicaciones instaladas en el escritorio en Mac, Windows y Linux con los mismos métodos Angular que has aprendido para la web, además de la capacidad de acceder a las APIs nativas del sistema operativo.
VELOCIDAD Y RENDIMIENTO		
Generación de código	Universalidad	División de código
Angular convierte sus plantillas en código altamente optimizado para las máquinas virtuales JavaScript de hoy en día, lo que le ofrece todas las ventajas del código escrito a mano con la	Sirve la primera vista de tu aplicación en Node.js®, .NET, PHP y otros servidores para una representación casi instantánea en HTML y CSS.	Las aplicaciones de Angular se cargan rápidamente con el nuevo enrutador de componentes, que ofrece división automática de código para que los usuarios solo carguen el código necesario para representar la vista que solicitan.

productividad de un framework.	También allana el camino para sitios que optimizan para SEO.	
PRODUCTIVIDAD		
Plantillas	Angular CLI	IDEs
Crea rápidamente vistas de interfaz de usuario con una sintaxis de plantilla sencilla y potente.	Herramienta de línea de comando: empieza a construir rápidamente, agregar componentes y pruebas, luego implementar instantáneamente.	Obtén la finalización inteligente del código, errores instantáneos y otros comentarios en editores e IDE populares.
HISTORIA DE DESARROLLO COMPLETO		
Pruebas	Animaciones	Accesibilidad
Karma para pruebas unitarias, puedes saber si has roto las cosas cada vez que ahorras. Protractor hace que sus pruebas de escenario se ejecuten más rápido y de una manera estable.	Crea coreografías complejas y de alto rendimiento y líneas de tiempo de animación con muy poco código a través de la API intuitiva de Angular.	Crea aplicaciones accesibles con componentes habilitados para ARIA, guías para desarrolladores e infraestructura de prueba a11y integrada.

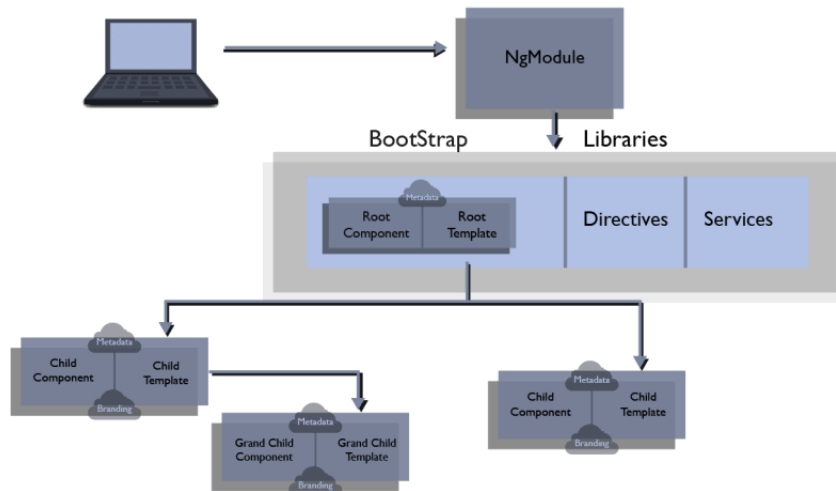
Elaborado por: Angular [37].

Arquitectura y Funcionamiento

En los primeros años del framework Angular, este inicio como Angularjs y empleaba la arquitectura más común entre los frameworks, MVC o modelo vista controlador.

Para su segunda versión Angular ya con TypeScript como lenguaje predominante de segundo su arquitectura inicial y adopto el modelo vista vista modelo MVVM, en el que resalta el concepto del 2-way binding, y tiene como objetivo principal el desacoplamiento de la interfaz y la lógica de negocio.

Ilustración 5: Arquitectura de Angular



Elaborado por: Vanessa Aristizabal [38].

La arquitectura de Angular está basada en la construcción por bloques, estos están organizados en NgModules los que recopilan código relacionado en conjuntos funcionales, las aplicaciones de Angular son un conjunto de NgModules.

Lenguajes de Programación

Con excepción de Flutter la mayoría de frameworks para el desarrollo web están basados en HTML, CSS y JavaScript, para desarrollar un proyecto relacionado con tecnologías web es menester conocer acerca de estas tecnologías.

Html

Es un lenguaje de programación con el que se construye sitios web, Hypertext Markup Language o el lenguaje de marcas hipertexto, sirve para definir el contenido de una página web. Es un lenguaje compuesto por etiquetas con las cuales se definen las partes

del documento web. “Desde los inicios de la Web esta plataforma se ha extendido a diferentes dispositivos como teléfonos o reproductores de música y hoy en día relojes, televisores, hogares y automóviles”

Los documentos HTML contienen una estructura de etiquetas con una organización básica, empezando por su etiqueta raíz <HTML> la cual es la principal y contiene a dos etiquetas:

<head>, etiqueta en la que se incluye la información de cabecera de la página. La información de esta etiqueta no se mostrará en el contenido del cuerpo de la página web, pero servirá como el identificativo del sitio.

<body>, etiqueta en la que se incluye la información del cuerpo, todo lo que se va a ver en el navegador web está dentro de esta sección, ya sea texto, imágenes, animaciones, videos, entre otros recursos se mostraran a través del <body>.

La base de la programación web es HTML, razón por la cual es sumamente necesario tener claro el concepto, funcionamiento, componentes y dominar el lenguaje, los frontend están basados en esta tecnología, así que, se tomara como punto de partida para el desarrollo de cualquier proyecto web.

Css

Es un lenguaje de diseño, con el que se da estilos a un sitio web, Hojas de Estilo en Cascada o Cascading Style Sheets, sirve para modelar el contenido de una página web. Es un lenguaje para definir la forma de presentación de un documento HTML, describe como se presentará los elementos que conforman una página web. Es el lenguaje utilizado para establecer el diseño visual de un documento HTML. Un sitio web por sí solo, no sería más que un documento con información plana y carente de diseño, CSS es el complemento que permite que una página sea visualmente llamativa para los usuarios, su modelo en cascada permite que todos los elementos puedan ser estilizados

de forma grupal o independiente uno de otro. El término cascada se utiliza porque existe una jerarquía para definir qué estilo es el que se le aplica a cada elemento

JavaScript

Es un lenguaje de programación, con el que se le puede aportar dinamismo a una página web, es un lenguaje ligero, interpretado, o compilado, conocido como un lenguaje de secuencia de comandos. JavaScript es un lenguaje de programación con soporte para programación orientada a objetos, imperativa y declarativa. “JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas” [7]. Este lenguaje es muy popular para el desarrollo de sitios web, pero que también es utilizado en entornos alejados de un navegador web como NodeJS.

Node js

Es un entorno en tiempo de ejecución, de código abierto y multiplataforma para servidores, está basado en JavaScript. Node Js se ejecuta en tiempo real incluye todo lo necesario para correr un programa escrito en JavaScript. “Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables”. Permite que el código JS se independice de un navegador para su ejecución, permitiendo que este se pueda ejecutar en un ordenador como si se tratase de una aplicación independiente. Al ser un entorno para la ejecución de código JavaScript, permitirá también que en este se ejecuten líneas de código de TypeScript [39].

TypeScript

Es un lenguaje de programación, libre y de código abierto desarrollado por Microsoft, es un superconjunto de JavaScript que agrega sintaxis adicionales a JavaScript para admitir una integración más estrecha con su editor. “TypeScript es un lenguaje de programación fuertemente tipado que se basa en JavaScript, lo que le brinda mejores herramientas a cualquier escala” [39]. Se puede considerar a TypeScript como una evolución de JavaScript el cual mantiene sus bases, pero lo dota de mejoras. Para su ejecución TypeScript a través de un compilador se traduce a código JavaScript original.

Bases de datos

Es un banco de datos o conjunto de información de un mismo contexto, que se almacena de forma ordenada para su posterior consulta, modificación, eliminación o transmisión. “Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático” [40]. Es una herramienta que permite recopilar, organizar y relacionar datos para realizar consultas rápidas y operaciones de forma rápida.

SQL

Structured Query Language o lenguaje de consulta estructurada, es el lenguaje de consulta de bases de datos más empleado. “SQL es un lenguaje de computación para trabajar con conjuntos de datos y las relaciones entre ellos. SQL es un estándar internacional reconocido por organismos de estándares como ISO y ANSI” [41]. SQL tiene una sintaxis basada en el idioma inglés, tiene cláusulas como SELECT que permite la selección de elementos, su facilidad de entendimiento hace que usuarios expertos e inexpertos puedan trabajar con el lenguaje.

Motor de base de datos

Son programas específicos, que sirven de intermediarios entre bases de datos y aplicaciones que las utilicen, un motor de base de datos debe permitir crear, modificar, configurar y eliminar una base de datos.

MySQL

MySQL es la base de datos de código abierto más popular del mundo, es el gestor de bases de datos más utilizado en el mundo. Es un sistema multiusuario y multihilo empleado en la mayoría de las páginas webs actuales y en aplicaciones de software libre. MySQL se caracteriza por su sencillez de uso y su buen rendimiento, además de su fácil instalación y configuración [42].

Características

- Conectividad segura.
- Uso de claves foráneas.
- Replicación.
- Tablas hash en memorias temporales.
- Uso de multihilos mediante hilos de kernel.
- Es muy estable.
- Permiten hasta 64 índices por tablas.
- Soporta más de 50 millones de registros.
- Búsqueda e indexación de campos de texto.
- Disponible en casi todas las plataformas o sistemas operativos.

Ventajas y Desventajas

Tabla 3: Ventajas y Desventajas de MySQL

VENTAJAS	DESVENTAJAS
Es software libre con licencia GPL.	Falta de documentación oficial.
Bajo consumo de recursos para su ejecución.	
No requiere hardware de alto rendimiento.	Fallos difíciles de detectar.
Alta velocidad de procesamiento.	
Fácil de instalar.	No es el más intuitivo.
Compatible con Windows, Mac y Linux.	
Baja probabilidad de corrupción de datos.	Falta de eficacia en la constante modificación de una DB.
Encriptación de datos para la seguridad.	

Elaborado por: El investigador

Botón de Pago

Es un elemento de pago digital, un botón de pagos es un enlace que se lo encuentra al final de realizar una compra en un sitio web, se lo genera al finalizar una compra y permite que los clientes paguen por un producto o un servicio. Funciona como un intermediario entre empresa y clientes.

Readypago

Es una plataforma asociada con el Banco del Austro, permite que sus clientes ofrezcan un medio de pago en sus sitios web. “Es una herramienta donde puedes Cobrar y Pagar por productos o servicios de una forma ágil, sencilla, segura y desde cualquier lugar que te encuentres” [43].

Metodología de desarrollo

Una metodología de desarrollo de software es un conjunto de métodos y técnicas que establecen una disciplina de trabajo para la creación de productos software, tiene como objetivo organizar de una manera óptima a el equipo de trabajo que va a desarrollar un proyecto.

Metodologías tradicionales

Las metodologías tradicionales tienen como característica, definir y establecer de forma rigurosa los requisitos del proyecto, son metodologías poco flexibles y no permiten realizar cambios, están orientadas en base a la planificación. La organización de trabajo es lineal sus etapas van ocurriendo una tras otra de forma unidireccional, este tipo de metodologías no se adaptan a los cambios.

Metodologías ágiles

En la actualidad este tipo de metodologías, son las más utilizadas, la flexibilidad y agilidad de estas metodologías han popularizado su uso [44], permiten que se pueda adaptar la forma de trabajo de acuerdo con el proyecto que se va a desarrollar. Se considera que son adaptativas, permiten maximizar la satisfacción del cliente. Las modificaciones y gestión de cambios son propios de este tipo de metodologías, están basadas en la metodología incremental, en la que en cada ciclo de desarrollo se van agregando nuevas funcionalidades al resultado final. Este tipo de metodologías permiten conformar equipos de trabajo autosuficientes e independientes.

XP

Programación Extrema, es una metodología de desarrollo ágil utilizada en la gestión de proyectos software, tiene como objetivo potenciar el trabajo en equipo [45],

aprendizaje continuo y buenas relaciones interpersonales del equipo de trabajo destinado a la realización del proyecto. XP es recomendada para proyectos en los cuales los requisitos son imprecisos y cambiantes. Esta metodología ágil es clasificada como evolutiva, está basada en la simplicidad, la comunicación y el reciclado continuo de código.

XP es una metodología muy utilizada actualmente para aplicaciones en procesos de consolidación de una organización se enfoca en mantener asesorados a los empleados brindándoles un soporte técnico y potenciando las comunicaciones y el trabajo grupal para disminuir los tiempos donde no hay actividad.

XP es de uso sencillo y permite al programador tener un soporte para servir al usuario y desarrollar un proyecto de aplicación dividido en módulos organizados de tal manera que es imposible perderse en las diferentes secciones del proyecto también permite que determinados usuarios tengan más acceso que otros hacer administradores o programadores del mismo fomenta el trabajo en más de una forma para que se desarrolle con dos o más elementos es decir la prioridad es el trabajo en equipo para que no solo una persona sea la que realice determinadas actividades incluyendo las de administración por lo cual determina las responsabilidades que van a tener en los equipos de trabajo y así se distribuye de mejor manera la carga impositiva y así se distribuye de mejor manera la carga impositiva sin que haya choques en la distribución de la administración

Características

- Constante comunicación entre cliente y equipo de trabajo.
- Cronograma flexible, planificación flexible.
- Desarrollo a través de fases (planificación, diseños, codificación y pruebas).
- Flujo constante de comunicación.
- Retroalimentación.

Ventajas y Desventajas de XP

Tabla 4: Ventajas y Desventajas de XP

VENTAJAS	DESVENTAJAS
Programación organizada.	Cambios constantes.
Bajo riesgo de errores.	No recomendable para proyectos a largo plazo
Rápida atención al cliente.	No define todos los requisitos al iniciarse el proyecto.
Adaptabilidad.	

Elaborado por: El investigador

Scrum

Es una metodología de desarrollo ágil, es un conjunto de reglas y buenas prácticas para el trabajo colaborativo, tiene como objetivo obtener el mejor resultado posible y maximizar el retorno de inversión para los desarrolladores [46]. Es un modelo de desarrollo iterativo y creciente que normalmente se enfoca a la gestión de proyectos software.

Características

- Adaptación
- Inspección
- Transparencia
- Colaboración
- Organización
- Priorización

Ventajas y Desventajas de Scrum

Tabla 5: Ventajas y Desventajas de Scrum

VENTAJAS	DESVENTAJAS
Se pueden obtener resultados anticipados.	Equipos de pocas personas.

Excelente nivel de planificación.	Exhaustiva definición de tareas y sus plazos.
Alto nivel de participación de los involucrados.	No define todos los requisitos al iniciarse el proyecto.
Adaptabilidad.	
Gestión sistemática de riesgos.	El equipo de trabajo tiene que estar empapado de la metodología y responsabilidades de su rol asignado.

Elaborado por: El investigador

Congreso Internacional CSEI

“Es una conferencia internacional especialmente dedicada a la informática, la electrónica e Ingeniería industrial y sus soluciones” [47]. Es un evento desarrollado en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial que tiene como objetivo proporcionar un foro en el cual investigadores, desarrolladores y profesionales puedan revisar y discutir las tendencias más recientes en el área y compartir direcciones de investigación innovadoras.

1.3. Objetivos

1.3.1 Objetivo general

Desarrollar un sitio web para la administración de congresos internacionales en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.3.2 Objetivos específicos

- Identificar los requerimientos funcionales para el desarrollo del sitio web.
- Analizar la metodología de trabajo que se adapte de mejor manera a las condiciones del proyecto.
- Seleccionar la tecnología open source para el desarrollo del sitio web.
- Implantar el sitio web para la administración de congresos internacionales en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

CAPITULO II

2 METODOLOGIA

2.1. Materiales

Para la recolección de información en el presente proyecto se aplicó una entrevista al comité organizador CSEI, así como a un representante de la EP UTA con el objetivo de determinar los requerimientos funcionales del sitio web.

Así como también aplicó una encuesta mediante un cuestionario a usuarios de tipo AUTHOR registrados en el congreso internacional CSEI 2021 con la finalidad de conocer las necesidades de estos.

2.1.1 ENTREVISTA

DETERMINACION DE REQUERIMIENTOS

ENTREVISTADOS: Miembros del comité organizador CSEI y secretaria EP UTA.

FECHA: 4 de enero del 2022

APLICACIÓN: Sitio Web para la Administración de Congresos Internacionales en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

OBJETIVO: Esta entrevista se realiza con el fin de conocer las necesidades que tienen los miembros del comité organizador CSEI y de la representante de la EP UTA. con respecto al sitio web para la administración de congresos internacionales CSEI.

Tabla 6: Entrevista

N.	PREGUNTA
1	¿Cuáles son los problemas del sitio web?
2	¿Con que tecnología de desarrollo el sitio web?
3	¿Qué información se difunde en el sitio web?
4	¿Qué información tiene que ser dinámica?
5	¿Qué funcionalidades requiere que integre el sitio web?
6	¿Cuáles son los tipos de usuarios del sitio web?
7	¿Cuáles son las actividades de los tipos de usuarios?
8	¿Qué funcionalidades requieren los tipos de usuarios?

9	¿Cuáles son sus expectativas con respecto a la funcionalidad del sitio web?
----------	---

Elaborado por: El investigador

2.1.2 ENCUESTA

- 1.Cuál es su nivel de satisfacción con respecto a la información que presenta el sitio web del congreso internacional CSEI (<https://csei.uta.edu.ec/csei2021/>):**
 - Muy Satisfactorio
 - Satisfactorio
 - Poco Satisfactorio
 - Nada Satisfactorio
- 2. ¿La información presentada en el sitio web está organizada y es entendible?**
 - SI
 - NO
- 3. Que información le gustaría consultar en el sitio web:**
 - Guía de registro
 - Biografía de conferencistas
 - Programación del evento
- 4. Como describiría el diseño del sitio web:**
 - Moderno
 - Organizado
 - Desorganizado
 - Anticuado
- 5.Cuál es su nivel de satisfacción con respecto al formulario de registro del congreso internacional CSEI:**
 - Muy Satisfactorio
 - Satisfactorio
 - Poco Satisfactorio
 - Nada Satisfactorio

6. Que funcionalidades le gustaría que integre el módulo de registro del sitio web:

- Sección Informativa
- Editor de Datos Personales
- Ingreso con contraseña
- Generador de Certificados
- Facturación Electrónica

2.2. Métodos

2.2.1 Modalidad de Investigación

Investigación de Campo

Permitió analizar la información recabada en el lugar donde se produjo el problema, teniendo contacto directo con los actores involucrados.

Investigación Bibliográfica

La investigación está basada en artículos y revistas científicas, sitios web especializados, libros e investigaciones relacionadas de manera que se pudo recopilar información sobre la metodología, tecnologías y desarrollo web.

2.2.2 Población y Muestra

La población para el desarrollo de este proyecto está conformada por cuatro miembros del comité organizador CSEI y por cincuenta usuarios de tipo AUTHOR extraídos de los registros de la edición 2021 del CSEI. Teniendo en cuenta que la población para investigar no supera las cien personas, se trabajará con la totalidad del universo.

Tabla 7: Población

N.	Tipo	Descripción	Cantidad	Porcentaje (%)
1	Coordinación UODIDE FISEI	Coordinador Unidad Operativa de Investigación UODIDE FISEI	1	1.85%
2	Comisión Congreso CSEI	Docentes miembros de la comisión del congreso CSEI	2	3.7%
3	Empresa Pública Uta	Secretaria del departamento financiero de la Empresa Publica Uta.	1	1.85%
4	Usuarios	Usuarios de tipo AUTHOR registrados CSEI 2021	50	92.60%
TOTAL			54	100%

Elaborado por: El investigador

2.2.3 Recolección de información

Una vez aplicada la entrevista a: Coordinador Unidad Operativa de Investigación UODIDE FISEI, Docentes miembros de la comisión del congreso CSEI y secretaria del departamento financiero de la Empresa Publica Uta, se obtuvo los siguientes resultados.

Tabla 8: Recolección información - Entrevista

PREGUNTAS	RESPUESTAS
¿Cuáles son los problemas del sitio web?	<ul style="list-style-type: none"> No se dispone de un panel de administración con el cual se pueda configurar la información y opciones presentadas en el sitio web, la información del sitio web se modifica a través del código fuente de este.

	<ul style="list-style-type: none"> • Existen procesos que se los realiza de forma manual, tal es el caso de la edición de un usuario registrado, acción la cual se realiza de forma frecuente. • Se tiene que acceder a la base de datos para actualizar la información. • Usuarios no disponen de una opción para enviar detalle de pago. • Usuarios no disponen de la opción de editar su información de registro o pago. • No se dispone de un generador de reportes. • No existe la opción de subir la factura de los registrados. • El diseño del sitio web no está a la altura de un evento internacional.
¿Con que tecnología de desarrollo el sitio web?	Los lenguajes empleados para el desarrollo del sitio web fueron HTML 5, CSS 3y PHP 7. El sitio web informativo se desarrolló con una plantilla mientras que el módulo de registro con la tecnología Bootstrap.
¿Qué información se difunde en el sitio web?	En el sitio web se difunde la siguiente información: <ul style="list-style-type: none"> • Presentación del evento • Detalles del evento (modalidad, horas de capacitación, fechas, información de contacto). • Información Turística de la ciudad sede del evento. • Programación del evento. • Costos de inscripción. • Comité organizador y científico del evento. • Galería de ediciones. • Patrocinadores del evento.
¿Qué información tiene que ser dinámica?	Toda la información del evento tiene que ser dinámica ya que esta cambia con cada edición.

<p>¿Qué funcionalidades requiere que integre el sitio web?</p>	<p>Se requiere de la integración de:</p> <ul style="list-style-type: none"> • Panel de Administrador que permita, la configuración de del sitio web. • Generador de reportes. • Sección informativa en el módulo de registro. • Sección de validación con campos para detallar la información de pago de un usuario registrado. • Sección de certificados que permita la visualización y descarga de certificados. • Sección de facturación en el módulo de registro que permita la visualización y descarga de facturas. • Funcionalidad para cargar facturas individuales y por lotes. • Editor de información de registrados. 	
<p>¿Cuáles son los tipos de usuarios del sitio web?</p>	<ul style="list-style-type: none"> • Administrador • UsuariosCO (Comité Organizador) • UsuarioEP (Empresa Pública) • UsuariosP (Participantes) 	
<p>¿Cuáles son las actividades de los tipos de usuarios?</p>	<p>Administrador</p> <ul style="list-style-type: none"> • Generación de reportes y certificados. • Configuración de la información del sitio web. • Visualización y edición de registrados. 	<p>UsuariosCO</p> <ul style="list-style-type: none"> • Generación de reportes y certificados. • Visualización de registrados.
	<p>UsuarioEP</p> <ul style="list-style-type: none"> • Validación de registrados. • Carga de facturas. 	<p>UsuariosP</p> <ul style="list-style-type: none"> • Consumo de información del sitio web.

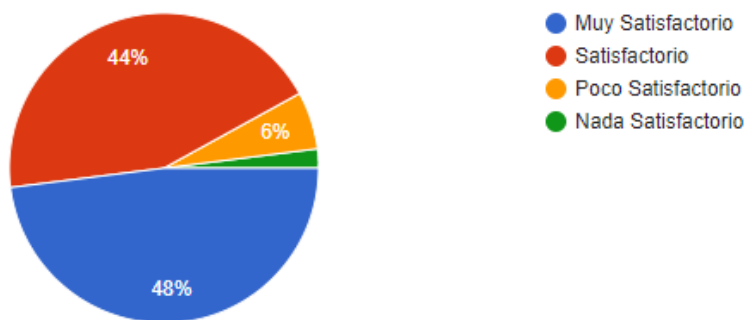
		<ul style="list-style-type: none"> • Registrar, modificar y validar su información de participación. • Visualización y descarga de certificados y facturas.
¿Qué funcionalidades requieren los tipos de usuarios?	Administrador <ul style="list-style-type: none"> • Generador de reportes y certificados. • Módulo de configuración de la información del sitio web. • Editor de registrados. • Validador de registrados. • Carga de facturas. • Envío de correos registrados. 	UsuariosCO <ul style="list-style-type: none"> • Generador de reportes y certificados. • Generador de certificados.
	UsuarioEP <ul style="list-style-type: none"> • Validador de registrados. • Carga de facturas. • Envío de correos registrados. 	UsuariosP <ul style="list-style-type: none"> • Módulos de Información, registro, validación, certificados, facturación y contacto.
¿Cuáles son sus expectativas con respecto a la funcionalidad del sitio web?	Que mejore la presentación del evento internacional, comparta de forma organizada su información, sea fácilmente configurable y que incluya todas las funcionalidades necesarias para enriquecer la experiencia de todos los usuarios al acceder al sitio web.	

Elaborado por: El investigador.

Al aplicarse las 6 preguntas de la encuesta a usuarios de tipo AUTHOR extraídos de los registros del congreso internacional CSEI 2021, se obtuvieron los siguientes resultados.

Pregunta 1: ¿Cuál es su nivel de satisfacción con respecto a la información que presenta el sitio web del congreso internacional CSEI (<https://csei.uta.edu.ec/csei2021/>) ?:

Ilustración 6: Resultado de Encuestas - Pregunta 1



Elaborado por: El investigador.

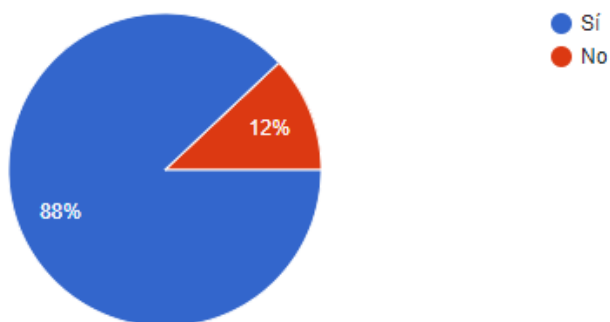
Análisis e interpretación

Casi por unanimidad los encuestados manifestaron estar muy satisfecho o satisfechos con la información presentada por el sitio web de CSEI.

De esta manera, es factible que la información que presenta la página se mantenga.

Pregunta 2: ¿La información presentada en el sitio web está organizada y es entendible?

Ilustración 7: Resultado de Encuestas - Pregunta 2



Elaborado por: El investigador.

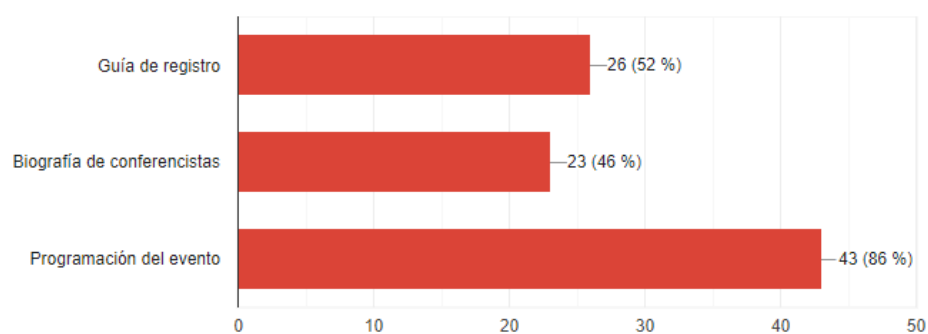
Análisis e interpretación

Al preguntar por la organización y la facilidad de entendimiento de la información presentada la gran mayoría de los encuestados respondió que si es entendible.

Por lo tanto, la distribución de la información se mantendrá igual, el desarrollador podrá realizar el diseño basado en la distribución del CSEI 2021.

Pregunta 3: ¿Qué información le gustaría consultar en el sitio web?

Ilustración 8: Resultado de Encuestas - Pregunta 3



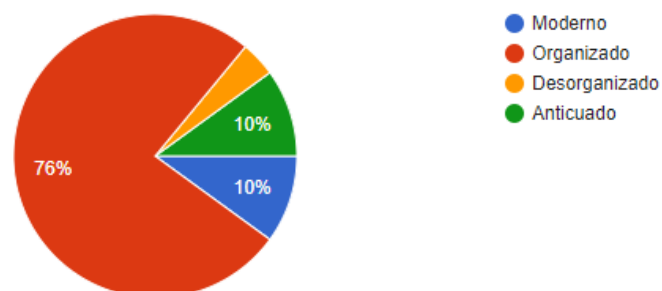
Elaborado por: El investigador.

Análisis e interpretación

Al preguntar qué información debería tener el sitio web para consulta las respuestas más frecuentes fueron programación del evento seguido de guía del registro, y la biografía de los conferencistas, es así como se concluye que es factible presentar esta información para los usuarios que accedan al sitio web.

Pregunta 4: ¿Cómo describiría el diseño del sitio web?:

Ilustración 9: Resultado de Encuestas - Pregunta 4



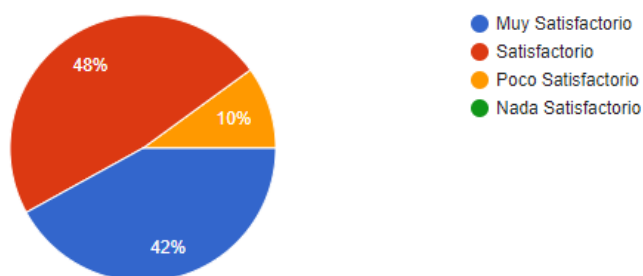
Elaborado por: El investigador.

Análisis e interpretación

Al preguntar sobre el diseño de la página web la mayoría de encuestados lo percibió como moderno y organizado, solo una pequeña parte de la población manifestó en cambio no estar conforme con el diseño de la página en sí. Conforme con el resultado obtenido los usuarios consideran que el diseño no debería sufrir cambios muy agresivos.

Pregunta 5: ¿Cuál es su nivel de satisfacción con respecto al formulario de registro del congreso internacional CSEI?:

Ilustración 10: Resultado de Encuestas - Pregunta 5



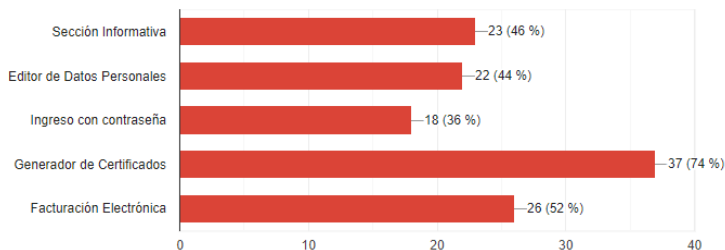
Elaborado por: El investigador.

Análisis e interpretación

Al preguntar sobre el formulario de registro la gran mayoría de los encuestados manifestaron estar satisfechos en gran manera con el formulario de registro. De acuerdo con el resultado obtenido el formulario de registro mantendría sus funciones base.

Pregunta 6: ¿Que funcionalidades le gustaría que integre el módulo de registro del sitio web?:

Ilustración 11: Resultado de Encuestas - Pregunta 6



Elaborado por: El investigador.

Análisis e interpretación

Al preguntar cuáles son las funcionalidades que les gustaría que se integren en el módulo de registro del sitio web, la más solicitada fue la de generación de certificado, seguida de la facturación electrónica y la sección informativa, aunque las opciones como edición de datos personales en ingreso fueron también muy votadas.

De acuerdo con los resultados obtenidos, es factible dotar de nuevas funcionalidades al módulo de registro del sitio web.

2.2.4 Procesamiento y análisis de datos

Mediante esta entrevista se determinó los requerimientos para el desarrollo del sitio web, así también se conoció la forma en la que llevan a cabo las actividades durante el desarrollo del congreso CSEI, concluyendo que:

- Se requiere de un módulo de configuración en el cual se pueda modificar toda la información presentada en el sitio web.
- Es importante que se disponga de una función para la carga de facturas de los usuarios registrados.
- Existe la necesidad de implementar un módulo de generación de certificados y reportes el cual funcione dentro del mismo sitio web.
- Se requiere de la función de enviar correos individuales y por lotes a los usuarios registrados.

De acuerdo con la encuesta realizada a los usuarios de tipo AUTHOR registrados en el congreso CSEI 2021, se pudo concluir que los usuarios están satisfechos con el diseño y la información presentada, así como con el módulo de registro, sin embargo, los encuestados manifestaron que se requiere de información y funciones adicionales tales como:

- Información sobre la programación del evento.
- Presentación de la biografía de los conferencistas.
- Módulo para la consulta de las facturas electrónicas.
- Módulo de registro que permita editar la información del usuario.

CAPÍTULO III

3 RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de resultados

3.1.1 Comparativa de metodologías ágiles

Tabla 9: Comparativa de metodologías ágiles.

Scrum	XP
Es un método rápido para desarrollo de proyectos que en algunos casos no permite cumplir con los requerimientos por las limitaciones que se generan para lograr la agilidad en el desarrollo.	En una metodología que se enfoca en las sentencias de código para lograr mayor funcionabilidad del proyecto sin que se afecte la rapidez del desarrollo, al igual que la dinámica de trabajo.
Los administradores y programadores tienden a trabajar en solitario y de a uno.	Esta metodología promueve el trabajo en equipos de dos personas en todos sus módulos como tendencia pudiendo trabajar también de manera individual sin problema.
Van de una a cuatro semanas los ciclos de trabajo.	Los ciclos de trabajo son más eficientes y se extienden a tres semanas como máximo
Una vez que se concluye con alguna tarea y cumple su función ya es muy difícil mejorar o cambiar su estructura.	Se puede realizar cambios con facilidad en cualquier tiempo durante el desarrollo del proyecto según vayan apareciendo nuevas necesidades.
Necesitan seguir un orden establecido a manera de algoritmo inamovible para seguir avanzando, pero en el caso de los Sprint se puede cambiar el orden original.	Es más organizado se debe seguir estrictamente el orden de planificación del administrador siempre.

Este hecho pensando en grandes corporaciones.	Hecho para empresas pequeñas, así como grandes instituciones o empresas.
---	--






Elaborado por: El investigador

3.1.2 Metodología seleccionada

Una vez que se ha podido comparar y analizar las metodologías de desarrollo ágil, tomando en cuenta el uso de la tecnología para el desarrollo del sitio web, se ha seleccionado XP, ya que brinda mayores beneficios que las alternativas a esta, además de que se puede adaptar con mayor facilidad a los cambios en el proyecto lo que permite tener actualizaciones continuas sin necesidad de tener que actualizar todo el proyecto de manera periódica y evitar perder recursos y tiempo en el desarrollo, permite tener equipos de trabajo reducidos, su comunicación continua y duradera con los miembros del proyecto minimiza la tasa de errores y facilita un rápido desarrollo del proyecto.

3.1.3 Comparativa de frameworks de desarrollo PWA

Tabla 30: Comparativa de frameworks de desarrollo PWA.

Características	 React	 Vue	 Angular	 Ionic	 Flutter
Modelo de Objetos del Documento (DOM)	Virtual DOM	Virtual DOM	Regular DOM	Virtual DOM	Virtual DOM
Línea de aprendizaje	Norma	Rápida	Rápida	Rápida	Rápida

Paquete de componentes	Fuerte	Fuerte	Fuerte	Fuerte	Fuerte
Abstracción	Medio	Medio	Fuerte	Fuerte	Fuerte
Depuración General	Buen HTML/Buen Js	Buen HTML/Medio Js	Buen HTML/Medio Js	Buen HTML/Medio Js	Buen HTML/Medio Js -
¿Etiqueta no cerrada mencionada?	Si	Si	Si	Si	Si
Enlace de datos	Una dirección, sin cambio en los datos	Ambas direcciones, cambio en los datos	Ambas direcciones, cambio en los datos	Ambas direcciones, cambio en los datos	Ambas direcciones, cambio en los datos
Plantillas	Si	Si	Si	Si	Si
Modelo de componente	Medio	Medio	Medio	Medio	Medio
¿Compilación Móvil?	React Native	Ionic Framework	Ionic Framework	Ionic Framework	Ionic Framework
MVC	vista de modelo	Dos vistas	MVC	vista de modelo	MVC
Renderizado	Lado del cliente	Lado del servidor	Lado del servidor	Lado del servidor	Lado del servidor
Router	React Router v4	vue-router	@angular/router	@angular/router	@angular/router

Lenguaje	JS ES6+	JS, TypeScript	JS, TypeScript	JS, TypeScript	JS, TypeScript
Componentes UI	Material-UI Library	Vuetify	Material Desing	Extensa librería de controles visuales, comportamie ntos, estilos, animaciones.	Librería propia
Inyección de dependencias	No soportado	Soportado	Soportado	Soportado	Soportado
Comunidad	Extensa	Extensa	Extensa	Extensa	Extensa
Compañías	Facebook , Twitter, Netflix, PayPal	Nintendo, Gitlab, Netflix	HBO, Apple, Adobe, Nike, Microsoft	Burger King, Huawei, Microsoft, General Electric	BMW, eBay

Elaborado por: El investigador

3.1.4 Framework de desarrollo seleccionado

Una vez que se ha analizado los tipos de frameworks que se pueden utilizar de ha decidido seleccionar Angular, y aunque han aparecido modelos recientes como Flutter con una interfaz más estética la fiabilidad y seguridad de Angular hace que sea por mucho la mejor opción, además que con este framework podemos contar con una gran cantidad de documentación técnica y una comunidad de desarrolladores conde resolver dudas ante posibles errores.

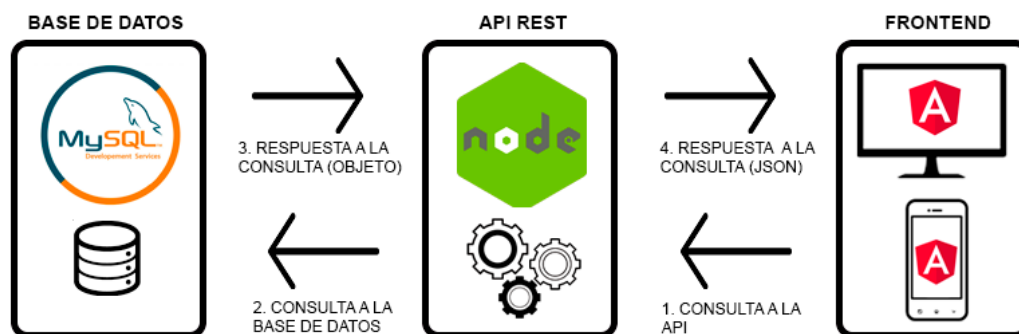
3.1.5 Determinación de la tecnología Back-End

La tecnología para el Back-End será Node.js por la facilidad de uso y debido a que es muy robusta, además de ser muy estable y tener una amplia documentación en la cual apoyarse durante el desarrollo del proyecto.

3.1.6 Arquitectura de la aplicación

La arquitectura seleccionada para el proyecto es cliente-servidor, los servicios Node.js se alojarán en un servidor y serán consumidos por los usuarios a través de la PWA en Angular.

Ilustración 12: Arquitectura Cliente-Servidor



Autor: Omar Fonseca

Elaborado por: El investigador

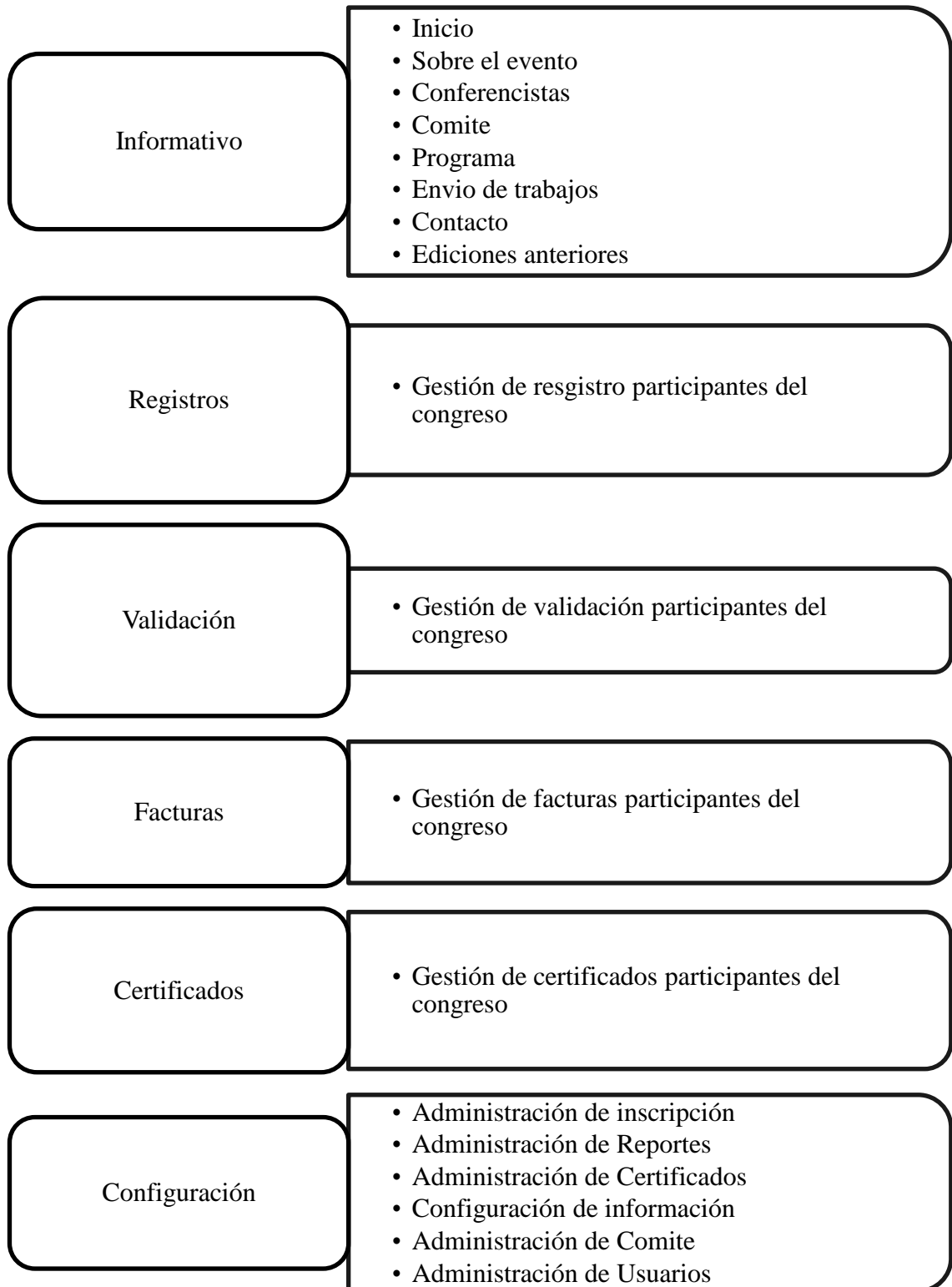
3.2. Desarrollo de la propuesta

3.2.1 Fase I: Planificación

Los organizadores del congreso internacional CSEI y la representante de la EP UTA que fueron entrevistados, con la finalidad de mejorar el sitio web para la administración de congresos internacionales, brindaron las facilidades para el levantamiento de requerimientos. Así como los usuarios de tipo AUTHOR encuestados ayudaron a que se tenga una idea más clara de las necesidades del sitio web.

En base a la entrevista y encuestas realizadas se establecieron los siguientes módulos que contendrá el sitio web.

Ilustración 13: Módulos establecidos



Elaborado por: El investigador

Definición de roles

Definir los roles, permite establecer las responsabilidades de cada uno de los involucrados.

Tabla 41: Definición de los roles

ROL XP	RESPONSABLE	RESPONSABILIDAD
Usuarios	Comité organizador CSEI, estudiantes, profesionales y autores interesado en el congreso internacional.	Definir los requerimientos funcionales y especificaciones del sitio web.
Programador	Omar Fonseca	Producir el sitio web. Diseñar, programar y probar el sitio web.
Entrenador (coach)	Ing. Mg. Carlos Israel Nuñez Miranda	Guiar al equipo, y realizar el seguimiento de los procesos verificando su cumplimiento.
Tester	Omar Fonseca Ing. Mg. Carlos Israel Nuñez Miranda	Realizar las pruebas de aceptación.

Elaborado por: El investigador

Descripción de los procesos

En esta fase del proyecto se procederá a modelar los procesos que son parte del desarrollo del CSEI. En la siguiente tabla se describen los procesos que conllevan la realización del congreso internacional desarrollado por la facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Tabla 5: Cuadro de actividades y procesos del desarrollo del CSEI.

ACTIVIDAD	PROCESOS
Convocatoria.	1. Mediante la notificación de aprobación y presentación del manuscrito final del congreso CSEI. Los organizadores del evento realizan la convocatoria para participar en el congreso a

	través de la página web, modificando de forma manual la información de esta.
Publicación del evento.	<ol style="list-style-type: none"> 1. Los miembros del comité organizador seleccionan la información que será presentará en la página web. 2. Modifican la página web de forma manual, editando todas las pestañas informativas.
Inicio Congreso	<ol style="list-style-type: none"> 1. Los organizadores del evento crean una nueva base de datos para la presente edición del congreso y la enlazan al sistema de registro. 2. Se activa el sistema de registro.
Inscripciones	<ol style="list-style-type: none"> 1. Los usuarios acceden a la página informativa para ser redirigidos hacia el sistema de registro que funciona de forma independiente.
Registro	<ol style="list-style-type: none"> 1. Los usuarios ingresan al sistema de registro. 2. Ingreso de cedula para verificar si existe o no en los registros de la base de datos, si existe los datos se auto completan, caso contrario al usuario se le permite rellenar todos los campos para la inscripción. 3. Los usuarios realizan su registro llenando los datos solicitados acorde a su tipo de participante. Aceptan su registro y finalizan con este para posteriormente esperar por un correo electrónico con la información de pago.
Validación	<ol style="list-style-type: none"> 1. Los usuarios acceden al apartado de validación del sistema de registro. 2. Ingresan su documento de identificación y realizan la validación de su registro subiendo la imagen y numero del comprobante de pago. 3. Los usuarios reciben un correo electrónico en el cual se indica que su solicitud será aprobada.
Facturación	<ol style="list-style-type: none"> 1. Un representante de la EP UTA ingresa a la parte administrativa del sistema de registro y procede a validar a los usuarios registrados que ya registren su comprobante de pago cargado.

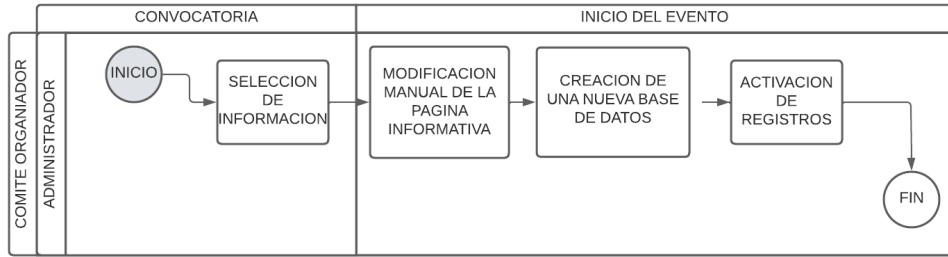
	<ol style="list-style-type: none"> 2. Revisan uno a uno los comprobantes de pago y en caso de un error por parte del participante, se copia de forma manual su correo electrónico para comunicarse e indicar del error en cuestión a través de aplicaciones de correo electrónico. En caso de que la información este correcta el representante de EP UTA valida al usuario. 3. Una vez que se valide el pago de un participante se factura mediante un sistema propio de la EP UTA y se envía la factura por correo electrónico mediante aplicaciones de correo electrónico.
Cierre del congreso	<ol style="list-style-type: none"> 1. Los organizadores del congreso deshabilitan la opción de registro.
Emisión de certificados	<ol style="list-style-type: none"> 1. El administrador enlaza la base de datos del congreso a un sistema local para la generación de certificados, esta acción se la realiza fuera del sistema de registro. 2. El administrador carga de forma manual los certificados al servidor. 3. Los certificados para conferencistas, miembros del comité organizador y científico se realizan de forma manual mediante aplicaciones de edición grafica.
Generación de reportes	<ol style="list-style-type: none"> 1. El comité organizador realiza consultas a la base de datos para conocer el número de registrado total y por categoría.

Elaborado por: El investigador

Modelamiento de los procesos

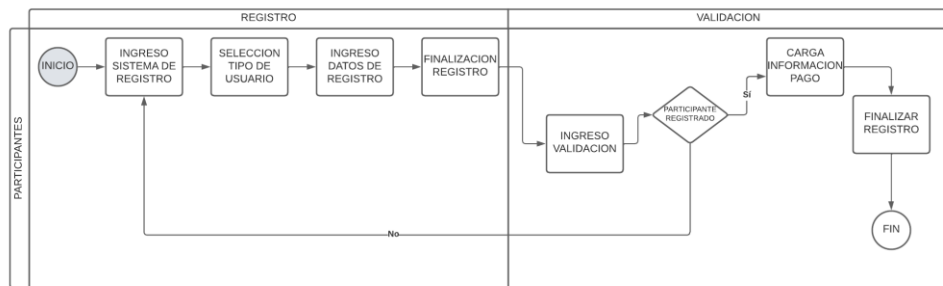
En la figura 14, 15, 16 y 17 se detallan gráficamente los procesos con los que se ha desarrollado el evento CEI.

Ilustración 14: Inicio CSEI



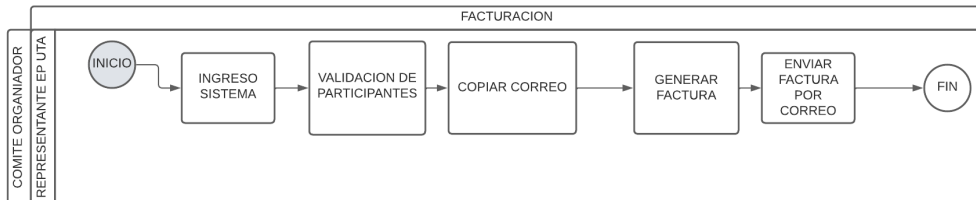
Elaborado por: El investigador

Ilustración 15: Registro y Validación



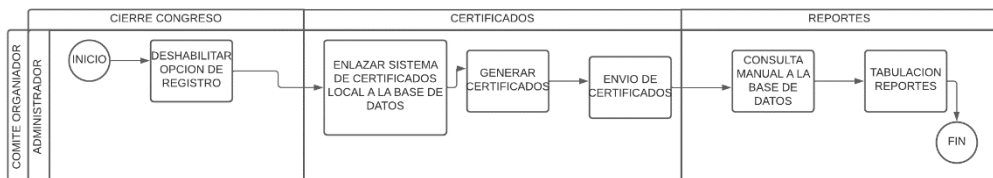
Elaborado por: El investigador

Ilustración 16: facturación



Elaborado por: El investigador

Ilustración 17: Fin CSEI



Elaborado por: El investigador

Historias De Usuario

Ejecutar historias de usuario es fundamental para mantener un control adecuado sobre cómo se desarrolla el proyecto, porque definen la funcionalidad que se implementará en el proyecto desde el punto de vista del usuario final. Una vez que se completan las historias de usuario, las actividades se desglosan para estimar cuánto tiempo llevará completarlas.

Tabla 62. Plantilla de historias de usuario

HISTORIA DE USUARIO	
Número:	Nombre:
Usuario:	Iteraciones:
Prioridad en Negocio:	Riesgo en Desarrollo:
Puntos estimados:	Puntos reales:
Responsable:	
Descripción:	
Observación:	

Elaborado por: El investigador

Descripción de las historias de usuarios.

Tabla 13. Descripción de las historias de usuarios

HISTORIA DE USUARIO	
Número: Código de identificación de la historia	Nombre: Título que se le da a la historia.
Usuario: Persona/s a la/s que va dirigida la historia.	Iteraciones: Ciclo completo de análisis, diseño, desarrollo y pruebas
Prioridad en Negocio: Rango que describe a importancia de la historia.	Riesgo en Desarrollo: Rango que describe el nivel según su evaluación.
Puntos estimados: Número de días estimados para el desarrollo de la historia	Puntos reales: Número de días empleados para el desarrollo de la historia.
Responsable: Programador responsable del desarrollo de la historia.	

Descripción: Información proporcionada por el cliente para el desarrollo de la historia.
Observación: Campo opcional que se utiliza para aclarar la información.

Elaborado por: El investigador

Historias de usuario

Tabla 7. Historia de usuario - Estructurar el proyecto

HISTORIA DE USUARIO	
Número: H001	Nombre: Estructurar el proyecto
Usuario: Usuario Final	Iteraciones: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Puntos estimados: 4	Puntos reales: 4
Responsable: Omar Fonseca	
Descripción: Establecer la estructura del proyecto en base al framework a utilizar.	
Observación: El framework seleccionado es Angular.	

Elaborado por: El investigador

Tabla 8. Historia de usuario - Diseño de la base de datos

HISTORIA DE USUARIO	
Número: H002	Nombre: Diseño de la base de datos
Usuario: Usuario Final	Iteraciones: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 5	Puntos reales: 5
Responsable: Omar Fonseca	
Descripción: Diseñar la base de datos.	
Observación: La base de datos se diseñará en MySQL.	

Elaborado por: El investigador

Tabla 9. Historia de usuario - Diseño de interfaz modulo informativo

HISTORIA DE USUARIO	
Número: H003	Nombre: Diseño de interfaz modulo informativo
Usuario: Usuario Final	Iteraciones: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 10	Puntos reales: 10
Responsable: Omar Fonseca	
Descripción: Diseño de interfaces que tendrá el módulo informativo para que el usuario interactúe.	
Observación: La interfaz estará compuesta por diferentes submódulos.	

Elaborado por: El investigador

Tabla 10. Historia de usuario - Diseño de interfaz módulo de registro

HISTORIA DE USUARIO	
Número: H004	Nombre: Diseño de interfaz módulo de registro
Usuario: Usuario Final	Iteraciones: 2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 5	Puntos reales: 5
Responsable: Omar Fonseca	
Descripción: Diseño de la interfaz que tendrá el módulo de registro para que el usuario interactúe. La interfaz de registro contará con 3 diferentes formularios, ingreso de documento de identidad, ingreso de datos personales y selección de método de pago.	
Observación:	

Elaborado por: El investigador

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

HISTORIA DE USUARIO	
Número: H005	Nombre: Diseño de interfaz módulo de validación
Usuario: Usuario Final	Iteraciones: 2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 4	Puntos reales: 4
Responsable: Omar Fonseca	
Descripción: Diseño de la interfaz que tendrá el módulo de validación para que el usuario interactúe, la interfaz de validación contara con 2 formularios, ingreso de documento de identidad y carga de comprobante de pago.	
Observación:	

Elaborado por: El investigador

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

HISTORIA DE USUARIO	
Número: H006	Nombre: Diseño de interfaz módulo de facturas
Usuario: Usuario Final	Iteraciones: 2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Puntos estimados: 4	Puntos reales: 4
Responsable: Omar Fonseca	
Descripción: Diseño de la interfaz que tendrá el módulo de facturas para que el usuario interactúe, esta interfaz contara con un formulario, ingreso de documento de identidad y una pantalla en la cual se podrá visualizar la factura del usuario.	
Observación:	

Elaborado por: El investigador

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

HISTORIA DE USUARIO	
Número: H007	Nombre: Diseño de interfaz módulo de certificados
Usuario: Usuario final	Iteraciones: 2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 3	Puntos reales: 3
Responsable: Omar Fonseca	
Descripción: Diseño de la interfaz que tendrá el módulo de certificados para que el usuario interactúe, esta interfaz contara con un formulario, ingreso de documento de identidad y una pantalla en la cual se podrá visualizar los certificados del usuario.	
Observación:	

Elaborado por: El investigador

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

HISTORIA DE USUARIO	
Número: H008	Nombre: Diseño de interfaz módulo de configuración
Usuario: Usuario Final	Iteraciones: 3
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 15	Puntos reales: 15
Responsable: Omar Fonseca	
Descripción: Diseño de la interfaz que tendrá el módulo de configuración para que los usuarios administrados interactúen, esta interfaz está compuesta por varios submódulos los cuales permitirán gestionar la información de la página, así como gestionar la información de los involucrados en el congreso CSEI.	
Observación:	

Elaborado por: El investigador

Tabla 15. Historia de usuario - Inicio de sesión

HISTORIA DE USUARIO	
Número: H009	Nombre: Inicio de sesión
Usuario: Usuario final	Iteraciones: 3
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 3	Puntos reales: 3
Responsable: Omar Fonseca	
Descripción: Los usuarios encargados de la administración de la información del sitio web y de los involucrados en el congreso CSEI, accederán con sus credenciales a una pantalla de inicio de sesión en la que serán validados para acceder al módulo de configuración.	
Observación: Se consultará la tabla de usuarios y de acuerdo con el tipo de usuario se habilitarán las opciones de configuración del módulo.	

Elaborado por: El investigador

Tabla 16. Historia de usuario - Desarrollo módulo informativo.

HISTORIA DE USUARIO	
Número: H010	Nombre: Desarrollo módulo informativo.
Usuario: Usuario final	Iteraciones: 4
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 15	Puntos reales: 15
Responsable: Omar Fonseca	
Descripción: Desarrollo del módulo informativo en el que se presentará información dinámica en los submódulos de inicio, sobre el evento, conferencistas, comité, programa, envío de trabajos, contacto y ediciones anteriores.	
Observación: Se consultará las tablas de botones_inicio, comité, conferencista, ediciones, enviotrabajos, enviotrabajos_fechasimportantes, enviotrabajos_formatos, galeria_informacion_turistica, galeria_inicio, galeria_lugar_del_evento, imagenes_portada, informacion_congreso, informacion_turistica, inicio, lugar_del_evento, patrocinadores, presentación,	

programa, programa_detalle, programa_dias, programa_jordana, registro_informacion, temarios, temarios_temas, tipo_comite, para extraer la información que será presentada en el módulo informativo.

Elaborado por: El investigador

Tabla 17. Historia de usuario - Desarrollo módulo de registro

HISTORIA DE USUARIO	
Número: H011	Nombre: Desarrollo módulo de registro
Usuario: Usuario final	Iteraciones: 4
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Puntos estimados: 7	Puntos reales: 7
Responsable: Omar Fonseca	
Descripción: Desarrollo del módulo de registro, en el que el usuario deberá completar un formulario con sus datos personales para realizar su correcta inscripción en el congreso.	
Observación: Se consultará la tabla personas para cargar los datos del usuario en caso de que exista, de igual manera se consultara a las tablas carreras, idiomas, paralelo, semestres, tipodocumento, tipoinscripcion, tipopago, tipopaper. Al realizar el registro se insertará los datos en las tablas de persona, inscripciones, paper, autores de acuerdo con el tipo de usuario que se registre.	

Elaborado por: El investigador

Tabla 18. Historia de usuario - Desarrollo módulo de validación

HISTORIA DE USUARIO	
Número: H012	Nombre: Desarrollo módulo de validación
Usuario: Usuario final	Iteraciones: 5
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 3	Puntos reales: 3
Responsable: Omar Fonseca	

Descripción: Desarrollo del módulo de validación: En este módulo el usuario tendrá la opción de subir su comprobante de pago en formato jpg o png para su posterior validación por la administración.
Observación: Se consultará la tabla de inscripciones para cargar los datos del usuario en caso de que exista, al realizar la validación se actualizarán los datos en la tabla de inscripciones.

Elaborado por: El investigador

Tabla 19. Historia de usuario - Desarrollo módulo de facturas

HISTORIA DE USUARIO	
Número: H013	Nombre: Desarrollo módulo facturas
Usuario: Usuario final	Iteraciones: 5
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 3	Puntos reales: 3
Responsable: Omar Fonseca	
Descripción: Desarrollo de funcionalidad del módulo de facturas: Este módulo mostrará la factura del usuario ingresado, permitirá su visualización y descarga.	
Observación: Se consultará la tabla de inscripciones para cargar la factura del usuario en caso de que esta exista.	

Elaborado por: El investigador

Tabla 20. Historia de usuario - Desarrollo módulo de certificados

HISTORIA DE USUARIO	
Número: H014	Nombre: Desarrollo módulo certificados
Usuario: Usuario final	Iteraciones: 5
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 3	Puntos reales: 3
Responsable: Omar Fonseca	

Descripción: Desarrollo de funcionalidad del módulo de certificados: Este módulo mostrará los certificados del usuario ingresado, permitirá su visualización y descarga.
Observación: Se consultará las tablas de inscripciones, papers, autores de acuerdo con el tipo de usuario, para cargar los certificados en caso de que existan.

Elaborado por: El investigador

Tabla 21. Historia de usuario - Desarrollo módulo de configuración

HISTORIA DE USUARIO	
Número: H015	Nombre: Desarrollo módulo de configuración
Usuario: Usuario final	Iteraciones: 6 y 7
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Puntos estimados: 35	Puntos reales: 40
Responsable: Omar Fonseca	
Descripción: Desarrollo de funcionalidad del módulo de configuración: A este módulo accederán usuarios validados para gestionar la información de la página informativa y de los involucrados en el congreso CSEI.	
Observación: Se realizarán operaciones de consulta, inserción, actualización y eliminación en las tablas autores, botones_inicio, carreras, comité, conferencista, configuraciones, confi_reportes, ediciones, enviotrabajos, enviotrabajos_fechaimportantes, enviotrabajos_formatos, galeria_informacion_turistica, galeria_inicio, galeria_lugar_del_evento, idiomas, imagenes_portada, informacion_congreso, informacion_turistica, inicio, inscripción, lugar_del_evento, pago_terceros, paper, paralelo, patrocinadores, persona, presentación, programa, programa_detalle, programa_dias, programa_jordana, registro_informacion, semestres, temarios, temarios_temas, tipocertificado, tipodocumento, tipoinscripcion, tipopago, tipopaper, tipo_comite, tipo_usuario_admin y usuario, para el manejo de la información del sitio web y de los involucrados en el congreso CSEI.	

Elaborado por: El investigador

Valoración de las historias de usuario

Después de detallar los requisitos descritos en la historia del usuario, valoraciones definidas para los cuales se consideró dedicar 6 horas de trabajo por día laborable.

Estimación de historias de usuario

Tabla 22. Estimación de historias de usuario

No	Historia de usuario	Tiempo Estimado	
		Días	Horas
H001	Estructurar el proyecto	4	24
H002	Diseño de la base de datos	5	30
H003	Diseño de interfaz modulo informativo	10	60
H004	Diseño de interfaz módulo de registro	5	30
H005	Diseño de interfaz módulo de validación	4	24
H006	Diseño de interfaz módulo de facturas	4	24
H007	Diseño de interfaz módulo de certificados	3	18
H008	Diseño de interfaz módulo de configuración	15	90
H009	Inicio de sesión	3	18
H010	Desarrollo módulo informativo.	15	90
H011	Desarrollo módulo de registro	7	42
H012	Desarrollo módulo de validación	3	18
H013	Desarrollo módulo de facturas	3	18
H014	Desarrollo módulo de certificados	3	18
H015	Desarrollo módulo de configuración	40	240
Total:		124	744

Elaborado por: El investigador

Tareas

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

Tabla 23. Plantilla de tareas de historias de usuario

TAREA	
Número:	Numero de Historia:
Nombre:	
Tipo de tarea:	Puntos estimados:
Fecha inicio:	Fecha fin:
Programador responsable:	
Descripción:	

Elaborado por: El investigador

Tabla 24. Descripción de las tareas

TAREA	
Número: Código de la tarea para identificarla	Numero de Historia: Código de la historia de usuario asociada a la tarea.
Nombre: Titulo que se le da a la tarea.	
Tipo de tarea: Referencia la clase de la actividad realizada	Puntos: Número de días empleados.
Fecha inicio: Fecha en la que se empezó a realizar la tarea.	Fecha fin: Fecha en la que se finalizó la tarea.
Responsable: Nombre del programador responsable de la realización de la tarea.	
Descripción: Detalla la actividad realizada.	

Elaborado por: El investigador

Tabla 25. Tarea - Estructurar el proyecto

TAREA	
Número: T001	Numero de Historia: H001
Nombre: Estructurar el proyecto	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Fecha inicio: 01/04/2022	Fecha fin: 04/06/2022
Programador responsable: Omar Fonseca	
Descripción: Establecer la estructura del proyecto en base al framework Angular.	

Elaborado por: El investigador

Tabla 26. Tarea - Diseño de la base de datos

TAREA	
Número: T002	Numero de Historia: H002
Nombre: Diseño de la base de datos	
Tipo de tarea: Desarrollo	Puntos estimados: 5
Fecha inicio: 07/04/2022	Fecha fin: 13/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseñar la base de datos en MySQL.	

Elaborado por: El investigador

Tabla 27. Tarea - Diseño inicio

TAREA	
Número: T003	Numero de Historia: H003
Nombre: Diseño página inicio	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 14/04/2022	Fecha fin: 14/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página de inicio del módulo informativo.	

Elaborado por: El investigador

Tabla 28. Tarea - Diseño página sobre el evento

TAREA	
Número: T004	Numero de Historia: H003
Nombre: Diseño página sobre el evento	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 15/04/2022	Fecha fin: 18/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página sobre el evento del módulo informativo.	

Elaborado por: El investigador

Tabla 29. Tarea - Diseño página conferencistas

TAREA	
Número: T005	Numero de Historia: H003
Nombre: Diseño página conferencistas	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 19/04/2022	Fecha fin: 19/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página conferencistas del módulo informativo.	

Elaborado por: El investigador

Tabla 30. Tarea - Diseño página comité

TAREA	
Número: T006	Numero de Historia: H003
Nombre: Diseño página comité	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 20/04/2022	Fecha fin: 20/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página comité del módulo informativo.	

Elaborado por: El investigador

Tabla 31. Tarea - Diseño página programa

TAREA	
Número: T007	Numero de Historia: H003
Nombre: Diseño página programa	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 21/04/2022	Fecha fin: 22/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página programa del módulo informativo.	

Elaborado por: El investigador

Tabla 32. Tarea - Diseño página envió de trabajos

TAREA

Número: T008	Numero de Historia: H003
Nombre: Diseño página envió de trabajos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/04/2022	Fecha fin: 25/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página envió de trabajos del módulo informativo.	

Elaborado por: El investigador

Tabla 33. Diseño página contacto

TAREA	
Número: T009	Numero de Historia: H003
Nombre: Diseño página de contacto	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 26/04/2022	Fecha fin: 26/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página de contacto del módulo informativo.	

Elaborado por: El investigador

Tabla 34. Tarea - Diseño página ediciones anteriores

TAREA	
Número: T010	Numero de Historia: H003
Nombre: Diseño página ediciones anteriores	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 27/04/2022	Fecha fin: 27/04/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de la página de ediciones anteriores del módulo informativo.	

Elaborado por: El investigador

Tabla 35. Tarea - Diseño interfaz módulo de registro

TAREA	
Número: T011	Numero de Historia: H004

Nombre: Diseño interfaz módulo de registro	
Tipo de tarea: Desarrollo	Puntos estimados: 5
Fecha inicio: 28/04/2022	Fecha fin: 04/05/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño interfaz módulo de registro para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 36. Tarea - Diseño interfaz módulo de validación

TAREA	
Número: T012	Numero de Historia: H005
Nombre: Diseño interfaz módulo de validación	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Fecha inicio: 05/05/2022	Fecha fin: 11/05/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño interfaz módulo de validación para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 37. Tarea - Diseño interfaz módulo de facturas

TAREA	
Número: T013	Numero de Historia: H006
Nombre: Diseño interfaz módulo de facturas	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Fecha inicio: 12/05/2022	Fecha fin: 17/05/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño interfaz módulo de facturas para que el usuario pueda interactuar.	

Elaborado por: El investigador

Tabla 38. Tarea - Diseño interfaz módulo de certificados

TAREA	
Número: T014	Numero de Historia: H007
Nombre: Diseño interfaz módulo de certificados	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 18/05/2022	Fecha fin: 20/05/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño interfaz módulo de certificados para que el usuario pueda interactuar. En esta interfaz el usuario inscrito podrá visualizar y descargar sus certificados.	

Elaborado por: El investigador

Tabla 39. Tarea - Diseño interfaz control de inscripciones

TAREA	
Número: T015	Numero de Historia: H008
Nombre: Diseño interfaz control de inscripciones	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 23/05/2022	Fecha fin: 23/05/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de interfaz control de inscripciones, esta interfaz contará con una tabla en la cual los administrativos podrán realizar acciones de actualización sobre los registros de inscritos.	

Elaborado por: El investigador

Tabla 40. Tarea - Diseño interfaz módulo de reportes

TAREA	
Número: T016	Numero de Historia: H008
Nombre: Diseño interfaz módulo de reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/05/2022	Fecha fin: 24/05/2022
Programador responsable: Omar Fonseca	

Descripción: Diseño de interfaz módulo reportes, en esta interfaz se podrá generar reportes de los inscritos.

Elaborado por: El investigador

Tabla 41. Tarea - Diseño de interfaz módulo de generación de certificados

TAREA	
Número: T017	Numero de Historia: H008
Nombre: Diseño de interfaz módulo de generación de certificados	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 25/05/2022	Fecha fin: 25/05/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño interfaz módulo de generación de certificados, en esta interfaz se encontrará una tabla en la cual se podrá generar los certificados para cada uno de los participantes del congreso.	

Elaborado por: El investigador

Tabla 42. Tarea - Diseño de interfaz módulo de configuración

TAREA	
Número: T018	Numero de Historia: H008
Nombre: Diseño de interfaz módulo de configuración	
Tipo de tarea: Desarrollo	Puntos estimados: 10
Fecha inicio: 26/05/2022	Fecha fin: 08/06/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de interfaz de configuración, en esta pantalla contará con diferentes pestañas en las cuales se podrá modificar la información del módulo informativo.	

Elaborado por: El investigador

Tabla 43. Tarea - Diseño de interfaz módulo de comité

TAREA	
Número: T019	Numero de Historia: H008

Nombre: Diseño de interfaz módulo de comité	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 09/06/2022	Fecha fin: 09/06/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo de interfaz módulo de comité, en esta pestaña se podrá insertar, editar, eliminar, validar y generar los certificados para el comité organizador y científico del congreso CSEI.	

Elaborado por: El investigador

Tabla 44. Tarea - Diseño de interfaz módulo de usuarios

TAREA	
Número: T020	Numero de Historia: H008
Nombre: Diseño de interfaz módulo de usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 10/06/2022	Fecha fin: 10/06/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño de interfaz módulo de usuarios, en esta pestaña se podrá gestionar a los usuarios que tienen acceso al sistema.	

Elaborado por: El investigador

Tabla 45. Tarea - Inicio de sesión front-end

TAREA	
Número: T021	Numero de Historia: H009
Nombre: Inicio de sesión front-end	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 13/06/2022	Fecha fin: 13/06/2022
Programador responsable: Omar Fonseca	
Descripción: Diseño del Login en Angular.	

Elaborado por: El investigador

Tabla 46. Tarea - Inicio de sesión back-end

TAREA	
Número: T022	Numero de Historia: H009
Nombre: Inicio de sesión back-end	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 14/06/2022	Fecha fin: 15/06/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo de métodos Rest en node.js para el inicio de sesión.	

Elaborado por: El investigador

Tabla 47. Tarea - Desarrollo página informativa inicio

TAREA	
Número: T023	Numero de Historia: H010
Nombre: Desarrollo página informativa inicio	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 16/06/2022	Fecha fin: 16/06/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa inicio, mostrar la información extraída de la base de datos en la página de inicio.	

Elaborado por: El investigador

Tabla 48. Tarea - Desarrollo página informativa sobre el evento

TAREA	
Número: T024	Numero de Historia: H010
Nombre: Desarrollo página informativa sobre el evento	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 17/06/2022	Fecha fin: 20/06/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa sobre el evento, mostrar la información extraída de la base de datos en la página sobre el evento.	

Elaborado por: El investigador

Tabla 49. Tarea - Desarrollo página informativa comité

TAREA	
Número: T026	Numero de Historia: H010
Nombre: Desarrollo página informativa comité	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 24/06/2022	Fecha fin: 24/06/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa comité, mostrar la información extraída de la base de datos en la página comité.	

Elaborado por: El investigador

Tabla 50. Tarea - Desarrollo de página informativa programa

TAREA	
Número: T026	Numero de Historia: H010
Nombre: Desarrollo de página informativa programa	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Fecha inicio: 27/06/2022	Fecha fin: 30/06/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa programa, mostrar la información extraída de la base de datos en la página programa.	

Elaborado por: El investigador

Tabla 51. Tarea - Desarrollo de página informativa envió de trabajos

TAREA	
Número: T027	Numero de Historia: H010
Nombre: Desarrollo de página informativa envió de trabajos	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha inicio: 01/07/2022	Fecha fin: 01/07/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa envió de trabajos, mostrar la información extraída de la base de datos en la página envió de trabajos.	

Elaborado por: El investigador

Tabla 52. Tarea - Desarrollo página informativa contacto

TAREA	
Número: T028	Numero de Historia: H010
Nombre: Desarrollo página informativa contacto	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 04/07/2022	Fecha fin: 05/07/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa contacto, creación del formulario de contacto utilizando Email.js.	

Elaborado por: El investigador

Tabla 53. Tarea - Desarrollo página informativa ediciones anteriores

TAREA	
Número: T029	Numero de Historia: H010
Nombre: Desarrollo página informativa ediciones anteriores	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 06/07/2022	Fecha fin: 07/07/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo página informativa ediciones anteriores, mostrar la información extraída de la base de datos en la página ediciones anteriores.	

Elaborado por: El investigador

Tabla 54. Tarea - Desarrollo de modulo registro

TAREA	
Número: T030	Numero de Historia: H011
Nombre: Desarrollo de modulo registro	
Tipo de tarea: Desarrollo	Puntos estimados: 7
Fecha inicio: 08/07/2022	Fecha fin: 18/07/2022
Programador responsable: Omar Fonseca	

Descripción: Desarrollo de modulo registro para que el usuario se inscriba en el congreso con sus datos personales.

Elaborado por: El investigador

Tabla 55. Tarea - Desarrollo módulo validación

TAREA	
Número: T031	Numero de Historia: H012
Nombre: Desarrollo módulo validación	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 19/07/2022	Fecha fin: 21/07/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo módulo validación para que el usuario valide su inscripción en el congreso mediante el registro de su información de pago.	

Elaborado por: El investigador

Tabla 56. Tarea - Desarrollo módulo facturas

TAREA	
Número: T032	Numero de Historia: H013
Nombre: Desarrollo módulo facturas	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 22/07/2022	Fecha fin: 26/07/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo módulo facturas para que el usuario consulte su factura electrónica.	

Elaborado por: El investigador

Tabla 57. Tarea - Desarrollo módulo certificados

TAREA	
Número: T033	Numero de Historia: H014
Nombre: Desarrollo módulo certificados	
Tipo de tarea: Desarrollo	Puntos estimados: 3

Fecha inicio: 27/07/2022	Fecha fin: 29/07/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo módulo certificados para que el usuario consulte sus certificados.	

Elaborado por: El investigador

Tabla 58. Tarea - Desarrollo submódulo control de inscripciones

TAREA	
Número: T034	Numero de Historia: H015
Nombre: Desarrollo submódulo control de inscripciones	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Fecha inicio: 01/08/2022	Fecha fin: 04/08/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo submódulo control de inscripciones: Desarrollo de funcionalidad carga de facturas, visualización de comprobantes y validación de la inscripción (a este módulo solo se le permitirá ingresar a los administradores del congreso CSEI22).	

Elaborado por: El investigador

Tabla 59. Tarea - Desarrollo submódulo reportes

TAREA	
Número: T035	Numero de Historia: H015
Nombre: Desarrollo submódulo reportes	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha inicio: 05/08/2022	Fecha fin: 08/08/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo submódulo reportes: Agregar funcionalidad de visualizar reportes y descargarlos (a este módulo solo se le permitirá ingresar a los administradores del congreso CSEI22).	

Elaborado por: El investigador

Tabla 60. Tarea - Desarrollo submódulo certificados

TAREA	
Número: T036	Numero de Historia: H015
Nombre: Desarrollo submódulo certificados	
Tipo de tarea: Desarrollo	Puntos estimados: 4
Fecha inicio: 09/08/2022	Fecha fin: 12/08/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo submódulo certificados: En este módulo se podrá generar los certificados de los participantes en el congreso tanto para participación, asistencia, autores y expositores (a este módulo solo se le permitirá ingresar a los administradores del congreso CSEI22).	

Elaborado por: El investigador

Tabla 61. Tarea - Desarrollo submódulo comité

TAREA	
Número: T037	Numero de Historia: H015
Nombre: Desarrollo submódulo comité	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 15/08/2022	Fecha fin: 17/08/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo submódulo comité: Permitir insertar, editar, eliminar y validar a los integrantes del comité, generar sus respectivos certificados (a este módulo solo se le permitirá ingresar a los administradores del congreso CSEI22).	

Elaborado por: El investigador

Tabla 62. Tarea - Desarrollo submódulo gestión usuarios

TAREA	
Número: T038	Numero de Historia: H015
Nombre: Desarrollo submódulo gestión usuarios	
Tipo de tarea: Desarrollo	Puntos estimados: 3
Fecha inicio: 18/08/2022	Fecha fin: 22/08/2022
Programador responsable: Omar Fonseca	

Descripción: Desarrollo submódulo gestión de usuarios: Desarrollar la funcionalidad para gestionar a los usuarios del sistema administrativo del congreso (a este módulo solo se le permitirá ingresar a los administradores del congreso CSEI22).

Elaborado por: El investigador

Tabla 63. Tarea - Desarrollo submódulo configuración sistema

TAREA	
Número: T039	Numero de Historia: H015
Nombre: Desarrollo submódulo configuración sistema	
Tipo de tarea: Desarrollo	Puntos estimados: 24
Fecha inicio: 23/08/2022	Fecha fin: 26/09/2022
Programador responsable: Omar Fonseca	
Descripción: Desarrollo submódulo configuración sistema: Este módulo únicamente podrá ser accedido por las personas autorizados para realizar sus respectivas modificaciones en este caso los administradores, desde este módulo se podrá realizar la gestionar toda la información de la página de inicio.	

Elaborado por: El investigador

Plan de entrega

El objetivo del plan de entrega es cronometrar cada historia de usuario utilizando la metodología XP, los puntos asignados están contemplados en días, tomando en cuenta a que un día representa 6 horas de trabajo, cada iteración durará 3 semanas y media.

Tabla 64. Planificación de entrega

NO	HISTORIA DE USUARIO	TIEMPO ESTIMADO		ITERACIÓN ASIGNADA							
		DÍA	HORA	1	2	3	4	5	6	7	
		S	S								
H01	Estructurar el proyecto	4	24	x							
H02	Diseño de la base de datos	5	30	x							

H03	Diseño de interfaz modulo informativo	10	60	x						
H04	Diseño de interfaz módulo registro	5	30		x					
H05	Diseño de interfaz módulo validación	4	24		x					
H06	Diseño de interfaz módulo facturas	4	24		x					
H07	Diseño de interfaz módulo certificados	3	18		x					
H08	Diseño de interfaz módulo configuración	15	90			x				
H09	Inicio de sesión	3	18			x				
H10	Desarrollo módulo informativo.	15	90				x			
H11	Desarrollo módulo registro	7	42				x			
H12	Desarrollo módulo validación	3	18					x		
H13	Desarrollo módulo facturas	3	18					x		
H14	Desarrollo módulo certificados	3	18					x		
H15	Desarrollo módulo configuración	40	240						x	x

Elaborado por: El investigador

3.2.2 Fase II: Diseño del proyecto

Después de recopilar datos que indiquen las necesidades del usuario, la segunda fase de la metodología XP sirve como guía y ayuda para iniciar el proyecto de codificación.

3.2.2.1 Metáfora de la aplicación

En el enfoque de desarrollo de XP, diseñar un proyecto implica desarrollar una metáfora de aplicación. En este caso, la aplicación se ha dividido en diferentes módulos en función de la información recopilada. Empezamos a desarrollar una metáfora para cada módulo porque no era posible proporcionar una metáfora para todo el proyecto dado que consta de diferentes partes.

El módulo informativo permitirá a los usuarios visualizar la información sobre el congreso CSEI22 en el cual se visualizarán los submódulos: inicio, sobre el evento, conferencista, comité, programa, envió de trabajos, contacto, ediciones anteriores.

A continuación, el módulo de registro permitirá a los usuarios realizar su registro para el congreso CSEI22, además de informarle el valor a cancelar para la participación de este.

El siguiente módulo corresponde a la validación, en este módulo los usuarios podrán subir sus comprobantes de pago para posteriormente ser validados por el administrador del sistema.

El módulo de facturas permitirá al usuario revisar la factura emitida por la institución pública.

También se cuenta con el módulo de certificados en el cual los usuarios podrán acceder una vez allá culminado el congreso para revisar sus respectivos certificados.

Adicionalmente se encuentra el módulo de configuración en donde se podrá realizar la modificación de todo el sitio web y también contará con los siguientes submódulos: control de inscripciones, reportes, certificados, configuración, comité, usuarios.

3.2.2.2 Iteración I

Tabla 65. Estimación de la primera iteración

Iteración	Identificador	Nombre historia	Tiempos estimados	
			Días	Horas
1	H001	Estructurar el proyecto	4	24
1	H002	Diseño de la base de datos	5	30
1	H003	Diseño de interfaz modulo informativo	10	60
Total:			19	114

Elaborado por: El investigador

Estructurar el proyecto.

Hardware

- Laptop Lenovo y50 2016
- Procesador Intel Core i7-4720HQ
- Memoria RAM: 16 GB
- Servidor Linux.

Software

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

Una vez definido el entorno del sistema, procedemos con el desarrollo de este:

Creación del proyecto

Para crear un nuevo proyecto en angular se ingresa el comando `ng new` (nombre del proyecto), una vez realizado esto solicitará una serie de preguntas para la configuración inicial del proyecto como si desea crear archivos de ruta dentro de la aplicación y que tipo de hoja de estilo se va a utilizar.

Ilustración 18. Inicialización del proyecto

```
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? (Use arrow keys)
> CSS
SCSS  [ https://sass-lang.com/documentation/syntax#scss ]
Sass  [ https://sass-lang.com/documentation/syntax#the-indented-syntax ]
Less  [ http://lesscss.org ]
```

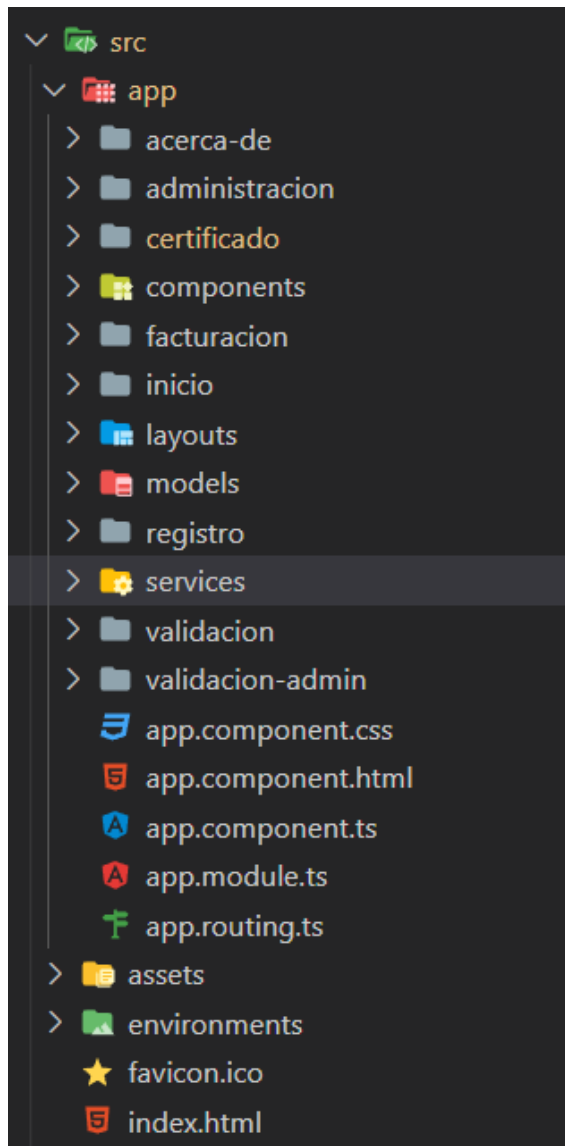
Elaborado por: El investigador

Estructura del proyecto

La estructura del proyecto está conformada de la siguiente manera:

- Páginas de la aplicación: En estas carpetas se encontrarán las diferentes páginas de la aplicación
- Componentes: En esta carpeta se encontrarán las secciones de la aplicación los cuales son reutilizables.
- Modelos: En esta carpeta se encuentran las interfaces.
- Servicios: En esta carpeta se encuentran las conexiones con el api.
- Assets: En donde se encontrarán las imágenes, archivos js y css de la aplicación.
- Environments: En donde se definirán las variables del sistema tales como la dirección del api.

Ilustración 19. Estructura del proyecto

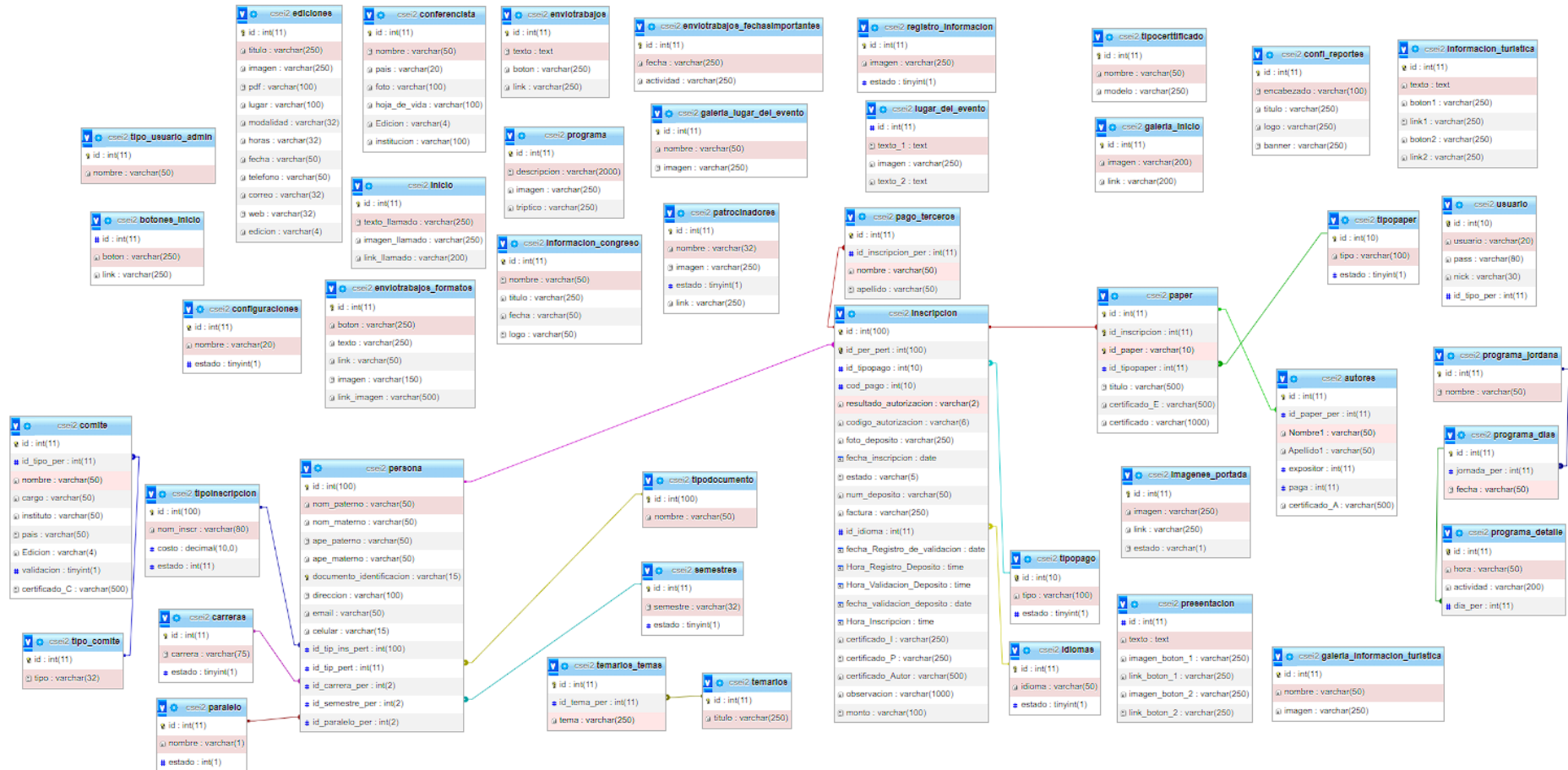


Elaborado por: El investigador

Diseñador de base de datos.

El diseño de la base de datos se realizó según la información proporcionada por los miembros organizadores del congreso CSEI22 con el fin de cumplir todos sus requerimientos.

Ilustración 20. Estructura de la base de datos



Elaborado por: El investigador

Diseño de interfaz módulo informativo

En la presente pantalla se mostrará la información del congreso la cual será modificable desde el módulo de configuración, en esta pantalla se podrá visualizar las páginas de inicio, sobre el evento, conferencistas, comité, programa, envío de trabajos, contacto y ediciones anteriores.

Ilustración 21. Diseño de interfaz módulo informativo



Elaborado por: El investigador

3.2.2.3 Iteración II

Tabla 66. Estimación de la segunda iteración

Iteración	Identificador	Nombre historia	Tiempos estimados	
			Días	Horas
2	H004	Diseño de interfaz módulo registro	5	30

2	H005	Diseño de interfaz módulo validación	4	24
2	H006	Diseño de interfaz módulo facturas	4	24
2	H007	Diseño de interfaz módulo certificados	3	18
Total			16	96

Elaborado por: El investigador

Diseño de interfaz módulo registro

En la siguiente pantalla se mostrará un formulario el cual solicitará el número de cedula para continuar con el proceso de registro.

Ilustración 22. Diseño de interfaz módulo registro 1

Elaborado por: El investigador

En la siguiente pantalla se mostrará un formulario el cual deberá ser completado por el participante, si estos datos no son llenados de forma correcta no se le permitirá avanzar en el proceso de registro

Ilustración 23. Diseño de interfaz módulo registro 2

The screenshot shows a web interface for registration. On the left is a dark red sidebar with the CSEI logo and the text 'CONGRESO DE CIENCIA DE LA COMPUTACION, ELECTRONICA E INDUSTRIAL'. Below the logo are five menu items: INFORMACION, REGISTRO (highlighted), VALIDACION, CERTIFICADOS, and FACTURAS. At the bottom of the sidebar is the copyright notice 'Copyright © Universidad Técnica de Ambato'. The main content area has a dark red header with the word 'REGISTRO'. Below the header is a white registration form with the following fields: 'Tipo de identificación' (dropdown menu), 'Documento de identificación' (text input with value '1829181625'), 'Nombres Completos' (text input with value 'Juan Sebastian'), 'Apellidos Completos' (text input with value 'Colcha Pozo'), 'Dirección' (text input), 'Tipo de inscripción' (dropdown menu), 'Teléfono' (text input with value '0936251521'), 'Email' (text input with value 'persona@gmail.com'), and 'Lenguaje' (dropdown menu). A dark red button labeled 'SIGUIENTE' is positioned at the bottom center of the form.

Elaborado por: El investigador

En la presente pantalla el participante deberá seleccionar el método de pago y finalizará la inscripción.

Ilustración 24. Diseño de interfaz módulo registro 3

The screenshot shows the same web interface as in Ilustración 23, but the main content area is now titled 'REGISTRO' and displays a 'Forma de pago' section. At the top of this section is an icon of two hands holding a stack of money. Below the icon, the text 'Forma de pago' is displayed. There are two radio button options: 'Tarjeta de crédito/debito' (which is currently unselected) and 'Efectivo' (which is selected). A dark red button labeled 'SIGUIENTE' is positioned at the bottom center of the form.

Elaborado por: El investigador

Diseño de interfaz módulo validación

En la presente pantalla el participante ingresará con su número de cedula y podrá subir el comprobante de pago al sistema, únicamente se aceptarán imágenes png y jpg con un tamaño inferior a 2MP.

Ilustración 25. Diseño de interfaz módulo validación



Elaborado por: El investigador

Diseño de interfaz módulo facturas

En la presente pantalla el participante ingresará con su número de cedula y tendrá la opción de revisar y descargar su factura desde el sistema.

Ilustración 26. Diseño de interfaz módulo facturas



Elaborado por: El investigador

Diseño de interfaz módulo certificados

En la presente pantalla el participante ingresará con su número de cedula y podrá revisar los certificados generados por el sistema, en esta sección se podrán visualizar los certificados de asistencia, participación, autores y expositores

Ilustración 27. Diseño de interfaz módulo certificados



Elaborado por: El investigador

3.2.2.4 Iteración III

Tabla 67. Estimación de la tercera iteración

Iteración	Identificado	Nombre historia	Tiempos estimados	
			Días	Horas
3	H008	Diseño de interfaz módulo de configuración	15	90
3	H009	Inicio de sesión	3	18
Total			18	108

Elaborado por: El investigador

Diseño de interfaz módulo de configuración.

En la siguiente pantalla se mostrarán los submódulos de Control de inscripciones, reportes, Certificados, Configuración, Comité y Usuarios los cuales se gestionarán de la siguiente manera:

Control de inscripciones. – Gestionar a las personas inscritas en el congreso.

Reportes. – Generar los diferentes reportes necesarios en el sistema.

Certificados. – Generar los certificados para cada uno de los participantes en el congreso.

Configuración. – Modificar la información del sitio web en general.

Comité. – Gestionar el comité participante en el congreso.

Usuarios. – Gestionar los usuarios que tienen acceso al panel de configuración.

Ilustración 28. Diseño de interfaz módulo configuración



Elaborado por: El investigador

Inicio de sesión.

En esta pantalla se deberá ingresar las respectivas credenciales para poder acceder al módulo de configuración

Ilustración 29. Inicio de sesión

CSEI - ADMINISTRACION



USUARIO:
CONTRASEÑA:
INGRESAR

Copyright © Universidad Técnica de
Ambato

Elaborado por: El investigador

3.2.2.5 Iteración IV

Tabla 68. Estimación de la cuarta iteración

Iteración	Identificado	Nombre historia	Tiempos estimados	
			Días	Horas
4	H010	Desarrollo módulo informativo.	15	90
4	H011	Desarrollo módulo registro	7	42
Total:			22	132

Elaborado por: El investigador

Desarrollo módulo informativo

Página de inicio

En la presente pantalla se mostrará la información principal acerca del congreso CSEI22, esta información es extraída de la tabla inicio de la base de datos.

Ilustración 30. Página de inicio

Organiza: Universidad Técnica de Ambato, Facultad de Ingeniería en Sistemas, Electrónica e Industrial

PUBLICACION : Springer

INDEXADO : Scopus®

EXTENSOS: information

Artículo científico	\$378,00
Estudiantes	\$54
Público en general	\$108

Modalidad: Virtual

40 Horas

Objetivo: Difundir el desarrollo científico, tecnológico empresarial de las últimas avances de la ciencia de la computación, la electrónica y la ingeniería industrial.

Agenda:

- Fecha límite de envío: 31 de julio del 2022
- Notificación de aceptación: 15 de agosto del 2022

Elaborado por: El investigador

Página sobre el evento

En la presente pantalla se mostrará información general sobre el congreso como el lugar del evento y la información turística, esta información es extraída de la tabla sobre el evento de la base de datos.

Ilustración 31. Página de sobre el evento

La Universidad Técnica de Ambato y las instituciones patrocinadoras, convocan a la comunidad científica y profesional al IV CONGRESO de Ciencia de la Computación, Electrónica e Industrial, CSEI 2022, que se realizará del 07 al 11 de noviembre de 2022 en Ambato, Ecuador. La convocatoria se realiza teniendo en cuenta el I, II y III CONGRESO de Ciencia de la Computación, Electrónica e Industrial, CSEI 2019, CSEI 2020 y CSEI 2021, el éxito de las Jornadas Internacionales de Investigación Científica de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, que han tenido lugar desde el año 2013 al 2017, con la participación de investigadores de Ecuador, España, Cuba, Colombia, entre otros. El congreso es organizado por académicos e investigadores de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial con el apoyo de investigadores de universidades de España, Colombia, Chile, México, USA, Japon. El Congreso de Ciencia de la Computación, Electrónica e Industrial pretende convertirse en un espacio para la actualización, entrenamiento y difusión científica en el área de la ciencia de la computación, la electrónica y la ingeniería industrial. En su primera edición, esperamos contar con la presencia de investigadores de diversas partes del mundo y prestigiosos conferencistas en las temáticas del congreso. Se consideran, como temas del congreso: desarrollo de software, inteligencia artificial, informática educativa, e-gestión, comunicaciones, sistemas integrados, sistemas interactivos, automatización de la producción, producción y operaciones y optimización basada en análisis numérico. En el marco de celebración del congreso, además de los cursos pre-congresos, el día 28 de octubre, las conferencias magistrales y las presentaciones de los artículos, tendremos paneles de discusión en las temáticas del congreso. Estas actividades permitirán conocer del avance científico y tecnológico en ciencia de la computación, electrónica e industrial. Este evento representa una especial oportunidad para el intercambio de ideas, establecer contactos profesionales y académicos y la organización del congreso se compromete a dar la más amplia difusión de los trabajos que sean aceptados, los mismos que serán indexados en Scopus.

DESCARGAR EL CALL FOR PAPERS

VER PROGRAMA

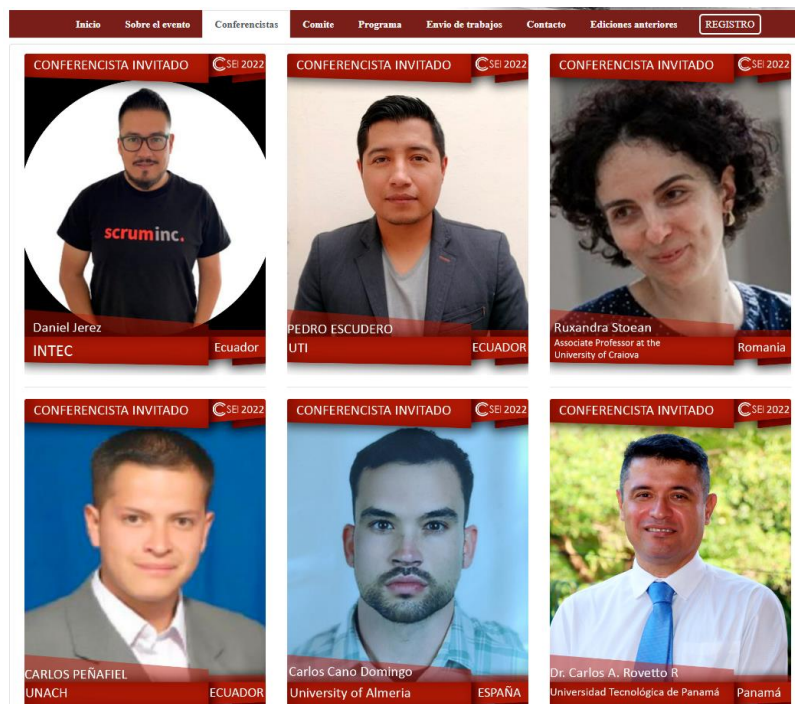
Elaborado por: El investigador

Página de conferencistas

En la presente pantalla se mostrará la información sobre los conferencistas del congreso, al dar clic sobre su imagen se abrirá una pestaña con el respectivo documento

pdf con información sobre el conferencista, esta información es extraída de la tabla de conferencistas de la base de datos.

Ilustración 32. Página de conferencistas



Elaborado por: El investigador

Página de comité

En la presente pantalla se presentara el listado del comité tanto organizador como científico que participara en el congreso CSEI22, esta información es extraída de la tabla comite la base de datos.

Ilustración 33. Página de comité

Inicio Sobre el evento Conferencistas Comite Programa Envío de trabajos Contacto Ediciones anteriores REGISTRO		
ORGANIZADOR		CIENTIFICO
Pilar Urrutia Urrutia Decana	Carlos Sánchez Rosero Subdecano	Carlos Israel Nuñez Miranda Coordinador de Investigación
Clay Aldás Flores Coordinador de Carrera Ingeniería en Sistemas	Christian Mariño Rivera Coordinador de Carrera Ingeniería Industrial	Félix Fernández Peña Docente Carrera de Sistemas
Marcelo García-Sánchez Docente Carrera de Industrial	Carlos Gordón Gallegos Docente Carrera de Electrónica	Fernando Ibarra Torres

Elaborado por: El investigador

Página de programa

En la presente pantalla se visualizará el cronograma del evento CSEI22 adicionalmente se podrá descargar el mismo, esta información es extraída desde la base de datos desde las tablas programa, dia programa, actividad programa, hora programa.

Ilustración 34. Página de programa

MODALIDAD VIRTUAL: Descarga el formato para la presentación.

DESCARGAR FORMATO DE PRESENTACION Y PROGRAMA

Mañana

Lunes 7 Nov 2022	Martes 8 Nov 2022	Miércoles 9 Nov 2022	Jueves 10 Nov 2022	Viernes 11 Nov 2022
09:00 - 09:30 INAUGURACION CSEI 2022 Autoridades	09:00 - 09:30 Conferencia magistral Expositor Ecuador Vanessa Salazar Caracterización de señales P300 mediante Causalidad de Granger en el contexto de Interfaces Humano-Maquina	09:00 - 10:00 Conferencia magistral Expositor España Carlos Cano Domingo On the use of ELF electromagnetic signals for ionosphere characterization: From signal processing to Deep Neural Network approaches.	09:00 - 10:00 Conferencia magistral Expositor Ecuador Carlos Peñañal Multifield antenna designed by Characteristic Modes Theory	09:00 - 10:00 Conferencia magistral Expositor Ecuador Pedro Escudero Sensores y microfónica, de las aplicaciones Biomédicas a la Infotelia
09:30 - 10:30 Conferencia magistral Expositor Ecuador Daniel Jerez Mayorga Seguridad informática agil	09:30 - 10:30 Conferencia magistral Expositor Panamá Carlos A. Rovetto Enfoque para optimizar la producción y demanda de energía eléctrica mediante sistemas de inecuaciones y modelo de regresión.	10:00 - 10:30 (ID064) Drone collaboration using OLSR protocol in a smart network for traffic monitoring in a smart city environment	10:00 - 10:30 (ID016) Comparison of results between Hypermedia and Simuloid Applied in the Structural Simulation in a car lift	10:00 - 10:30 (ID067) Plant distribution based on a resilient approach in textile SMEs
10:30 - 11:30 Conferencia magistral Expositor Rumania Roxandra Niculescu Deep learning for computational support in the restoration of cultural heritage artefacts	10:30 - 11:00 (ID031) Mobile marketing as a communication strategy in politics	10:30 - 11:00 (ID038) Solar Panels for Low Power Energy Harvesting	10:30 - 11:00 (ID062) Application of CFD for the redesign of a positive pressure mechanical flow generation device in covid treatment 19	10:30 - 11:00 (ID054) Diagnosis of digital maturity of SMEs in the province of Imbabura - Ecuador
11:30 - 12:00 (ID032) PCAnEn - Handcasting with Analogic Ensembles of Neural Networks			11:00 - 11:30	11:00 - 11:30 (ID068) Model production based on...

Elaborado por: El investigador

Página de envío de trabajos

En la presente pantalla se mostrara información acerca del envío de trabajos además de los temarios y las fechas importantes acerca del evento, esta información es extraída desde la base de datos desde la tabla envío de trabajos.

Ilustración 35. Página de envío de trabajos

INFORMACION TEMARIO FECHAS IMPORTANTES

El comité organizador del congreso convoca a la comunidad académica y científica a presentar artículos originales e innovadores en las áreas de interés. Los trabajos serán evaluados por el Comité Científico Internacional del Congreso en un proceso de revisión por pares doble ciego. En el texto del artículo para evaluación por el Comité Científico no deberá incluirse información que permita la identificación de los autores, sus datos, incluyendo filiaciones, no deberán aparecer en la versión inicial. Esta información solamente se incluirá en la versión final del artículo. Los artículos enviados no deben estar publicados ni en revista ni en ninguna otra conferencia o publicación nacional o internacional. Los trabajos deberán ser enviados con el formato de la Revista Técnica de Sistemas y Tecnologías de la Información o con formato Springer.

ENVIAR TRABAJOS

Lecture Notes in Networks and Systems
Q4
Computer Networks and Communications
Nov 2021
SJR 2021 0.15
powered by scimago.com

FORMATOS

FORMATOS PARA SPRINGER WORD
FORMATO PARA SPRINGER WORD

FORMATOS PARA SPRINGER LATEX
FORMATO PARA SPRINGER LATEX

Elaborado por: El investigador

Página de contacto

En la presente pantalla se visualizará un formulario con el cual los participantes del congreso pueden contactarse con los miembros organizadores del congreso.

Ilustración 36. Página de contacto

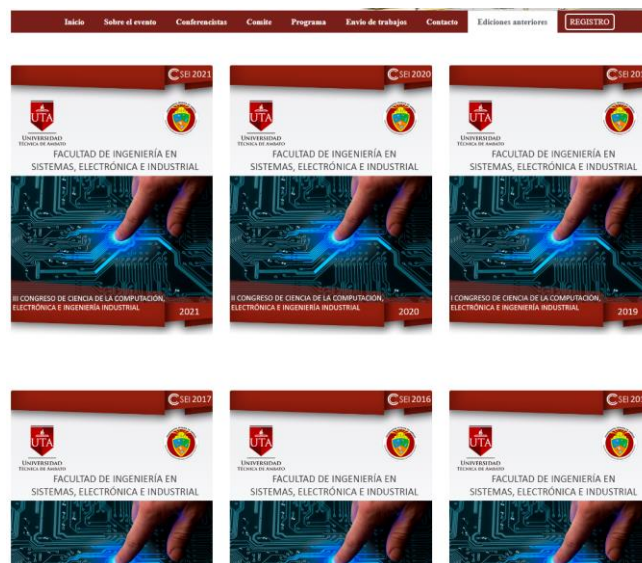
The screenshot shows a contact form on a website. At the top, there is a navigation menu with the following items: Inicio, Sobre el evento, Conferencistas, Comité, Programa, Envío de trabajos, Contacto, Ediciones anteriores, and REGISTRO. The main content area is divided into two columns. The left column has a dark red background and contains the following text: "Gracias por su interés en el CONGRESO DE CIENCIA DE LA COMPUTACION, ELECTRONICA E INDUSTRIAL. Proporcione la siguiente información sobre sus necesidades para ayudarnos a brindarle un mejor servicio. Esta información nos permitirá enviar su solicitud a la persona adecuada. Debería recibir una respuesta dentro de las 48 horas." and "Correo electrónico: csei@uta.edu.ec". The right column has a white background and is titled "Envía un mensaje". It contains four input fields: "Nombres *", "Telefono / Celular", "Dirección de correo electrónico", and "Mensaje *". Below these fields is a dark red button labeled "Enviar".

Elaborado por: El investigador

Página de ediciones anteriores

En la presente pantalla se mostrara el listado de las anteriores ediciones del congreso, esta información es extraída desde la base de datos desde la tabla ediciones.

Ilustración 37. Página de ediciones anteriores



Elaborado por: El investigador

Desarrollo módulo registro

Ingreso del módulo de registros

En la presente pantalla mostrará una caja de texto la cual solicitará el ingreso del documento de identidad para poder continuar con el proceso de registro.

Ilustración 38. Ingreso al módulo de registro

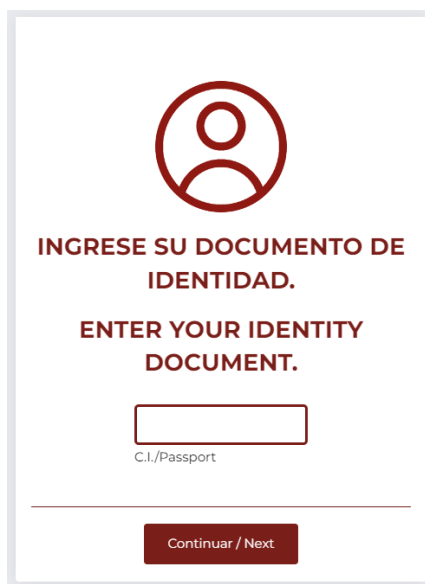


Ilustración 38 muestra una pantalla de inicio de sesión con un ícono de usuario en rojo. El texto principal indica: "INGRESE SU DOCUMENTO DE IDENTIDAD." y "ENTER YOUR IDENTITY DOCUMENT." Debajo hay un campo de entrada de texto con el placeholder "C.I./Passport" y un botón rojo que dice "Continuar / Next".

Elaborado por: El investigador

Formulario de registro

En la presente pantalla presentara un formulario de registro en el cual se solicitará los datos necesarios para la inscripción del congreso, adicionalmente el formulario es dinámico en caso de que sea (profesional, estudiante, autor o autor uta). Si la persona ya se encuentra inscrito el sistema buscara en la base de datos desde la tabla persona e inscripción y presentara los datos para ser editados.

Ilustración 39. Formulario de registro

The screenshot shows a registration form with the following fields and options:

- Tipo de Identificación/ Identification Type:** A dropdown menu with 'IDENTIFICATION CARD' selected.
- Documento de Identificación/ Identification Document:** An empty text input field.
- Nombres Completos/ Complete Names:** An empty text input field.
- Apellidos Completos/ Complete Surnames:** An empty text input field.
- Dirección/ Address:** A wide empty text input field.
- Tipo de Inscripción/ Type of Inscription:** A dropdown menu with 'STUDENT' selected.
- Carrera/Degree:** A dropdown menu with 'Seleccione...' selected.
- Semestre/Semester:** A dropdown menu with 'Seleccione...' selected.
- Paralelo/Parallel:** A dropdown menu with 'Seleccione...' selected.
- Red text:** 'Seleccione el semestre en el que tenga mas creditos.'
- Teléfono/ Phone Number:** An empty text input field.
- Email:** An empty text input field.
- Lenguaje/Language:** An empty text input field.

Elaborado por: El investigador

Selección de método de pago

En la presente pantalla se seleccionará el método de pago.

Ilustración 40. Selección de método de pago

The screenshot shows a payment selection screen with the following elements:

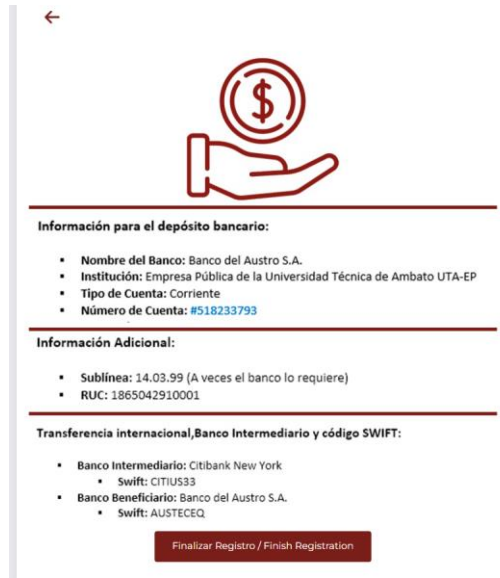
- Back arrow:** A red arrow pointing left in the top left corner.
- Icon:** A red icon of a document with a dollar sign.
- Total a pagar / Total to pay:** The text 'Total a pagar / Total to pay:' followed by '50 \$' in a larger font.
- Forma de pago / Way to pay:** A section header in bold red text.
- Radio button:** A selected radio button (blue dot) next to the text 'Deposito ó Transferencia / Deposit or Transfer'.
- Continuar / Next button:** A dark red button with white text 'Continuar / Next' at the bottom.

Elaborado por: El investigador


Finalizar la inscripción

En la presente pantalla se presentarán los datos bancarios a los cuales se tendrá que realizar el pago para que se valide la inscripción.

Ilustración 41. Finalizar la inscripción



←



Información para el depósito bancario:

- Nombre del Banco: Banco del Austro S.A.
- Institución: Empresa Pública de la Universidad Técnica de Ambato UTA-EP
- Tipo de Cuenta: Corriente
- Número de Cuenta: #518233793

Información Adicional:

- Sublínea: 14.03.99 (A veces el banco lo requiere)
- RUC: 1865042910001

Transferencia internacional, Banco Intermediario y código SWIFT:

- Banco Intermediario: Citibank New York
 - Swift: CITIUS33
- Banco Beneficiario: Banco del Austro S.A.
 - Swift: AUSTECEQ

Finalizar Registro / Finish Registration

Elaborado por: El investigador

3.2.2.6 Iteración V

Tabla 69. Estimación de la quinta iteración

Iteración	Identificador	Nombre historia	Tiempos estimados	
			Días	Horas
5	H012	Desarrollo módulo validación	3	18
5	H013	Desarrollo módulo facturas	3	18
5	H014	Desarrollo módulo certificados	3	18
Total:			9	54

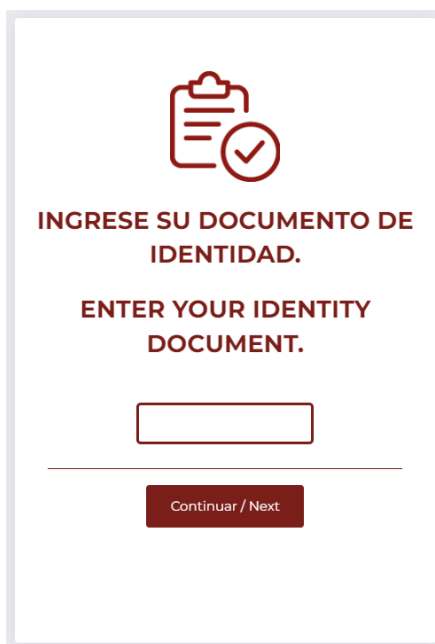
Elaborado por: El investigador

Desarrollo módulo validación

Ingreso de módulo de validación

En la presente pantalla mostrará una caja de texto la cual solicitará el ingreso del documento de identidad para poder continuar con el proceso de validación.

Ilustración 42. Ingreso módulo de validación



The image shows a user interface for entering an identity document. At the top, there is a red icon of a clipboard with a checkmark. Below the icon, the text reads "INGRESE SU DOCUMENTO DE IDENTIDAD." and "ENTER YOUR IDENTITY DOCUMENT." in red. Underneath the text is a red-outlined rectangular input field. Below the input field is a red button with the text "Continuar / Next" in white.

Elaborado por: El investigador

Formulario de validación

La presente pantalla presenta un formulario en el cual el participante podrá subir su comprobante de pago para posteriormente ser validado, el comprobante de pago deberá subirse en formato JPG o PNG y deberá pesar 2MB o menos. Si el usuario ya subió su comprobante anteriormente el sistema buscara en la tabla inscripción y presentara un mensaje preguntando si desea continuar con la edición.

Ilustración 43. Formulario de validación



VALIDACIÓN / VALIDATION

C.I / Passport Registrado / Registered

Imagen de su comprobante de depósito/transferencia Número de comprobante de depósito
Image of your deposit/transfer receipt Deposit receipt number

Seleccionar archivo... / Selected File

Nota: extensión de la imagen (jpg) y tamaño (menos de 2 MB) Toca aquí para ver un ejemplo

Observación/Observation

En caso de requerir factura con otros datos ingresarlos aquí./If you require an invoice with other data, enter it here.
En caso de tener problemas con su comprobante ingresar aquí./If you have problems with your receipt, enter here.


Elaborado por: El investigador

Desarrollo módulo facturas

Ingreso de módulo de facturas

En la presente pantalla se mostrará una caja de texto la cual solicitará el ingreso del documento de identidad para poder visualizar la factura del participante

Ilustración 44. Ingreso módulo de facturas



INGRESE SU DOCUMENTO DE IDENTIDAD.

ENTER YOUR IDENTITY DOCUMENT.

Elaborado por: El investigador

Módulo de facturas

En la presente pantalla se mostrará la factura del participante la cual también permitirá ser descargada. La información será extraída desde la tabla inscripción

Ilustración 45. Módulo de facturas



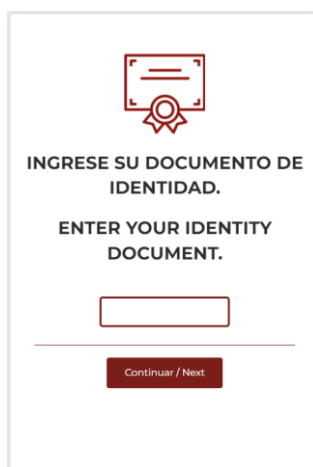
Elaborado por: El investigador

Desarrollo módulo certificados

Ingreso módulo de certificados

En la presente pantalla se mostrará una caja de texto la cual solicitará el ingreso del documento de identidad para poder visualizar los certificados del participante.

Ilustración 46. Ingreso módulo de certificados



Elaborado por: El investigador

Módulo de certificados

En el presente módulo se presentarán los certificados generados del participante tanto los certificados de participación, asistencia, autores y expositores. Esta información será tomada desde la tabla inscripción en donde se encuentran las rutas de los certificados, en caso de también se generen certificados de autores y expositores se procederá a revisar esta información desde la tabla autor y paper.

Ilustración 47. Módulo de certificados

Querido participante:

Gracias por participar en CSEI. Puede descargar sus certificados a continuación:

Certificados:	Descargar
Participación	Descargar
Asistencia	Descargar
Certificados de papers	
Drone collaboration using OLSR protocol in a fanet network for traffic monitoring in a smart city environment	Descargar
Application of the MQTT protocol for the control of a Scorbob Robot by means of EGG electroencephalographic signals.	Descargar
Application of CFD for the redesign of a positive pressure mechanical flow generation device in covid treatment 19	Descargar
Certificados de expositores	
- Drone collaboration using OLSR protocol in a fanet network for traffic monitoring in a smart city environment	Descargar

Elaborado por: El investigador

3.2.2.7 Iteración VI y VII

Tabla 70. Estimación de la sexta y séptima iteración

Iteración	Identificador	Nombre historia	Tiempos estimados	
			Días	Horas
6 y 7	H015	Desarrollo módulo configuración	40	240
Total:			40	240

Elaborado por: El investigador

Submódulo control de inscripciones

La presente pantalla permitirá realizar la gestión de las personas inscritas en el congreso CSEI22 entendiéndose por gestionar que permitirá editar, validar y enviar correos a los inscritos en el congreso. Esta información viene de un inner join entre la tabla inscripción y la tabla persona.

Ilustración 48. Submódulo control de inscripciones

Estado	Número	DI	Apellidos	Nombres	Tipo inscrito	Tipo Pago	Comprobante	Número Deposito	Correo	Celular	Dirección	Factura	Correo	Editar
✓	1				AUTNDR	CASH								
✓	2				AUTNDR	CASH								
✓	3				AUTNDR	CASH								
✓	4				AUTNDR	CASH								
✓	5				AUTNDR UTA	CASH								
✓	6				AUTNDR UTA	CASH								
✓	7				AUTNDR UTA	CASH								

Elaborado por: El investigador

Submódulo de reportes

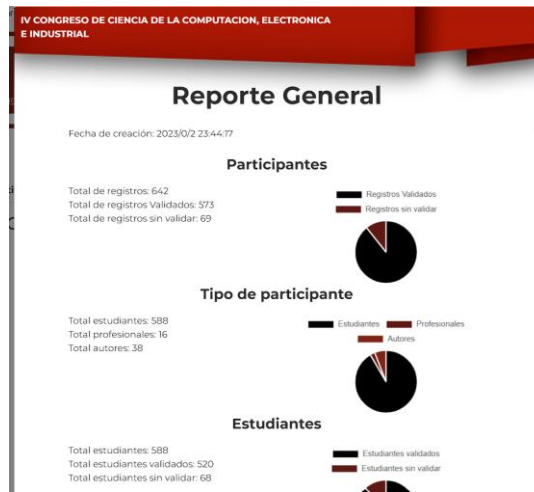
La presente pantalla permitirá la generación de los reportes correspondientes del sistema.

Ilustración 49. Submódulo de reportes



Elaborado por: El investigador

Ilustración 50. Reporte general



Elaborado por: El investigador

Ilustración 51. Reportes filtrados

IV CONGRESO DE CIENCIA DE LA COMPUTACION, ELECTRONICA E INDUSTRIAL

Inscritos validados

#	DI	Apellidos	Nombres	Tipo inscrito	Costo
1	FW809163	TRINKS	MAICKEL JOSUE	AUTHOR	350
2	493344	MACHADO DA SILVA	ROGERIO	AUTHOR	350
3	0105670202001	REINO CHERREZ	FABIOLA SUSANA	AUTHOR	350
4	1712789989	MONTALVO LOPEZ	WILLIAM MANUEL	AUTHOR	350

Elaborado por: El investigador

Ilustración 52. Reportes paper

IV CONGRESO DE CIENCIA DE LA COMPUTACION, ELECTRONICA E INDUSTRIAL

Reporte Papers

ID: 002
TITLE: Drone design for urban fire mitigation
AUTHORS:
 • FELIX MELCHOR SANTOS LOPEZ
 • ROBERT HUMBERTO PINEDO PIMENTEL
 • JOSE BALBUENA
 • EULOGIO GUILLERMO SANTOS DE LA CRUZ
AUTHOR PAYS:
 • FELIX MELCHOR SANTOS LOPEZ
AUTHOR EXPOSES:
 • ROBERT HUMBERTO PINEDO PIMENTEL

Elaborado por: El investigador

Submódulo de certificados asistencia y participación

La presente pantalla permitirá generar los certificados de asistencia y participación para las personas que se encuentren previamente validadas. Esta información será consultada en la tabla inscripción y persona.

Ilustración 53. Submódulo de certificados asistencia y participación

Certificado Asistencia	Certificado Participación	Número DI	Apellidos	Nombres	Tipo Inscrito	Tipo Pago	Correo	Celular	Dirección
✓	✓	1			AUTHOR	CASH	mackel@unsa@yahoo.com.br	550781942254	SMRVV 25, C1 A, LT 6, CASA A
✓	✓	2			AUTHOR	CASH	rogerio@ceci@gmail.com	550791942254	925 414 N. 302, 35A SUL, BRASLIA - DF
✓	✓	3			AUTHOR	CASH	fabola.rehco@ucuenca.edu.ec	098429776	EUGENIO ESPEDO 2120 Y ANTONIO FARFAN
✓	✓	4			AUTHOR	CASH	wmorales@uta.edu.ec	098553388	AV 12 DE OCTUBRE
✓	✓	5			AUTHOR UTA	CASH	jgarcia@uta.edu.ec	098423370	AV PEDRO VICENTE MALDONADO Y PASAJE ESPOSA
✓	✓	6			AUTHOR UTA	CASH	jumoreno09@gmail.com	0983044524	GARCIA MORENO Y VARGAS TORRES, AMBATO - ECUADOR
✓	✓	7			AUTHOR UTA	CASH	je.naranjo@uta.edu.ec	098424886	AMBATO
✓	✓	8			AUTHOR	CASH	mymarcumbajin@uta.edu.ec	098267808	AV ACRIAHONTE Y MANUELITA SAENZ
✓	✓	9			AUTHOR	CASH	gremolobae@unimam.edu.pe	97071448	AV TINGO MARIA 122 CERCADO DE LIMA, LIMA, PERU

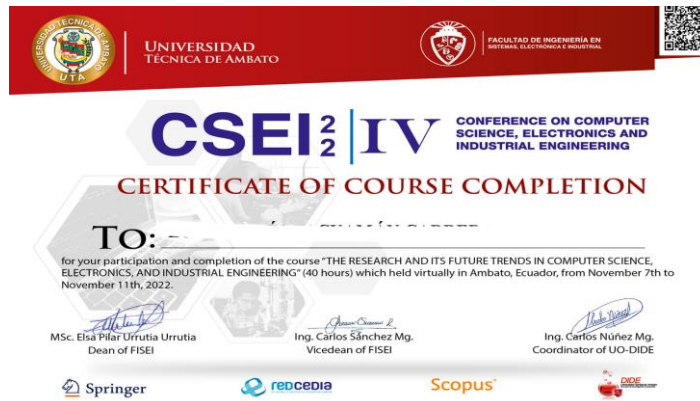
Elaborado por: El investigador

Ilustración 54. Certificado de asistencia



Elaborado por: El investigador

Ilustración 55. Certificado de participación



Elaborado por: El investigador

Submódulo de certificados de autores

La presente pantalla permitirá generar los certificados de los autores que hayan sido previamente validados. Esta información consultara mediante un inner join entre la tabla paper y autor.

Ilustración 56. Submódulo de certificados de autores

Asistencia y Participación		Autores	Expositores
Generar certificados Autor			
Certificado Autor	CI	Nombre	Título paper
<input checked="" type="checkbox"/>			SPRINGER Development of a controller using the generalized minimum variance algorithm for a twin rotor mimo system (TRMS)
<input checked="" type="checkbox"/>			SPRINGER Statistical model for production planning in a vehicle assembler applying lean manufacturing
<input checked="" type="checkbox"/>			SPRINGER Information and Communication Technologies Adoption Model for SMEs: Case Studies
<input checked="" type="checkbox"/>			SPRINGER Failure of Tech Startups: A Systematic Literature Review
<input checked="" type="checkbox"/>			SPRINGER A Flawless-based approach to efficient layout planning for a tire company
<input checked="" type="checkbox"/>			SPRINGER RF Energy Harvesting System based on Spiral Logarithmic Dipole Rectenna Array
<input checked="" type="checkbox"/>			SPRINGER Optimizing User Information Value in a Web Search through the Whisker Index

Elaborado por: El investigador

Ilustración 57. Certificado autores

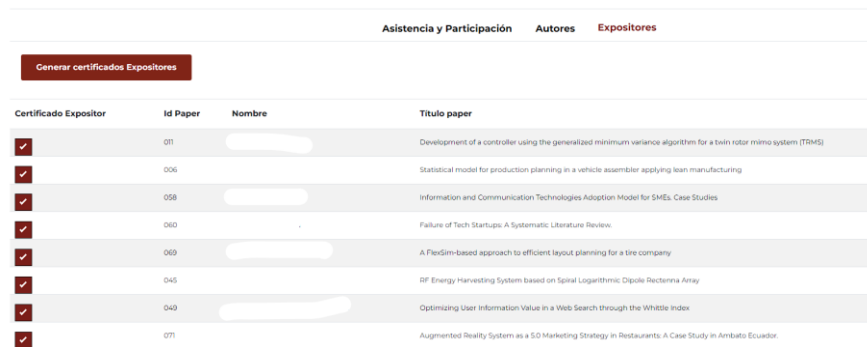


Elaborado por: El investigador

Submódulo de certificados de expositores

La presente pantalla permitirá generar los certificados de los expositores que hayan sido previamente validados. Esta información consultara mediante un inner join entre la tabla paper y autor.

Ilustración 58. Submódulo de certificados de expositores



Certificado Expositor	Id Paper	Nombre	Titulo paper
✓	011	[Redacted]	Development of a controller using the generalized minimum variance algorithm for a twin rotor mimo system (TRMS)
✓	006	[Redacted]	Statistical model for production planning in a vehicle assembler applying lean manufacturing
✓	058	[Redacted]	Information and Communication Technologies Adoption Model for SMEs. Case Studies
✓	060	[Redacted]	Failure of Tech Startups: A Systematic Literature Review
✓	069	[Redacted]	A FleqSim-based approach to efficient layout planning for a tire company
✓	045	[Redacted]	RF Energy Harvesting System based on Spiral Logarithmic Dipole Rectenna Array
✓	049	[Redacted]	Optimizing User Information Value in a Web Search through the Whittle Index
✓	071	[Redacted]	Augmented Reality System as a SD Marketing Strategy in Restaurants: A Case Study in Ambato Ecuador

Elaborado por: El investigador

Ilustración 59. Certificados expositores



Elaborado por: El investigador

Submódulo administración del comité

La presente pantalla permitira validar y generar los certificados del comité organizador y científico. Esta información sera consultada desde la tabla comité.

Ilustración 60. Submódulo administración del comité

Id	Validación	Certificado	Tipo	Nombre	Cargo	Instituto	País	Edición	Opciones
1	<input type="checkbox"/>		COMITE ORGANIZADOR	Pilar Urutia Urutia	Decana			2022	
2	<input type="checkbox"/>		COMITE ORGANIZADOR	Carlos Sánchez Rosero	Subdecano			2021	
3	<input type="checkbox"/>		COMITE ORGANIZADOR	Carlos Israel Nuñez Miranda	Coordinador de Investigación			2021	

Elaborado por: El investigador

Submódulo gestión de usuarios

La presente pantalla permitira realizar la gestión de los usuarios del sistema, entienda se por gestión el agregar, editar, eliminar los usuairos(esta función solo estara disponible para el usuario administrador). Esta información sera consultada desde la tabla usuarios.

Ilustración 61. Submódulo gestión de usuarios

#	Usuario	Tipo Usuario	Opciones
1	caradimon	ADMINISTRADOR	
2	carap	EMPRESA_PUBLICA	

Elaborado por: El investigador

Submódulo de configuración – Opciones de inscripción

En la presente pantalla del submódulo de configuración el usuario podra habilitar o deshabilitar las opciones de registro y validación dentro del sistema. Esta información sera consultada desde la tabla configuración.

Ilustración 62. Submódulo de configuración – Opciones de inscripción

Opciones de inscripción	Patrocinadores	Información Registro	Ediciones	Revistas	Carreras	Semestres	Tipo de inscripción	Idiomas	Conferencistas
Información congreso	Imágenes Portada	Configuración Certificados	Galería Lugar	Galería información	Presentación	Lugar del evento	Información turística	Programa	

Habilitar Register

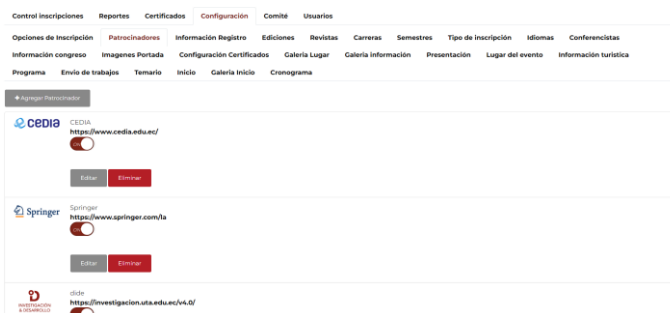
Habilitar Validation

Elaborado por: El investigador

Submódulo de configuración – Patrocinadores

En la presente pantalla del submódulo de configuración el usuario podrá gestionar los patrocinadores del evento, entendiéndose por gestionar que podrá agregar, editar y eliminar los patrocinadores. Esta información será consultada desde la tabla patrocinadores.

Ilustración 63. Submódulo de configuración – Patrocinadores

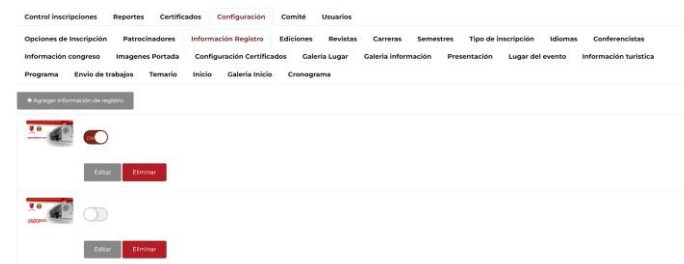


Elaborado por: El investigador

Submódulo de configuración – Información del registro

En la presente pantalla del submódulo de configuración el usuario podrá gestionar las imágenes que aparecieran en el slider de la sección de registro, entendiéndose por gestionar que podrá agregar, editar y eliminar las imágenes que contendrán el slider. Esta información será consultada desde la tabla información_registro.

Ilustración 64. Submódulo de configuración – Información del registro



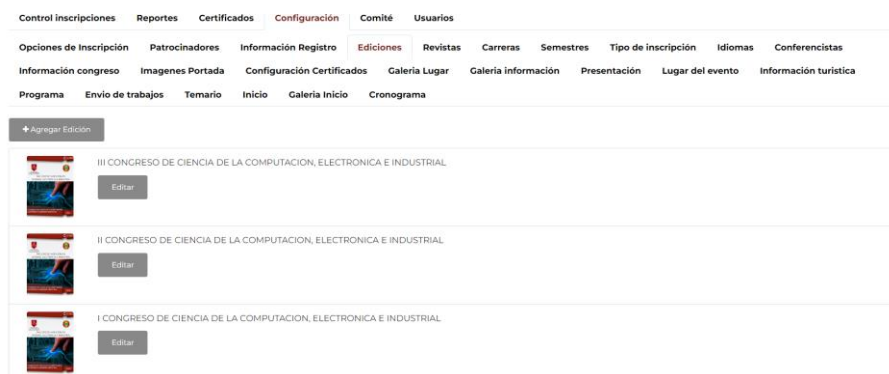
Elaborado por: El investigador

Submódulo de configuración – Ediciones

En la presente pantalla del submódulo de configuración el usuario podrá gestionar las ediciones anteriores realizadas por la institución, entendiéndose por gestionar que podrá

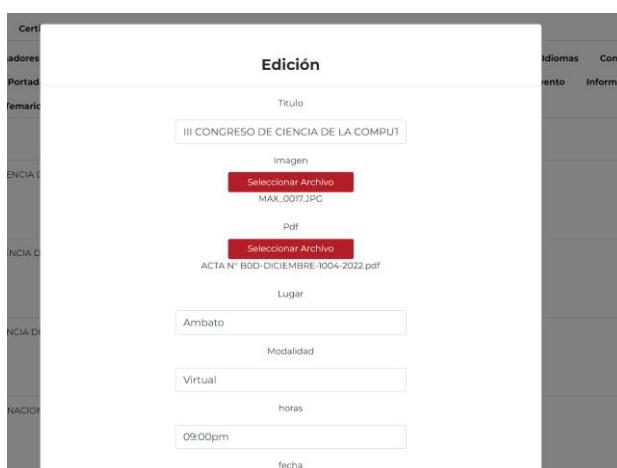
agregar y editar las ediciones anteriores. Esta información sera consultada desde la tabla ediciones.

Ilustración 65. Submódulo de configuración – Ediciones



Elaborado por: El investigador

Ilustración 66. Agregar nueva edición

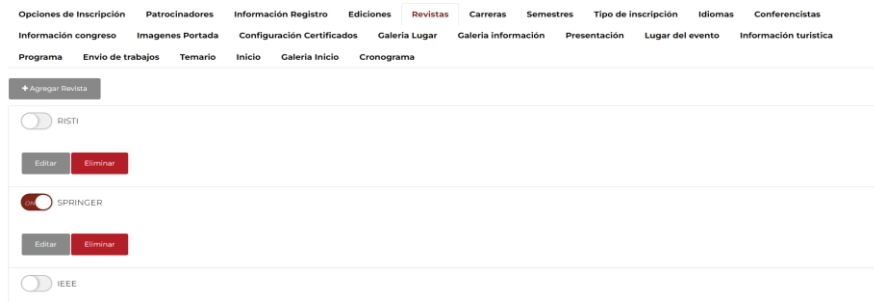


Elaborado por: El investigador

Submódulo de configuración – Revistas

En la presente pantalla del submódulo de configuración el usuario podra gestionar los formatos de revistar que podran ser ingresadas al momento de realizar el registro, entiendase por gestionar que podra agregar, editar y eliminar los formatos. Esta información sera consultada desde la tabla. Esta información sera consultada desde la tabla revistas

Ilustración 67. Submódulo de configuración – Revistas



Elaborado por: El investigador

Ilustración 68. Agregar revista

Revista

Nombre

Estado

ON

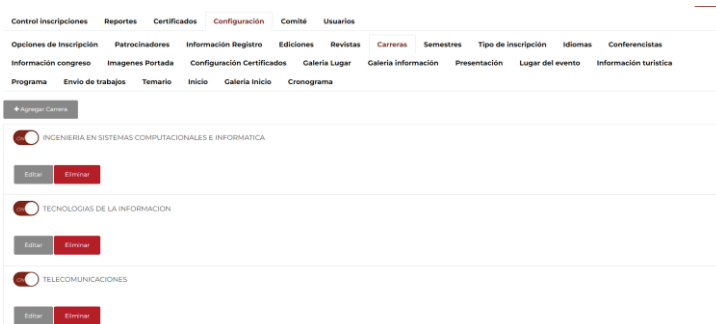
Cancelar Agregar

Elaborado por: El investigador

Submódulo de configuración – Carreras

En la presente pantalla del submódulo de configuración el usuario podrá gestionar las carreras que podrán seleccionar los estudiantes al momento de realizar su inscripción al congreso, entendiéndose por gestionar que podrá agregar, editar y eliminar las carreras. Esta información será consultada desde la tabla carrera.

Ilustración 69. Submódulo de configuración – Carreras

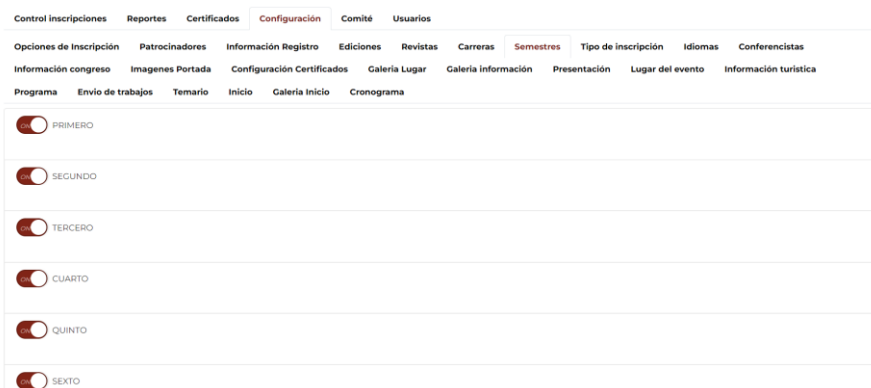


Elaborado por: El investigador

Submódulo de configuración – Semestres

En la presente pantalla del submódulo de configuración el usuario podrá gestionar los cursos que podran seleccionar los estudiantes al momento de realizar su inscripción al congreso, entiendase por gestionar que podra habilitar o deshabilitar los cursos. Esta información sera consultada desde la tabla semetre.

Ilustración 70. Submódulo de configuración – Semestres

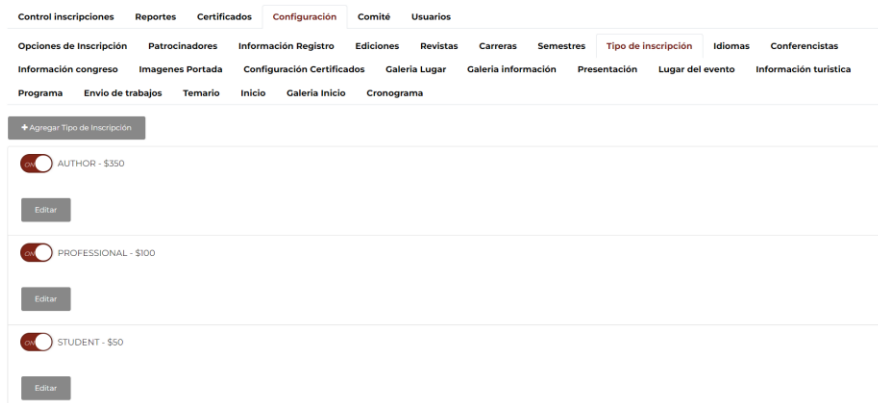


Elaborado por: El investigador

Submódulo de configuración – Tipo de inscripción

En la presente pantalla del submódulo de configuración el usuario podrá gestionar los tipos de inscripción que los interesados podran seleccionar al momento de realizar su inscripción al congreso, entiendase por gestionar que podra habilitar o deshabilitar el tipo de inscripción ademas de editarlo y agregar nuevos. Esta información sera consultada desde la tabla tipo_inscripcion.

Ilustración 71. Submódulo de configuración – Tipo de inscripción

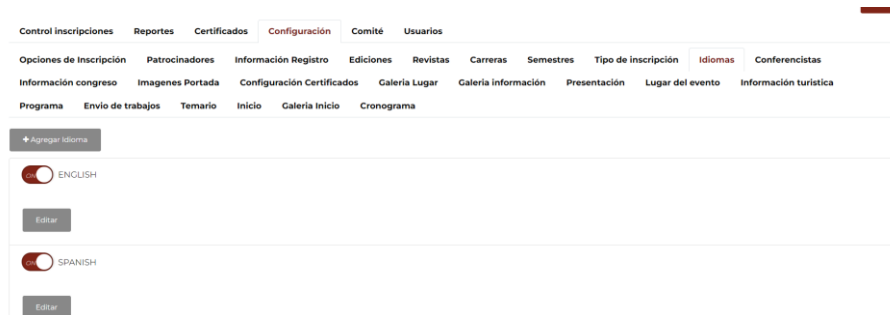


Elaborado por: El investigador

Submódulo de configuración – Idiomas

En la presente pantalla del submódulo de configuración el usuario podrá gestionar lenguajes que maneja el sistema, entendiéndose por gestionar que podrá habilitar o deshabilitar el idioma además de editarlo y agregar nuevos idiomas. Esta información será consultada desde la tabla idiomas.

Ilustración 72. Submódulo de configuración – Idiomas

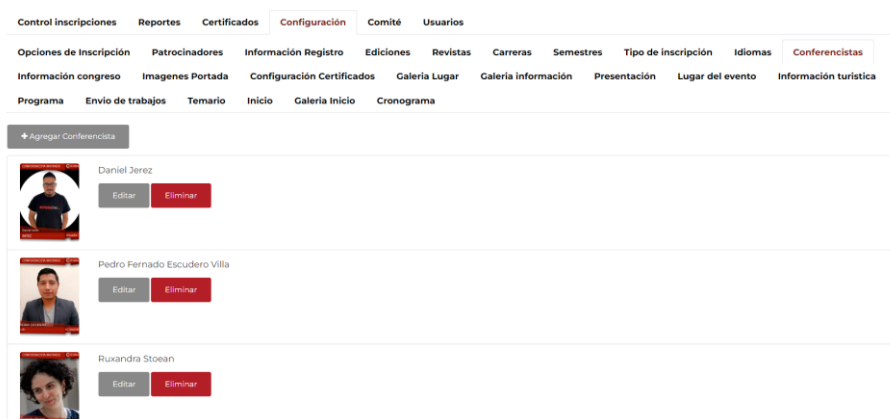


Elaborado por: El investigador

Submódulo de configuración – Conferencistas

En la presente pantalla del submódulo de configuración el usuario podrá gestionar los conferencistas del congreso, entendiéndose por gestionar que podrá agregar, editar y eliminar los conferencistas. Esta información será consultada desde la tabla conferencistas

Ilustración 73. Submódulo de configuración – Conferencistas



Elaborado por: El investigador

Submódulo de configuración – Información del congreso

En la presente pantalla del submódulo de configuración el usuario podrá gestionar la información que apareciera en la cabecera de las paginas, entendiase por gestionar que podrá editar dicha información. Esta información sera consultada desde la tabla informacion_congreso.

Ilustración 74. Submódulo de configuración – Información del congreso

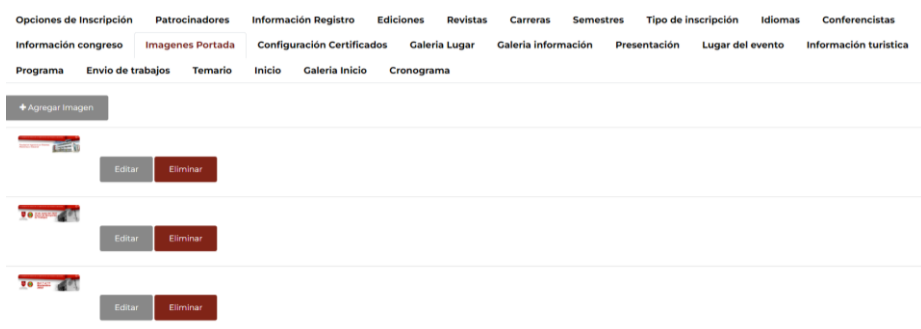


Elaborado por: El investigador

Submódulo de configuración – Imágenes de portada

En la presente pantalla del submódulo de configuración el usuario podrá gestionar las imágenes del slider que aparecieran en la página informativa, entendiase por gestionar que podrá agregar, editar y eliminar dichas imágenes. Esta información sera consultada desde la tabla imágenes_portada

Ilustración 75. Submódulo de configuración – Imágenes de portada



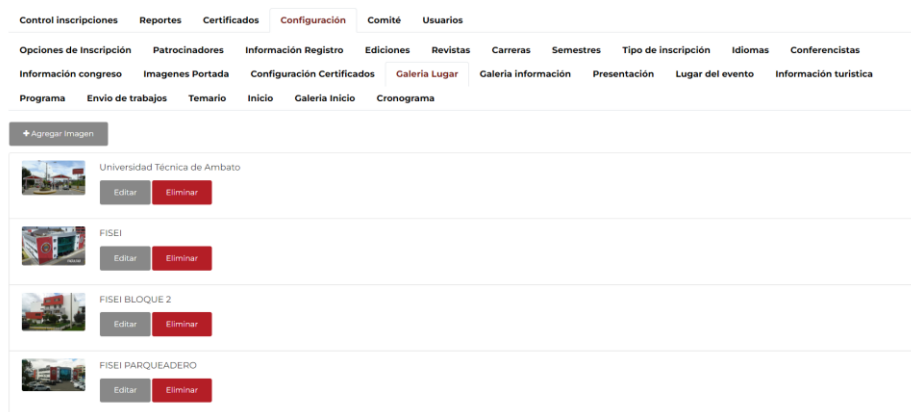
Elaborado por: El investigador

Submódulo de configuración – Galería del lugar

En la presente pantalla del submódulo de configuración el usuario podrá gestionar las imágenes que aparecieran en la sección de sobre el evento de la página informativa,

entiendase por gestionar que podra agregar, editar y eliminar dichas imagenes. Esta información sera consultada desde la tabla galeria_lugar.

Ilustración 76. Submódulo de configuración – Galería del lugar

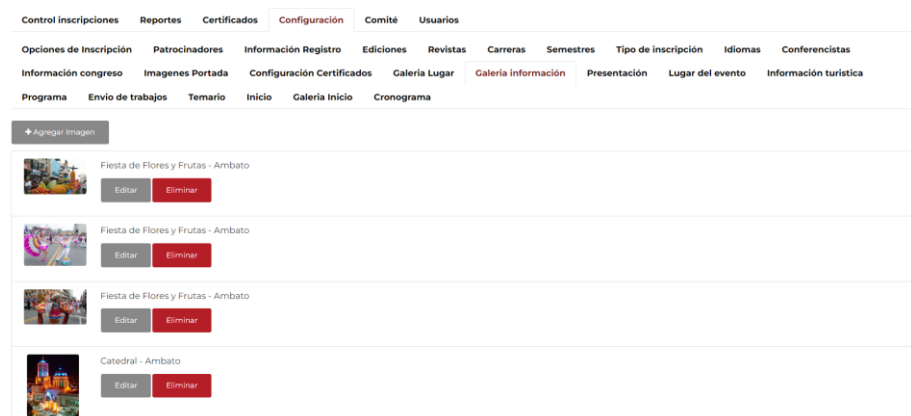


Elaborado por: El investigador

Submódulo de configuración – Galería informativa

En la presente pantalla del submódulo de configuración el usuario podra gestionar las imágenes que aparecan en la sección de información turistica de la página informativa, entiendase por gestionar que podra agregar, editar y eliminar dichas imagenes. Esta información sera consultada desde la tabla galeria_informativa.

Ilustración 77. Submódulo de configuración – Galería informativa



Elaborado por: El investigador

Submódulo de configuración – Presentación

En la presente pantalla del submódulo de configuración el usuario podra gestionar la información de presentación de la página informativa, entiendase por gestionar que

podrá editar dicha información. Esta información será consultada desde la tabla presentación.

Ilustración 78. Submódulo de configuración – Presentación



Elaborado por: El investigador

Submódulo de configuración – Lugar del evento

En la presente pantalla del submódulo de configuración el usuario podrá gestionar la información del lugar del evento de la página informativa, entendiéndose por gestionar que podrá editar dicha información. Esta información será consultada desde la tabla lugar_del_evento.

Ilustración 79. Submódulo de configuración – Lugar del evento



Elaborado por: El investigador

Submódulo de configuración – Información turística

En la presente pantalla del submódulo de configuración el usuario podrá gestionar la información turística de la página informativa, entendiéndose por gestionar que podrá editar dicha información. Esta información será consultada desde la tabla informacion_turistica.

Ilustración 80. Submódulo de configuración – Información turística

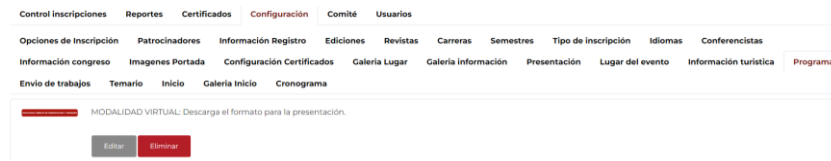


Elaborado por: El investigador

Submódulo de configuración – Programa

En la presente pantalla del submódulo de configuración el usuario podrá gestionar el formato de los artículos que serán presentados en el congreso, entendiéndose por gestionar que podrá editar dicha información. Esta información será consultada desde la tabla programa.

Ilustración 81. Submódulo de configuración – Programa



Elaborado por: El investigador

Submódulo de configuración – Envío de trabajos

En la presente pantalla del submódulo de configuración el usuario podrá gestionar la información sobre los envíos de trabajo de la página informativa, entendiéndose por gestionar que podrá agregar, editar y eliminar dicha información. Esta información será consultada desde la tabla envío_de_trabajos.

Ilustración 82. Submódulo de configuración – Envío de trabajos

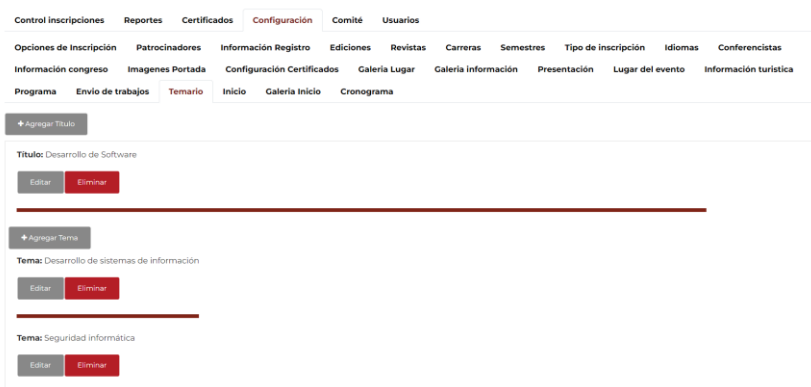


Elaborado por: El investigador

Submódulo de configuración – Temario

En la presente pantalla del submódulo de configuración el usuario podrá gestionar la información del temario de la página informativa, entendiéndose por gestionar que podrá agregar, editar y eliminar dicha información. Esta información será consultada desde la tabla temario, temario_tema, temario_detalle.

Ilustración 83. Submódulo de configuración – Temario



Elaborado por: El investigador

Submódulo de configuración – Inicio

En la presente pantalla del submódulo de configuración el usuario podrá gestionar la información del inicio de la página informativa, entendiéndose por gestionar que podrá editar dicha información. Esta información será consultada desde la tabla inicio.

Ilustración 84. Submódulo de configuración – Inicio



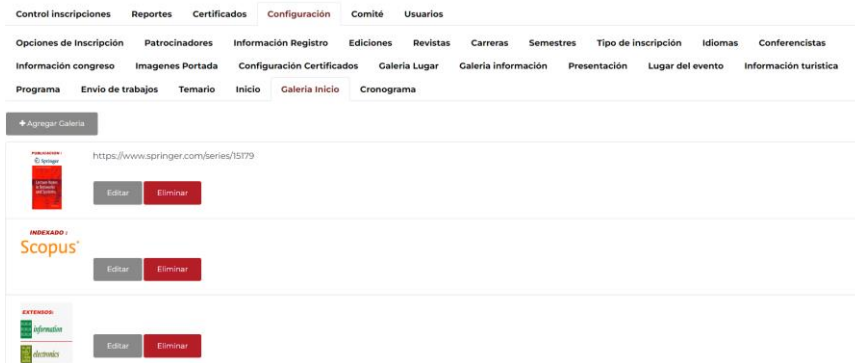
Elaborado por: El investigador

Submódulo de configuración – Galería del inicio

En la presente pantalla del submódulo de configuración el usuario podrá gestionar las imágenes de la galería de inicio de la página informativa, entendiéndose por gestionar que

podrá agregar, editar y eliminar dichas imágenes. Esta información será consultada desde la tabla galería_inicio.

Ilustración 85. Submódulo de configuración – Galería del inicio

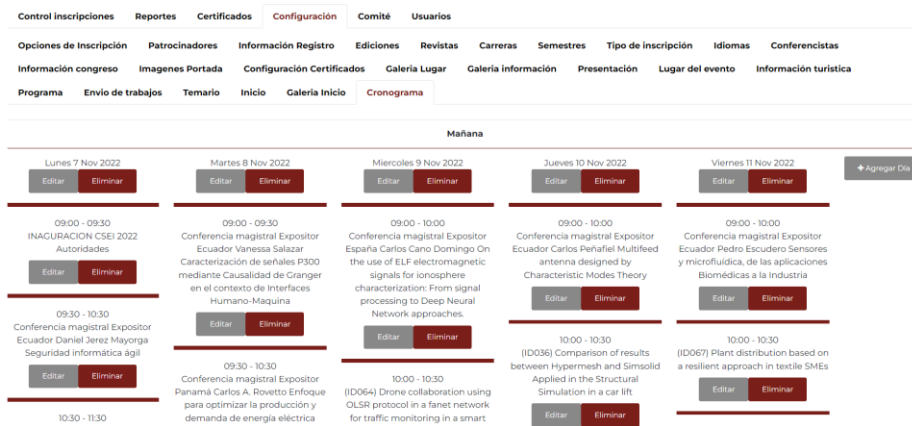


Elaborado por: El investigador

Submódulo de configuración – Cronograma

En la presente pantalla del submódulo de configuración el usuario podrá gestionar el cronograma del evento que apareciera en la página informativa, entendiéndose por gestionar que podrá agregar, editar y eliminar dicha información. Esta información será consultada desde la tabla cronograma.

Ilustración 86. Submódulo de configuración – Cronograma



Elaborado por: El investigador

3.2.3 Fase III: Codificación

3.2.3.1 Métodos del backend Node.js

El presente método valida los datos para realizar el inicio de sesión en la aplicación.

Ilustración 87. Backend - Login

```
public async Login(req: Request, res: Response): Promise<any> {
  const user = req.body.user;
  const pass = req.body.pass;
  const query = await pool.query(
    `SELECT u.id,u.usuario,tu.nombre as rol,u.pass as pass FROM usuario as u INNER JOIN tipo_usuario_admin as tu on(u.id_tipo_per=tu.id) WHERE u.usuario = ?`,
    [user]
  );
  if(query.length===0){
    res.status(400).json({ message: "Usuario o contraseña incorrecto!" });
    return;
  }

  if (!bcryptjs.compareSync(pass, query[0].pass)) {
    res.status(400).json({ message: "Usuario o contraseña incorrecto!" });
    return;
  } else {
    const token = jwt.sign(
      {
        userId: query[0].id,
        username: query[0].usuario,
        "CSEI",
        { expiresIn: "1h" }
      );
    res.json({
      token,
      userId: query[0].id,
      rol: query[0].rol,
    });
  }
}
```

Elaborado por: El investigador

El presente método valida el token para realizar las operaciones de modificación.

Ilustración 88. Backend – Check de Token

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

Elaborado por: El investigador

Ilustración 89. Backend – Guardar archivos

```
const storage = multer.diskStorage({
  destination: `${ruta}`,
  filename: (req, file, cb) => {
    cb(null, file.originalname)
  }
});
export default multer({storage});
```

Elaborado por: El investigador

Ilustración 90. Backend – Consulta listado de inscripciones

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(
      `SELECT i.estado estado,i.id numero, p.documento_identificacion di,CONCAT(ape_paterno, ' ',
ape_materno) apellidos,CONCAT(nom_paterno, ' ', nom_materno) nombres, ti.nom_inscr tipo_Inscrito ,
tp.Tipo tipo_Pago, i.foto_deposito comprobante,i.num_deposito numero_Deposito,p.email correo,
p.celular celular, p.direccion direccion, i.factura factura,p.id carrera_per,p.id semestre_per,
p.id paralelo_per,i.id_per_pert,i.certificado_I, i.certificado_P, idiomas.idioma,i.observacion FROM inscripcion i,persona p, tipopago tp, tipoinscripcion ti, idiomas
WHERE i.id_per_pert = p.id AND i.id_tipopago = tp.id AND ti.id = p.id_tip_ins_pert AND idiomas.id=i.id_idioma ORDER BY i.id ASC`
    );
    res.json(games);
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 91. Backend – Consulta inscripción según cedula

```
public async getInscripcionByCI(req: Request, res: Response): Promise<void> {
  try {
    console.log(req.body)
    const {ci} = req.body;
    const games = await pool.query(
      `SELECT i.estado estado,i.id numero, p.documento_identificacion di,CONCAT(ape_paterno, ' ',
ape_materno) apellidos,CONCAT(nom_paterno, ' ', nom_materno) nombres, ti.nom_inscr tipo_Inscrito ,
tp.tipo tipo_Pago, i.foto_deposito comprobante,i.num_deposito numero_Deposito,p.email correo,
p.celular celular, p.direccion direccion, i.factura factura,p.id carrera_per,p.id semestre_per,
p.id paralelo_per,i.id_per_pert,i.certificado_I,i.certificado_Autor, i.certificado_P,i.observacion FROM inscripcion i,persona p, tipopago tp, tipoinscripcion ti
WHERE i.id_per_pert = p.id AND i.id_tipopago = tp.id AND ti.id = p.id_tip_ins_pert AND p.documento_identificacion=${ci}`
    );
    res.json(games);
    console.log(games)
  } catch (error) {
    console.log("error")
  }
}
```

Elaborado por: El investigador

Ilustración 92. Backend – Actualizar inscripción

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query(`UPDATE inscripcion set ? WHERE id = ?`, [req.body, id]);
    res.json({ message: "La inscripcion fue actualizada" });
  } catch (error) {

  }
}
```

Elaborado por: El investigador

Ilustración 93. Backend – Listar las carreras

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query('SELECT * FROM carreras');
    res.json(games);
  } catch (error) {

  }
}
```

Elaborado por: El investigador

Ilustración 94. Backend – Obtener carrera según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query('SELECT * FROM carreras WHERE id = ?', [id]);
    console.log(games.length);
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "La carrera no existe" });
  } catch (error) {

  }
}
```

Elaborado por: El investigador

Ilustración 95. Backend – Actualizar carrera

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO carreras set ?", [req.body]);
    res.json({ message: 'Carrera Guardada' });
  } catch (error) {

  }
}
```

Elaborado por: El investigador

Ilustración 96. Backend – Actualizar carrera

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE carreras set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "La carrera fue actualizada" });
  } catch (error) {

  }
}
```

Elaborado por: El investigador

Ilustración 97. Backend – Eliminar carrera

```
public async delete(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query('DELETE FROM carreras WHERE id = ?', [id]);
    res.json({ message: "La carrera fue eliminada" });
  } catch (error) {

  }
}
```

Elaborado por: El investigador

Ilustración 98. Backend– Listar comité

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(
      "SELECT c.id, tc.tipo,c.validacion, c.nombre, c.cargo, c.instituto, c.pais, c.edicion,c.certificado,C FROM comite as c, tipo_comite as tc WHERE c.id_tipo_per=tc.id ORDER BY c.id ASC"
    );
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 99. Backend – Obtener comité según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM comite WHERE id = ?", [id]);
    console.log(games.length);
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El comite no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 100. Backend – Agregar comité

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO comite set ?", [req.body]);
    res.json({ message: "Comite Guardada" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 101. Backend – Actualizar comité

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE comite set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "Comite fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 102. Backend – Eliminar comité

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

Elaborado por: El investigador

Ilustración 103. Backend – Listar ediciones

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(
      "SELECT * FROM ediciones ORDER BY id DESC"
    );
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 104. Backend – Crear comité

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));

    obj.imagen[0] ? (req.body.imagen = obj.imagen[0].path) : "";
    obj.pdf[0] ? (req.body.pdf = obj.pdf[0].path) : "";
    const result = await pool.query("INSERT INTO ediciones set ?", [
      req.body,
    ]);
    res.json(result);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 105. Backend – Actualizar comité

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      obj.imagen[0] ? (req.body.imagen = obj.imagen[0].path) : "";
    } catch (error) {}
    try {
      obj.pdf[0] ? (req.body.pdf = obj.pdf[0].path) : "";
    } catch (error) {}
    const { id } = req.params;
    await pool.query("UPDATE ediciones set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "La edición fue actualizada" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 106. Backend – Listar envíos de trabajos

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM enviotrabajos");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 107. Backend – Obtener envío de trabajo según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM enviotrabajos WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 108. Backend – Crear envío de trabajo

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    if (req.file?.path !== undefined) {
      req.body.boton = req.file?.path;
    }
    const result = await pool.query("INSERT INTO enviotrabajos set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 109. Backend – Actualizar envío de trabajos

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.boton = req.file?.path;
    }
    await pool.query("UPDATE enviotrabajos set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 110. Backend – Eliminar envío de trabajos

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

Elaborado por: El investigador

Ilustración 111. Backend – Listar fechas importantes

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(
      "SELECT * FROM enviotrabajos_fechaimportantes"
    );
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 112. Backend – Obtener fecha importante según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM enviotrabajos_fechasimportantes WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 113. Backend – Crear fecha importante

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query(
      "INSERT INTO enviotrabajos_fechasimportantes set ?",
      [req.body]
    );
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 114. Backend – Actualizar fechas importantes

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query(
      "UPDATE enviotrabajos_fechasimportantes set ? WHERE id = ?",
      [req.body, id]
    );
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 115. Backend – Eliminar fecha importante

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

Elaborado por: El investigador

Ilustración 116. Backend – Listar formatos envío de trabajos

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM enviotrabajos_formatos");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 117. Backend – Obtener formato envío de trabajo según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM enviotrabajos_formatos WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 118. Backend – Crear formato envío de trabajo

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    if (req.file?.path !== undefined) {
      req.body.boton = req.file?.path;
    }
    const result = await pool.query(
      "INSERT INTO enviotrabajos_formatos set ?",
      [req.body]
    );
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 119. Backend – Actualizar formatos envío de trabajos

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.boton = req.file?.path;
    }
    await pool.query("UPDATE enviotrabajos_formatos set ? WHERE id = ?",
      req.body,
      id,
    );
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 120. Backend – Eliminar formatos envío de trabajos

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

Elaborado por: El investigador

Ilustración 121. Backend – Listar inscripciones

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(
      `SELECT a.id, CONCAT(a.Nombre1, ' ',a.Apellido1) as nombre, p.titulo,a.certificado_A,p.id_paper FROM inscripcion as i, paper as p,
      autores as a where i.id=p.id_inscripcion AND p.id=a.id_paper_per AND a.expositor=true AND i.estado='V'`
    );
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 122. Backend – Obtener expositor paper según id

```
public async ExpositorPaper(req: Request, res: Response): Promise<void> {
  const {id}=req.params;
  console.log(id)
  try {
    const games = await pool.query(
      `SELECT a.id, CONCAT(a.Nombre1, ' ',a.Apellido1) as nombre, p.titulo,a.certificado_A,p.id_paper FROM inscripcion as i, paper as p,
      autores as a where i.id=p.id_inscripcion AND p.id=a.id_paper_per AND a.expositor=true AND i.estado='V' AND a.id_paper_per=${id}`
    );
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 123. Backend – Actualizar inscripción

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    req.body.factura = req.file?.path;
    await pool.query('UPDATE inscripcion set ? WHERE id_per_pert = ?', [req.body, id]);
    res.json({ message: "Inscripción actualizada" });
  } catch (error) {}
}
}
```

Elaborado por: El investigador

Ilustración 124. Backend – Listar información turística

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(
      "SELECT * FROM galeria_informacion_turistica"
    );
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 125. Backend – Obtener información turística según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM galeria_informacion_turistica WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 126. Backend – Crear galería información turística

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    req.body.imagen = req.file?.path;
    const result = await pool.query(
      "INSERT INTO galeria_informacion_turistica set ?",
      [req.body]
    );
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 127. Backend – Actualizar galería información turística

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    await pool.query(
      "UPDATE galeria_informacion_turistica set ? WHERE id = ?",
      [req.body, id]
    );
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 128. Backend – Eliminar galería información turística

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

Elaborado por: El investigador

Ilustración 129. Backend – Listar galería lugar del evento

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM galeria_lugar_del_evento");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 130. Backend – Crear galería lugar del evento

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    const result = await pool.query(
      "INSERT INTO galeria_lugar_del_evento set ?",
      [req.body]
    );
    res.json({ message: "Registro Creado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 131. Backend – Actualizar galería lugar del evento

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }

    await pool.query(`UPDATE galeria_lugar_del_evento set ? WHERE id = ?`, [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 132. Backend – Eliminar galería lugar del evento

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

Elaborado por: El investigador

Ilustración 133. Backend – Listar tipo de documento de identificación

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM tipodocumento");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 134. Backend – Obtener tipo de documento de identificación según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM tipodocumento WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El documento no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 135. Backend – Crear tipo de documento de identificación

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO tipodocumento set ?", [
      req.body,
    ]);
    res.json({ message: "Documento Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 136. Backend – Actualizar tipo de documento de identificación

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE tipodocumento set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "El documento fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 137. Backend – Eliminar tipo de documento de identificación

```
public async delete(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("DELETE FROM tipodocumento WHERE id = ?", [id]);
    res.json({ message: "El documento fue eliminado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 138. Backend – Listar los idiomas

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM idiomas");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 139. Backend – Obtener idioma según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM idiomas WHERE id = ?", [
      id,
    ]);
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El idioma no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 140. Backend – Crear idioma

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO idiomas set ?", [req.body]);
    res.json({ message: "Idioma Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 141. Backend – Actualizar idioma

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE idiomas set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "El idioma fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 142. Backend – Eliminar idiomas

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

Elaborado por: El investigador

Ilustración 143. Backend – Listar imágenes de portada

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM imagenes_portada");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 144. Backend – Obtener imagen de portada según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM imagenes_portada WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }

    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 145. Backend – Crear imagen de portada

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    req.body.imagen = req.file?.path;
    const result = await pool.query("INSERT INTO imagenes_portada set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 146. Backend – Actualizar imagen de portada

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    await pool.query("UPDATE imagenes_portada set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 147. Backend – Eliminar imagen de portada

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

Elaborado por: El investigador

Ilustración 148. Backend – Listar información del congreso

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM informacion_congreso");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 149. Backend – Actualizar información del congreso

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      obj.logo[0] ? (req.body.logo = obj.logo[0].path) : "";
    } catch (error) {}
    try {
      obj.favicon[0] ? (req.body.favicon = obj.favicon[0].path) : "";
    } catch (error) {}
    await pool.query("UPDATE informacion_congreso set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 150. Backend – Listar información turística

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM informacion_turistica");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 151. Backend – Obtener información turística según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM informacion_turistica WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 152. Backend – Crear información turística

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      req.body.boton1 = obj.boton1[0].path;
    } catch (error) {}
    try {
      req.body.boton2 = obj.boton2[0].path;
    } catch (error) {}
    const result = await pool.query(
      "INSERT INTO informacion_turistica set ?",
      [req.body]
    );
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 153. Backend – Actualizar información turística

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      req.body.boton1 = obj.boton1[0].path;
    } catch (error) {}
    try {
      req.body.boton2 = obj.boton2[0].path;
    } catch (error) {}
    await pool.query("UPDATE informacion_turistica set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 154. Backend – Eliminar información turística

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

Elaborado por: El investigador

Ilustración 155. Backend – Listar información del inicio

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM inicio");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 156. Backend – Obtener información del inicio según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM inicio WHERE id = ?", [id]);

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 157. Backend – Actualizar información del inicio

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen_llamado = req.file?.path;
    }
    await pool.query("UPDATE inicio set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 158. Backend – Eliminar información de inicio

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

Elaborado por: El investigador

Ilustración 159. Backend – Listar galería de inicio

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM galeria_inicio");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 160. Backend – Obtener galería de inicio según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM galeria_inicio WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 161. Backend – Crear galería de inicio

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    const result = await pool.query("INSERT INTO galeria_inicio set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 162. Backend – Actualizar galería de inicio

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    await pool.query("UPDATE galeria_inicio set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 163. Backend – Eliminar galería de inicio

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

Elaborado por: El investigador

Ilustración 164. Backend – Listar inscripciones

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM inscripcion");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 165. Backend – Obtener inscripción según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      `SELECT i.id, i.id_per_pert,i.id_tipopago,i.id_idioma,i.cod_pago,i.resultado_autorizacion,i.codigo_autorizacion,
      i.foto_deposito,i.fecha_inscripcion,i.Hora_Inscripcion,i.estado,i.num_deposito,i.factura,i.id_idioma,i.fecha_Registro_de_validacion,
      i.Hora_Registro_Deposito,i.Hora_Validacion_Deposito,i.fecha_validacion_deposito,i.certificado_I,i.certificado_P,i.certificado_Autor,
      id.idioma FROM inscripcion i,idiomas id WHERE i.id_idioma=id.id and id_per_pert = ?`,
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "Inscripción no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 166. Backend – Crear inscripción

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO inscripcion set ?", [
      req.body,
    ]);
    const id = await pool.query(
      "Select id from inscripcion where id_per_pert = ?",
      [req.body.id_per_pert]
    );
    res.json(id);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 167. Backend – Actualizar inscripción

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE inscripcion set ? WHERE id_per_pert = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Inscripción actualizada" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 168. Backend – Eliminar inscripción

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

Elaborado por: El investigador

Ilustración 169. Backend – Listar conferencistas

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM conferencista");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 170. Backend – Crear conferencista

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      req.body.foto = obj.foto[0].path;
    } catch (error) {}
    try {
      req.body.hoja_de_vida = obj.pdf[0].path;
    } catch (error) {}
    const result = await pool.query("INSERT INTO conferencista set ?", [
      req.body,
    ]);
    res.json(result);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 171. Backend – Actualizar conferencistas

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      req.body.foto = obj.foto[0].path;
    } catch (error) {}
    try {
      req.body.hoja_de_vida = obj.pdf[0].path;
    } catch (error) {}
    const { id } = req.params;
    await pool.query("UPDATE conferencista set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "conferencista actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 172. Backend – Eliminar conferencistas

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

Elaborado por: El investigador

Ilustración 173. Backend – Listar lugares del evento

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM lugar_del_evento");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 174. Backend – Obtener lugar del evento según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM lugar_del_evento WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 175. Backend – Crear lugar del evento

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    const result = await pool.query("INSERT INTO lugar_del_evento set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 176. Backend – Actualizar lugar del evento

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;

    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    await pool.query("UPDATE lugar_del_evento set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 177. Backend – Eliminar lugar del evento

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

Elaborado por: El investigador

Ilustración 178. Backend – Enviar mail a los inscritos

```
public async MailInscritos(req: Request, res: Response): Promise<void> {
  const { mensaje, destinatario, asunto } = req.body;

  try {
    nodeoutlook.sendEmail({
      port: 587,
      auth: {
        user: "csei@uta.edu.ec",
        pass: "Congreso?2021",
      },
      from: "csei@uta.edu.ec",
      to: `${destinatario}`,
      subject: `${asunto}`,
      html: `${mensaje}`,
    });
    res.json({ message: "ok" });
  } catch (error) {
    console.log(error);
    res.json({ message: "error" });
  }
}
```

Elaborado por: El investigador

Ilustración 179. Backend – Enviar mail creación certificados

```
public async CreacionCertificado(ruta:string,destinatario:string,asunto:string){
  const mensaje = `
  <!DOCTYPE html>
<html>
<body style="text-align:center">
<p> Estimado participante su certificado de ${asunto} a sido generado </p>
<p> Puedes contrarlo en la siguiente dirección: </p>
<p> ${ruta} </p>
<hr />
</body>
</html>
`;
  try {
    nodeoutlook.sendEmail({
      port: 587,
      auth: {
        user: "csei@uta.edu.ec",
        pass: "Congreso?2021",
      },
      from: "csei@uta.edu.ec",
      to: `${destinatario}`,
      subject: "Certificado ${asunto}",
      html: `${mensaje}`,
    });
  } catch (error) {
    console.log(error);
  }
}
```

Elaborado por: El investigador

Ilustración 180. Backend – Listar tipos de pago

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM tipopago");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 181. Backend – Obtener tipo de pago según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM tipopago WHERE id = ?", [
      id,
    ]);

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "Tipo de pago no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 182. Backend – Crear tipo de pago

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO tipopago set ?", [req.body]);
    res.json({ message: "Tipo de pago Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 183. Backend – Actualizar tipo de pago

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE tipopago set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "Tipo de pago fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 184. Backend – Eliminar tipo de pago

```
public async delete(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("DELETE FROM tipopago WHERE id = ?", [id]);
    res.json({ message: "Tipo de pago eliminado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 185. Backend – Listar paper

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM paper order by id_paper ASC");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 186. Backend – Obtener paper según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM paper WHERE id_inscripcion = ? order by id DESC", [id]);

    if (games.length > 0) {
      return res.json(games);
    }
    res.status(404).json({ text: "El paper no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 187. Backend – Crear paper

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO paper set ?", [req.body]);
    const id = await pool.query(
      "Select id from paper where id_inscripcion = ?",
      [req.body.id_inscripcion]
    );
    res.json(id);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 188. Backend – Actualizar paper

```
public async update(req: Request, res: Response): Promise<void> {
  console.log(req.body)
  try {
    const { id } = req.params;
    await pool.query("UPDATE paper set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "El paper fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 189. Backend – Eliminar paper

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

Elaborado por: El investigador

Ilustración 190. Backend – Listar paralelos

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM paralelo");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 191. Backend – Obtener paralelo según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM paralelo WHERE id = ?", [
      id,
    ]);

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El paralelo no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 192. Backend – Crear paralelo

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO paralelo set ?", [req.body]);
    res.json({ message: "Paralelo Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 193. Backend – Actualizar paralelo

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE paralelo set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "El paralelo fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 194. Backend – Eliminar paralelo

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

Elaborado por: El investigador

Ilustración 195. Backend – Listar patrocinadores

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM patrocinadores");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 196. Backend – Obtener patrocinador según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM patrocinadores WHERE id_per_pert = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El Patrocinador no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 197. Backend – Crear paralelo

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    req.body.imagen = req.file?.path;
    const result = await pool.query("INSERT INTO patrocinadores set ?", [
      req.body,
    ]);
    res.json(result);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 198. Backend – Actualizar paralelo

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    await pool.query("UPDATE patrocinadores set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Patrocinador actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 199. Backend – Eliminar paralelo

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

Elaborado por: El investigador

Ilustración 200. Backend – Listar personas

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM persona");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 201. Backend – Obtener persona según CI

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM persona WHERE documento_identificacion = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "La persona no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 202. Backend – Crear persona

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO persona set ?", [req.body]);
    const id = await pool.query(
      "Select id from persona where documento_identificacion = ?",
      [req.body.documento_identificacion]
    );
    res.json(id);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 203. Backend – Actualizar persona

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE persona set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "La persona fue actualizada" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 204. Backend – Eliminar persona

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

Elaborado por: El investigador

Ilustración 205. Backend – Listar presentación

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM presentacion");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 206. Backend – Obtener presentación según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM presentacion WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 207. Backend – Crear presentación

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      req.body.imagen_boton_1 = obj.imagen_boton_1[0].path;
    } catch (error) {}
    try {
      req.body.imagen_boton_2 = obj.imagen_boton_2[0].path;
    } catch (error) {}
    const result = await pool.query("INSERT INTO presentacion set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 208. Backend – Actualizar presentación

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    const obj = JSON.parse(JSON.stringify(req.files));
    try {
      req.body.imagen_boton_1 = obj.imagen_boton_1[0].path;
    } catch (error) {}
    try {
      req.body.imagen_boton_2 = obj.imagen_boton_2[0].path;
    } catch (error) {}

    await pool.query("UPDATE presentacion set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 209. Backend – Actualizar presentación

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

Elaborado por: El investigador

Ilustración 210. Backend – Eliminar presentación

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query('SELECT * FROM programa');
    res.json(games);
  } catch (error) {}
}
}
```

Elaborado por: El investigador

Ilustración 211. Backend – Listar programa

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM programa WHERE id = ?", [id]);

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
}
```

Elaborado por: El investigador

Ilustración 212. Backend – Crear programa

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const obj = JSON.parse(JSON.stringify(req.files));
    try { (obj.imagen[0]) ? req.body.imagen = obj.imagen[0].path : ""; } catch (error) {}
    try { (obj.triptico[0]) ? req.body.triptico = obj.triptico[0].path : ""; } catch (error) {}
    const result = await pool.query('INSERT INTO programa set ?', [req.body]);
    res.json({ message: 'Registro Guardado' });
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 213. Backend – Actualizar programa

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    const obj = JSON.parse(JSON.stringify(req.files));
    try { (obj.imagen[0]) ? req.body.imagen = obj.imagen[0].path : ""; } catch (error) {}
    try { (obj.triptico[0]) ? req.body.triptico = obj.triptico[0].path : ""; } catch (error) {}
    await pool.query('UPDATE programa set ? WHERE id = ?', [req.body, id]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 214. Backend – Eliminar programa

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

Elaborado por: El investigador

Ilustración 215. Backend – Listar detalles del programa

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM programa_detalle");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 216. Backend – Obtener detalle del programa según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM programa_detalle WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 217. Backend – Crear detalle del programa

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO programa_detalle set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 218. Backend – Actualizar detalle del programa

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;

    await pool.query("UPDATE programa_detalle set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 219. Backend – Eliminar detalle del programa

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

Elaborado por: El investigador

Ilustración 220. Backend – Listar días del programa

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM programa_dias");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 221. Backend – Obtener día de programa según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM programa_dias WHERE id = ?",
      [id]
    );

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 222. Backend – Crear día de programa

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO programa_dias set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 223. Backend – Actualizar día del programa

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;

    await pool.query("UPDATE programa_dias set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 224. Backend – Eliminar día del programa

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

Elaborado por: El investigador

Ilustración 225. Backend – Listar jornada del programa

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM programa_jordana");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 226. Backend – Obtener jornada del programa

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM programa_jordana WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 227. Backend – Crear jornada del programa

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO programa_jordana set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 228. Backend – Actualizar jornada del programa

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;

    await pool.query("UPDATE programa_jordana set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 229. Backend – Eliminar jornada del programa

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

Elaborado por: El investigador

Ilustración 230. Backend – Listar información del registro

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM registro_informacion");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 231. Backend – Obtener información del registro según
id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM registro_informacion WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 232. Backend – Crear información del registro

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    req.body.imagen = req.file?.path;
    const result = await pool.query(
      "INSERT INTO registro_informacion set ?",
      [req.body]
    );
    res.json(result);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 233. Backend – Actualizar información del congreso

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    if (req.file?.path !== undefined) {
      req.body.imagen = req.file?.path;
    }
    await pool.query("UPDATE registro_informacion set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 234. Backend – Eliminar información del congreso

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

Elaborado por: El investigador

Ilustración 235. Backend – Listar semestres

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM semestres");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 236. Backend – Obtener semestre según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM semestres WHERE id = ?", [
      id,
    ]);

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El semestre no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 237. Backend – Crear semestre

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO semestres set ?", [
      req.body,
    ]);
    res.json({ message: "Semestre Guardada" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 238. Backend – Actualizar semestre

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE semestres set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "El semestres fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 239. Backend – Eliminar semestre

```
public async delete(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("DELETE FROM semestres WHERE id = ?", [id]);
    res.json({ message: "El semestre fue eliminado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 240. Backend – Listar temarios

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM temarios");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 241. Backend – Obtener temario según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM temarios WHERE id = ?", [
      id,
    ]);
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 242. Backend – Crear temario

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO temarios set ?", [req.body]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 243. Backend – Actualizar temario

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE temarios set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 244. Backend – Eliminar temario

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

Elaborado por: El investigador

Ilustración 245. Backend – Listar temas del temario

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM temarios_temas");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 246. Backend – Obtener tema del temario según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM temarios_temas WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 247. Backend – Crear tema del temario

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO temarios_temas set ?", [
      req.body,
    ]);
    res.json({ message: "Registro Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 248. Backend – Actualizar tema del temario

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE temarios_temas set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "Registro actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 249. Backend – Eliminar tema del temario

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

Elaborado por: El investigador

Ilustración 250. Backend – Listar tipo de comité

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query('SELECT * FROM tipo_comite');
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 251. Backend – Obtener tipo de inscripción

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(
      "SELECT * FROM tipoinscripcion WHERE id = ?",
      [id]
    );
    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El tipo de inscripción no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 252. Backend – Crear tipo de inscripción

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO tipoinscripcion set ?", [
      req.body,
    ]);
    res.json({ message: "El tipo de inscripción Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 253. Backend – Actualizar tipo de inscripción

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE tipoinscripcion set ? WHERE id = ?", [
      req.body,
      id,
    ]);
    res.json({ message: "El tipo de inscripción fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 254. Backend – Eliminar tipo de inscripción

```
public async delete(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("DELETE FROM tipoinscripcion WHERE id = ?", [id]);
    res.json({ message: "El tipo de inscripción fue eliminado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 255. Backend – Listar tipo de paper

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query("SELECT * FROM tipopaper");
    res.json(games);
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 256. Backend – Obtener tipo de paper según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query("SELECT * FROM tipopaper WHERE id = ?", [
      id,
    ]);

    if (games.length > 0) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El paper no existe" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 257. Backend – Crear tipo de paper

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const result = await pool.query("INSERT INTO tipopaper set ?", [
      req.body,
    ]);
    res.json({ message: "paper Guardado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 258. Backend – Actualizar tipo de paper

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query("UPDATE tipopaper set ? WHERE id = ?", [req.body, id]);
    res.json({ message: "El paper fue actualizado" });
  } catch (error) {}
}
```

Elaborado por: El investigador

Ilustración 259. Backend – Listar Eliminar tipo de paper

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

Elaborado por: El investigador

Ilustración 260. Backend – Listar tipo de usuario administrador

```
public async list(req: Request, res: Response): Promise<void> {
  try {
    const games = await pool.query(`SELECT u.id,u.usuario,tu.nombre as tipo_usuario from usuario as u INNER JOIN
    tipo_usuario_admin as tu on(u.id_tipo_per=tu.id)`);
    res.json(games);
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 261. Backend – Obtener tipo de usuario administrador según id

```
public async getOne(req: Request, res: Response): Promise<any> {
  try {
    const { id } = req.params;
    const games = await pool.query(`SELECT * FROM usuario WHERE usuario = ?`, [id]);

    if (games.length > 0 && games.length < 2) {
      return res.json(games[0]);
    }
    res.status(404).json({ text: "El registro no existe" });
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 262. Backend – Crear usuario

```
public async create(req: Request, res: Response): Promise<void> {
  try {
    const salt = bcryptjs.genSaltSync(10);
    req.body.pass = bcryptjs.hashSync(req.body.pass, salt);
    const result = await pool.query(`INSERT INTO usuario set ?`, [req.body]);
    res.json( { message: 'Usuario Guardado' });
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 263. Backend – Actualizar usuario

```
public async update(req: Request, res: Response): Promise<void> {
  try {
    const { id } = req.params;
    await pool.query('UPDATE usuario set ? WHERE id = ?', [req.body, id]);
    res.json({ message: "El registro fue actualizado" });
  } catch (error) {
  }
}
```

Elaborado por: El investigador

Ilustración 264. Backend – Eliminar usuario

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

Elaborado por: El investigador

3.2.3.2 Métodos del Frontend Angular

Página de inicio

El presente método obtiene los detalles del evento y los guarda en un array.

Ilustración 265. Frontend – Obtener detalle del programa

```
getProgramaDetalle() {
  this._programaDetalleService.getProgramaDetalle().subscribe((result) => {
    this.programaDetalle = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene los días del programa y los guarda en un array.

Ilustración 266. Frontend – Obtener días del programa

```
getProgramaDias() {
  this._programaDiasService.getProgramaDias().subscribe((result) => {
    this.programaDias = result;
    result.forEach((dia) => {
      dia.jornada_per === 1
        ? this.DiasMatutina.push(dia)
        : this.DiasVespertina.push(dia);
    });
  });
}
```

Elaborado por: El investigador

El presente método obtiene las jornadas del evento y la guarda en un array.

Ilustración 267. Frontend – Obtener jornada del programa

```
getProgramaJornada() {
  this._programaJornadaService.getProgramaJornada().subscribe((result) => {
    this.programaJornada = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene la galería del inicio y la guarda en un array.

Ilustración 268. Frontend – Obtener galería de inicio

```
getInicioGaleria() {
  this._inicioGaleriaService.getInicioGaleria().subscribe((result) => {
    this.inicioGaleria = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene la información del inicio y la guarda en un objeto.

Ilustración 269. Frontend – Obtener información del inicio

```
getInicio() {
  this._inicioService.getInicio().subscribe((result) => {
    this.inicio = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene la información de los temarios y la guarda en un array.

Ilustración 270. Frontend – Obtener temarios

```
getTemarios() {  
  this._temarioService.getTemarios().subscribe((result) => {  
    this.temarios = result;  
  });  
}
```

Elaborado por: El investigador

El presente método obtiene los temas del temario y los guarda en un array.

Ilustración 271. Frontend – Obtener tema de los temarios

```
getTemarioTemas() {  
  this._temarioTemasServices.getTemarioTemas().subscribe((result) => {  
    this.temarioTemas = result;  
  });  
}
```

Elaborado por: El investigador

El presente método obtiene la información de los envíos de trabajo y los guarda en un array.

Ilustración 272. Frontend – Obtener información envío de trabajos

```
getEnvioTrabajosFormatos() {  
  this._envioTrabajosFormatosService  
    .getEnvioTrabajoFormatos()  
    .subscribe((result) => {  
      this.envioTrabajosFormatos = result;  
    });  
}
```

Elaborado por: El investigador

El presente método obtiene la información de las fechas de envío de trabajos y las guarda en un array.

Ilustración 273. Frontend – Obtener información fechas envío de trabajos

```
getEnvioTrabajosFechas() {
  this._envioTrabajosFechasService
    .getEnvioTrabajoFechas()
    .subscribe((result) => {
      this.envioTrabajosFechas = result;
    });
}
```

Elaborado por: El investigador

El presente método obtiene la información del programa y la guarda en un objeto.

Ilustración 274. Frontend – Obtener información del programa

```
getPrograma() {
  this._programaService.getProgramas().subscribe((result) => {
    this.programa = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene los detalles del evento y los guarda en un array.

Ilustración 275. Frontend – Obtener información turística

```
getInformacionTuristica() {
  this._informacionTuristicaService
    .getInformacionTuristicas()
    .subscribe((result) => {
      this.informacionTuristica = result;
    });
}
```

Elaborado por: El investigador

El presente método obtiene la información del evento y la guarda en su respectivo array.

Ilustración 276. Frontend – Obtener información sobre el evento

```
getSobreElEvento() {
  //Presentacion
  this._presentacionService.getPresentaciones().subscribe((result) => {
    this.presentacion = result;
  });
  //Lugar del Luga del evento
  this._lugarDelEventoService.getLugarDelEvento().subscribe((result) => {
    this.lugarDelEvento = result;
  });
  //Galeria del lugar
  this._galeriaLugarService.getGaleria().subscribe((result) => {
    this.galeriaLugar = result;
  });
  //Galeria informativa
  this._galeriaInformacionService.getGaleria().subscribe((result) => {
    this.galeriaInformacion = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene la información del comité y la guarda en un array.

Ilustración 277. Frontend – Obtener comité

```
getComite() {
  this._comiteService.getComite().subscribe((result) => {
    this.comiteOrganizador = result.filter(
      (comite) => comite.tipo == "COMITÉ ORGANIZADOR"
    );
    this.comiteCientifico = result.filter(
      (comite) => comite.tipo != "COMITÉ ORGANIZADOR"
    );
  });
}
```

Elaborado por: El investigador

El presente método obtiene la información de las ediciones anteriores y la guarda en un array.

Ilustración 278. Frontend – Obtener ediciones

```
getEdiciones() {
  this._edicionService.getEdiciones().subscribe((result) => {
    this.ediciones = result;
  });
}
```

Elaborado por: El investigador

El presente método obtiene la información de los investigadores y la guarda en un array.

Ilustración 279. Frontend – Obtener investigadores

```
getInvestigadores() {
  this._investigadoresService.getInvestigadores().subscribe((result) => {
    this.investigadores = result;
  });
}
```

Elaborado por: El investigador

El presente método envía un correo mediante Email para consultas sobre el congreso.

Ilustración 280. Frontend – Enviar email de contacto

```
public sendEmail(e: Event) {
  e.preventDefault();
  emailjs
    .sendForm(
      "service_tog3xds",
      "template_7jqkydj",
      e.target as HTMLFormElement,
      "yeeju8ULL7f1DQq3x"
    )
    .then(
      (result: EmailJSResponseStatus) => {
        this.resetContacto();
      },
      (error) => {
      }
    );
}
```

Elaborado por: El investigador

El presente método muestra un mensaje de confirmación de que se envió el respectivo email.

Ilustración 281. Frontend – Confirmar envío de email

```
confirmacionEmail() {
  Swal.fire({
    position: "center",
    icon: "success",
    title: "Correo enviado",
    text: "Pronto nos pondremos en contacto contigo",
    showConfirmButton: false,
    timer: 1500,
  });
}
```

Elaborado por: El investigador

Página de registro

El presente método permite cambiar el expositor del paper.

Ilustración 282. Frontend – Cambiar expositor

```
cambiarEstadoExpositor(autores: any, index: number) {  
  const val = autores[index].expositor;  
  autores.forEach((element) => {  
    element.expositor = false;  
  });  
  autores[index].expositor = !val;  
}
```

Elaborado por: El investigador

El presente método verifica si la página de registro se encuentra habilitada.

Ilustración 283. Frontend – Habilitar pantalla de registro

```
habiliar() {  
  this.configService.getConf().subscribe((result) => {  
    result.forEach((element) => {  
      if (element.nombre === "Registro") {  
        if (!element.estado) {  
          this.router.navigate(["/acerca-de"]);  
        }  
      }  
    });  
  });  
}
```

Elaborado por: El investigador

El presente método permite cargar los tipos de inscripción habilitadas.

Ilustración 284. Frontend – Cargar tipo de inscripción

```
async CargarTipoInscripcion() {  
  await this.tipoInscripcionService  
    .getTipoInscripciones()  
    .subscribe((result) => {  
      this.inscripciones = result.filter((element) => element.estado);  
    });  
}
```

Elaborado por: El investigador

El presente método permite cargar los tipos de cargos que aceptara la aplicación.

Ilustración 285. Frontend – Cargar tipos de pagos

```
    async CargarPagos() {  
        await this.pagoService.getPagos().subscribe((result) => {  
            this.pagos = result;  
        });  
    }
```

Elaborado por: El investigador

El presente método permite cargar los tipos de idiomas en los que se enviara el correo de confirmación.

Ilustración 286. Frontend – Cargar idiomas

```
    async CargarIdiomas() {  
        await this.idiomaService.getIdiomas().subscribe((result) => {  
            this.idiomas = result;  
        });  
    }
```

Elaborado por: El investigador

El presente método permite cargar los tipos de papers que aceptara el sistema para su posterior ingreso.

Ilustración 287. Frontend – Cargar tipos de paper

```
    async cargarPapers() {  
        await this.tipoPaperService.getPapers().subscribe((result) => {  
            this.papers = result.filter((element) => element.estado);  
        });  
    }
```

Elaborado por: El investigador

El presente método permite cargar los diferentes paralelos que aceptara el sistema.

Ilustración 288. Frontend – Cargar paralelos

```
    async CargarParalelos() {  
        await this.paraleloService.getParalelos().subscribe((result) => {  
            this.paralelos = result;  
        });  
    }
```

Elaborado por: El investigador

El presente método permite cargar los diferentes semestres que aceptara el sistema.

Ilustración 289. Frontend – Cargar semestres

```
async CargarSemestres() {  
    await this.semestreService.getSemestres().subscribe((result) => {  
        this.semestres = result.filter((element) => element.estado);  
    });  
}
```

Elaborado por: El investigador

El presente método permite cargar las diferentes carreras que aceptara el sistema.

Ilustración 290. Frontend – Cargar carreras

```
async CargarCarreras() {  
    await this.carreraService.getCarreras().subscribe((result) => {  
        this.carreras = result.filter((element) => element.estado);  
    });  
}
```

Elaborado por: El investigador

El presente método permite cargar los tipos de identificaciones que aceptara el sistema.

Ilustración 291. Frontend – Cargar tipo de identificación

```
async CargarIdentificacion() {  
    await this.identificacionService  
        .getIdenticaciones()  
        .subscribe((result) => {  
            this.identificaciones = result;  
        });  
}
```

Elaborado por: El investigador

El presente método permitirá cambiar el paralelo del inscrito.

Ilustración 292. Frontend – Cambiar tipo de identificación

```
CambiarParalelo(val: Paralelo): void {  
    this.tipoParalelo = val.nombre;  
    this.idParalelo = val.id;  
}
```

Elaborado por: El investigador

El presente método permitirá cambiar el tipo de documento de identidad del inscrito.

Ilustración 293. Frontend – Cambiar tipo de documento de identidad

```
CambiarDocumento(val: Identificacion): void {  
    this.tipoIdentificacion = val.nombre;  
    this.idtipoDocumentoIdentidad = val.id;  
}
```

Elaborado por: El investigador

El presente método permitirá cambiar el tipo de semestre del inscrito.

Ilustración 294. Frontend – Cambiar semestres

```
CambiarSemestre(val: Semestre): void {  
    this.tipoSemestre = val.semestre;  
    this.idSemestre = val.id;  
}
```

Elaborado por: El investigador

El presente método permitirá cambiar la carrera del inscrito.

Ilustración 295. Frontend – Cambiar carrera

```
CambiarCarrera(val: Carrera): void {  
    this.tipoCarrera = val.carrera;  
    this.idCarrera = val.id;  
}
```

Elaborado por: El investigador

El presente método permitirá cambiar el tipo de lenguaje de la aplicación.

Ilustración 296. Frontend – Confirmar lenguaje del correo

```
CambiarLenguaje(val: Idioma): void {  
    this.tipoIdioma = val.idioma;  
    this.idLenguaje = val.id;  
}
```

Elaborado por: El investigador

El presente método permitirá cambiar el tipo de inscripción del participante.

Ilustración 297. Frontend – Cambiar tipo de inscripción

```
CambiarInscripcion(val: TipoInscripcion) {
  this.idTipoInscripcion = val.id;
  this.tipoInscripcion = val.nom_inscr;
  if (this.tipoInscripcion === "STUDENT") {
    this.tipoEstudiante = true;
    this.tipoAutor = false;
    this.autores = [];
  } else if (
    this.tipoInscripcion === "AUTHOR" ||
    this.tipoInscripcion === "AUTHOR UTA"
  ) {
    this.tipoEstudiante = false;
    this.tipoAutor = true;
    if (this.arrayPapers.length === 0) {
      this.agregarPaper();
    }
  } else {
    this.tipoEstudiante = false;
    this.tipoAutor = false;
    this.arrayPapers = [];
  }
  this.inscripciones.forEach((element) => {
    if (element.nom_inscr === this.tipoInscripcion) {
      this.costo = element.costo;
      this.precioFinal = this.costo;
    }
  });
}
```

Elaborado por: El investigador

El presente método permitirá navegar entre los formularios del sistema.

Ilustración 298. Frontend – Cambiar de formulario

```
cambiarForm(form: string) {
  if (form === "registroID") {
    this.registroID = true;
    this.registroForm = false;
    this.registroPago = false;
    this.RestablecerDatos();
  } else if (form === "registroForm") {
    if (this.documentoIdentidad.length !== 0) {
      this.CargarDatos();
      this.registroID = false;
      this.registroForm = true;
      this.registroPago = false;
    } else {
      this.cedulaImcompleta = true;
      setTimeout(() => {
        this.cedulaImcompleta = false;
      }, 3000);
    }
  } else if (form === "registroPago") {
    this.registroID = false;
    this.registroForm = false;
    this.registroPago = true;
  }
}
```

Elaborado por: El investigador

El presente método permitirá validar que todos los datos hayan sido ingresados de forma correcta.

Ilustración 299. Frontend – Verificar los datos del registro

```
verificarDatosRegistro(): boolean {
  this.validarCedula();
  this.validarEmail();
  if (
    this.idtipoDocumentoIdentidad != null &&
    this.documentoIdentidad != "" &&
    this.nombres != "" &&
    this.apellidos != "" &&
    this.direccion != "" &&
    this.email != "" &&
    this.telefono != "" &&
    this.idLenguaje != null &&
    this.mailValido
  ) {
    if (this.idTipoInscripcion === 1 || this.idTipoInscripcion === 4) {
      this.idCarrera = null;
      this.idParalelo = null;
      this.idSemestre = null;
      return true;
    } else if (this.idTipoInscripcion === 3) {
      if (
        this.idCarrera != null &&
        this.idParalelo != null &&
        this.idSemestre != null
      ) {
        this.idPaper = null;
        this.idTipoPaper = null;
        return true;
      }
    } else if (this.idTipoInscripcion === 2) {
      this.idCarrera = null;
      this.idParalelo = null;
      this.idSemestre = null;
      this.idPaper = null;
      this.idTipoPaper = null;
      return true;
    }
  }
  return false;
}
```

Elaborado por: El investigador

El presente método permite eliminar un autor del array de autores.

Ilustración 300. Frontend – Eliminar autor del array

```
eliminarAutor(paper: any, index: number) {
  if (
    paper.autores[index].id != null &&
    paper.autores[index].id != undefined
  ) {
    this.autoresEliminar.push(paper.autores[index].id);
  }

  paper.autores.splice(index, 1);
}
```

Elaborado por: El investigador

El presente método permite cargar los datos de los participantes que ya se encontraban registrados anteriormente.

Ilustración 301. Frontend – Cargar datos de los inscritos 1

```
async CargarDatos() {
  this.InscripcionExiste = false;
  this.datosCargados = false;
  return await this.personaService
    .getPersona(this.documentoIdentidad)
    .subscribe(
      (result) => {
        this.persona = result;
        this.InscripcionService.getInscripcion(result.id).subscribe(
          (result) => {
            this.tipoIdioma = this.idiomas[result.id_idioma - 1].idioma;
            this.idLenguaje = result.id_idioma;
            //
            this.idInscripcion = result.id;
            if (
              this.persona.id_tip_ins_pert === 1 ||
              this.persona.id_tip_ins_pert === 4
            ) {
              this.paperService
                .getPaper(this.idInscripcion)
                .subscribe((result) => {
                  result.forEach((paper) => {
                    this.autoresService
                      .getAutoresPaper(paper.id)
                      .subscribe((result) => {
                        let tipoPaper = "";
                        this.papers.forEach((element) => {
                          if (element.id === paper.id_tippaper) {
                            tipoPaper = element.tipo;
                          }
                        });
                        this.arrayPapers.push({
                          id: paper.id,
                          tipoPaper: tipoPaper,
                          idTipoPaper: paper.id_tippaper,
                          idPaper: paper.id_paper,
                          titulo: paper.titulo,
                          autores: result,
                        });
                      });
                    });
                  });
                });
            this.InscripcionExistente();
          },
          (err) => {
            this.datosCargados = true;
          }
        );
      }
    );
};
```

Elaborado por: El investigador

Ilustración 302. Frontend – Cargar datos de los inscritos 2

```
if (this.persona.id != undefined) {
  this.idPersona = this.persona.id;
  this.documentoIdentidad = this.persona.documento_identificacion;
  this.personaExiste = this.persona.documento_identificacion;
  this.nombres =
    this.persona.nom_paterno +
    " " +
    (this.persona.nom_materno != null
      ? this.persona.nom_materno
      : "");
  this.apellidos =
    this.persona.ape_paterno +
    " " +
    (this.persona.ape_materno != null
      ? this.persona.ape_materno
      : "");
  this.direccion =
    this.persona.direccion != null ? this.persona.direccion : "";
  this.email = this.persona.email != null ? this.persona.email : "";
  this.telefono =
    this.persona.celular != null ? this.persona.celular : "";

  this.tipoInscripcion =
    this.inscripciones[this.persona.id_tip_ins_pert - 1].nom_inscr;
  this.idTipoInscripcion = this.persona.id_tip_ins_pert;

  if (this.persona.id_tip_ins_pert == 3) {
  }
  if (
    this.persona.id_tip_ins_pert === 1 ||
    this.persona.id_tip_ins_pert === 4
  ) {
  }
  if (this.identificaciones[0].id === this.persona.id_tip_pert) {
    this.tipoIdentificacion = this.identificaciones[0].nombre;
    this.idTipoDocumentoIdentidad = this.identificaciones[0].id;
  } else {
    this.tipoIdentificacion = this.identificaciones[1].nombre;
    this.idTipoDocumentoIdentidad = this.identificaciones[1].id;
  }
  this.inscripciones.forEach((element) => {
    if (element.nom_inscr === this.tipoInscripcion) {
      this.costo = element.costo;
    }
  });
};
```

Elaborado por: El investigador

El presente método permite verificar si el participante ya se encontraba inscrito anteriormente en el congreso.

Ilustración 303. Frontend – Verificar si el participante ya se encuentra inscrito

```
inscripcionExistente() {
  Swal.fire({
    title:
      "Este usuario ya se encuentra registrado / This user is already registered",
    text: "¿Deseas editar los datos? / Do you want to edit the data?",
    icon: "info",
    iconColor: "#7A1E19",
    color: "#7A1E19",
    showCancelButton: true,
    confirmButtonColor: "#7A1E19",
    cancelButtonColor: "#85929E",
    confirmButtonText: "SI",
    cancelButtonText: "NO",
  }).then((result) => {
    if (result.value) {
      this.inscripcionExiste = true;
    } else {
      this.cambiarForm("registroID");
      this.RestablecerDatos();
    }
  });
}
```

Elaborado por: El investigador

El presente método realizara la validación de la cedula.

Ilustración 304. Frontend – Validar cedula

```
validarCedula() {
  if (this.documentoIdentidad.length !== 10) {
    const digitoRegion = this.documentoIdentidad.substring(0, 2);
    if (digitoRegion >= String(1) && digitoRegion <= String(24)) {
      const ultimoDigito = Number(this.documentoIdentidad.substring(9, 10));
      const pares =
        Number(this.documentoIdentidad.substring(1, 2)) +
        Number(this.documentoIdentidad.substring(3, 4)) +
        Number(this.documentoIdentidad.substring(5, 6)) +
        Number(this.documentoIdentidad.substring(7, 8));
      let numeroUno = any = this.documentoIdentidad.substring(0, 1);
      numeroUno = numeroUno * 2;
      if (numeroUno > 9) {
        numeroUno = numeroUno - 9;
      }
      let numeroTres = any = this.documentoIdentidad.substring(2, 3);
      numeroTres = numeroTres * 2;
      if (numeroTres > 9) {
        numeroTres = numeroTres - 9;
      }
      let numeroCinco = any = this.documentoIdentidad.substring(4, 5);
      numeroCinco = numeroCinco * 2;
      if (numeroCinco > 9) {
        numeroCinco = numeroCinco - 9;
      }
      let numeroSiete = any = this.documentoIdentidad.substring(6, 7);
      numeroSiete = numeroSiete * 2;
      if (numeroSiete > 9) {
        numeroSiete = numeroSiete - 9;
      }
      let numeroNueve = any = this.documentoIdentidad.substring(8, 9);
      numeroNueve = numeroNueve * 2;
      if (numeroNueve > 9) {
        numeroNueve = numeroNueve - 9;
      }
      const impares =
        numeroUno + numeroTres + numeroCinco + numeroSiete + numeroNueve;
      const sumaTotal = pares + impares;
      const primerDigitoSuma = String(sumaTotal).substring(0, 1);
      const decena = (Number(primerDigitoSuma) + 1) * 10;
      let digitoValidador = decena - sumaTotal;
      if (digitoValidador === 10) {
        digitoValidador = 0;
      }
      if (digitoValidador === ultimoDigito) {
        this.cedulaCorrecta = true;
      } else {
        this.cedulaCorrecta = false;
      }
    } else {
      this.cedulaCorrecta = false;
    }
  } else {
    if (this.idtipoDocumentoIdentidad === 1) {
      this.cedulaCorrecta = false;
    } else {
      this.cedulaCorrecta = true;
    }
  }
}
```

Elaborado por: El investigador

El presente método realizara la validación del email.

Ilustración 305. Frontend – Validar email

```
validarEmail() {
  var EMAIL_REGEX =
    /^((([^\<>()\[\]\\\\.,;:\s@"]+)|(\^<>()\[\]\\\\.,;:\s@"]+))*@((\[[0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\. [0-9]{1,3}\)|((\a-zA-Z\d-0-9)+\.)+[a-zA-Z]{2,}))$/;
  if (this.email.match(EMAIL_REGEX)) {
    this.mailValido = true;
  } else {
    this.mailValido = false;
  }
}
```

Elaborado por: El investigador

El presente método guardara los datos de la persona siempre y cuando haya cumplido con todas las validaciones.

Ilustración 306. Frontend – Guardar persona

```
async guardarPersona() {
  if (this.idTipoInscripcion === 1 || this.idTipoInscripcion === 4) {
    this.precioFinal = this.arrayPapers.length * this.costo;
  } else {
    this.precioFinal = this.costo;
  }

  if (this.idTipoInscripcion === 1 || this.idTipoInscripcion === 4) {
    let cont = 0;
    this.arrayPapers.forEach((paper) => {
      if (
        paper.titulo === "" ||
        paper.id_tipopaper === "" ||
        paper.idPaper === "" ||
        paper.autores === 0
      ) {
        this.mostrarToas = true;
      } else {
        this.mostrarToas = false;
      }

      paper.autores.forEach((autor) => {
        if (autor.Nombre1 !== "" && autor.Apellido1 !== "") {
          if (autor.expositor) {
            cont++;
          }
        }
      });
    });
    if (this.mostrarToas) {
      return;
    }

    if (cont < this.arrayPapers.length) {
      this.MensajeInfo(
        "Advertencia / Warning",
        "Se debe seleccionar al menos a un expositor / At least one exhibitor must be selected."
      );
      return;
    }
  }
}
```

Elaborado por: El investigador

El presente método guardara la inscripción una vez halla validado correctamente todos los anteriores datos.

Ilustración 307. Frontend – Guardar inscripción

```
async guardarInscripcion() {
  var interval = setInterval(function () {
    if (this.idPersona !== "") {
      clearInterval(interval);
    }
  }, 2000);
  const fecha = new Date();

  this.inscripcionGuardar = {
    id_per_pert: this.idPersona,
    id_tipopago: this.idTipoPago,
    cod_pago: null,
    resultado_autorizacion: null,
    codigo_autorizacion: null,
    foto_deposito: null,
    fecha_inscripcion:
      fecha.getFullYear() + "-" + fecha.getMonth() + "-" + fecha.getDay(),
    estado: "NV",
    num_deposito: null,
    id_idioma: this.idlenguaje,
    fecha_Registro_de_validacion: null,
    Hora_Registro_Deposito: null,
    Hora_Validacion_Deposito: null,
    fecha_validacion_deposito: null,
    monto: String(this.precioFinal),
    Hora_Inscripcion:
      fecha.getHours() + ":" + fecha.getMinutes() + ":" + fecha.getSeconds(),
  };

  await this.inscripcionService
    .insertInscripcion(this.inscripcionGuardar)
    .subscribe((result) => {
      this.idInscripcion = result[0].id;

      if (this.idTipoInscripcion === 1 || this.idTipoInscripcion === 4) {
        for (let i = 0; i < this.arrayPapers.length; i++) {
          this.paperGuardar = {
            id_inscripcion: this.idInscripcion,
            id_paper: this.arrayPapers[i].idPaper.toString(),
          };
        }
      }
    });
}
```

Elaborado por: El investigador

Página de validación

El presente método verifica si este habilitado o no el módulo de validación.

Ilustración 308. Frontend – Habilitar página de validación

```
habiliar() {
  this.configService.getConf().subscribe((result) =>
    result.forEach((element) => {
      if (element.nombre === "Validación") {
        if (!element.estado) {
          this.router.navigate(["/acerca-de"]);
        }
      }
    });
  });
}
```

Elaborado por: El investigador

El presente método permite realizar la búsqueda de las personas inscritas según su cedula.

Ilustración 309. Frontend – Buscar inscripción

```
async buscarInscripcion() {
  if (this.documentoIdentidad.length != 0) {
    await this.personaService
      .getPersona(this.documentoIdentidad)
      .subscribe((result) => {
        this.nombre = result.nom_paterno + " " + result.ape_paterno;
        this.email = result.email;
        this.inscripcionService
          .getInscripcion(result.id)
          .subscribe((result) => {
            this.idPersona = result.id_per_pert;
            this.idioma=result.idioma;
            if (result.num_deposito != null && result.foto_deposito != null) {
              this.inscripcionExistente();
              this.comprobanteDeposito = result.num_deposito;
            } else {
              this.cambiarEncontrado();
            }
          });
        });
    setTimeout(() => {
      if (this.idPersona === null) {
        this.mensaje("Error!", "Usted no se encuentra registrado / You are not registered", "error");
      }
    }, 2000);
  }
}
```

Elaborado por: El investigador

El presente método presente un mensaje con los parámetros enviados.

Ilustración 310. Frontend – Mensaje general registro

```
mensaje(titulo: string, texto: string, icono: any) {
  Swal.fire({
    title: titulo,
    text: texto,
    icon: icono,
    confirmButtonText: "Aceptar",
    confirmButtonColor: "#7A1E19",
    iconColor: "#7A1E19",
    color: "#7A1E19",
  });
}
```

Elaborado por: El investigador

El presente método renombrara el archivo seleccionado con el número de cedula del inscrito.

Ilustración 311. Frontend – Renombrar archivo

```
renameFile(originalFile, newName) {
  return new File([originalFile], newName, {
    type: originalFile.type,
    lastModified: originalFile.lastModified,
  });
}
```

Elaborado por: El investigador

El presente método permitirá seleccionar un archivo jpg/png dese el ordenador o móvil para su posterior subida.

Ilustración 312. Frontend – Seleccionar imagen

```
onFileSelected(event) {
  this.file = this.renameFile(
    event.target.files[0],
    this.documentoIdentidad + ".jpg"
  );
  if (this.file.name != null && this.file.name != "") {
    if (this.file.size <= 2000000) {
      this.textoImg = this.file.name;
    } else {
      /* Mensaje archivo */
      this.Mensaje("Nota / Note:", "La imagen no puede superar los 2MB de tamaño. / The image cannot exceed 2MB in size.");
      this.file = null;
    }
  } else {
    this.textoImg = "Seleccionar archivo... / Select File...";
    this.file = null;
  }
}
```

Elaborado por: El investigador

El presente método se encarga de guardar los respectivos datos, los datos son enviados en un formulario de datos y no en un json.

Ilustración 313. Frontend – Enviar datos

```
enviarDatos() {
  if (this.file != null && this.comprobanteDeposito != "") {
    const fecha = new Date();
    const fechaValidacion =
      fecha.getFullYear() + "-" + fecha.getMonth() + "-" + fecha.getDay();

    var data = new FormData();
    data.append("num_deposito", this.comprobanteDeposito);
    data.append("foto_deposito", this.file);
    const HoraValidacion =
      fecha.getHours() + ":" + fecha.getMinutes() + ":" + fecha.getSeconds();
    data.append("Hora_Registro_Deposito", HoraValidacion);
    data.append("fecha_Registro_de_validacion", fechaValidacion);
    data.append("observacion", this.observacion);
    this.inscripcionService.updateComprobante(this.idPersona, data).subscribe(
      this.mensaje("Validación / Validation ", "Registro Validado / Validated Registration", "success"),
      this.mailerService
        .mailSubidaTiket(this.email, this.infoCongreso.nombre, this.idioma)
        .subscribe((result) => {}),
      this.router.navigate(["/acerca-de"])
    );
  } else {
    this.mostrarMensaje = true;
    setTimeout(() => {
      this.mostrarMensaje = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método verifica si el participante cuenta con un comprobante subido o no.

Ilustración 314. Frontend – Verificar si existe un comprobante subido

```
ComprobanteExistente() {
  Swal.fire({
    title: '¿Usted ya subió su comprobante, desea continuar? / You already uploaded your receipt, do you want to continue?',
    text: '¿Deseas editar los datos? / Do you want to edit the data?',
    icon: 'info',
    iconColor: '#7A1E19',
    color: '#7A1E19',
    showCancelButton: true,
    confirmButtonColor: '#7A1E19',
    cancelButtonColor: '#89299C',
    confirmButtonText: 'SI',
    cancelButtonText: 'NO',
  }).then((result) => {
    if (result.value) {
      console.log(result.value);
      this.cambiarEncontrado();
    } else {
      console.log(result.value);
      this.comprobanteDeposito = '';
    }
  });
}
```

Elaborado por: El investigador

Página de certificados

El presente método realiza la búsqueda de certificados de asistencia y participación según el número de cedula.

Ilustración 315. Frontend – Obtener inscripción

```
getInscripcionByCI() {
  this.inscripciones = [];
  this.papers = [];
  this._adminService.getInscripcionByCI(this.cedula).subscribe((result) => {
    this.inscripciones = result.filter(
      (result) =>
        result.certificado_I != null ||
        result.certificado_P != null ||
        result.certificado_Autor != null
    );
    if (this.inscripciones.length > 0) {
      this.getCertificadosAutor(this.inscripciones[0].numero);
      this.cambiarEncontrado();
    } else {
      this.Mensaje(
        "Estimado participante su certificado no se encuentra registrado al momento",
        "Dear participant, your certificate is not registered at the moment"
      );
    }
  });
}
```

Elaborado por: El investigador

El presente método realiza la búsqueda de los certificados de los autores.

Ilustración 316. Frontend – Obtener certificados de autores

```
getCertificadosAutor(id: number) {
  this._paperService.getPaper(String(id)).subscribe((result) => {
    this.papers = result.filter(
      (element) => element.certificado != null && element.certificado != ""
    );
    this.getCertificadoExpositor();
  });
}
```

Elaborado por: El investigador

El presente método realiza la búsqueda de los certificados de los expositores.

Ilustración 317. Frontend – Obtener certificados de expositores

```
getCertificadoExpositor() {
  this.papers.forEach((element) => {
    this._expositorService.getExpositor(element.id).subscribe((result) => {
      console.log(result)
      result.forEach((expositor)=>{
        if (expositor.certificado_A != null && expositor.certificado_A != "") {
          this.expositores.push(expositor);
        }
      })
    });
  });
}
```

Elaborado por: El investigador

Página de facturas

El presente método realiza la búsqueda de las facturas que el usuario tenga registrado.

Ilustración 318. Frontend – Obtener factura según cedula

```
getInscripcionByCI(){
  this._adminService.getInscripcionByCI(this.cedula).subscribe((result)->{
    this.inscripciones=result.filter(result => result.factura.length!=0);
    if(this.inscripciones.length>0){
      this.cambiarEncontrado();
    }else{
      this.Mensaje("Estimado participante su factura no se encuentra registrada al momento","Dear participant, your invoice is not registered at the moment")
    }
  })
}
```

Elaborado por: El investigador

El presente método permite visualizar la factura.

Ilustración 319. Frontend – Visualizar factura

```
verFactura( content,PDF) {
  this.documento=URL+PDF;

  this.modalRef = this._modalService.open(content, { size: "x1" });
  this.modalRef.result.then(
    (result) => {
      },
    (reason) => {
      this.closeResult = `Dismissed ${this.getDismissReason(reason)}`;
    }
  );
}
```

Elaborado por: El investigador

Página de administración

El presente método permite generar el reporte de los papers

Ilustración 320. Frontend – Generar reporte papers

```
getPaperReporte(){
  this.arrayPapersReportes=[];
  this._paperService.getPapers().subscribe((result)=>{
    result.forEach(paper => {
      this._autoresService
        .getAutoresPaper(paper.id)
        .subscribe((result) => {
          let tipoPaper = "";
          this.papers.forEach((element) => {
            if (element.id === paper.id_tipopaper) {
              tipoPaper = element.tipo;
            }
          });
          this.arrayPapersReportes.push({
            id: paper.id,
            tipoPaper: tipoPaper,
            idTipoPaper: paper.id_tipopaper,
            idPaper: paper.id_paper,
            titulo: paper.titulo,
            autores: result,
          });
          this.arrayPapersReportes.sort(((a, b) => a.idPaper - b.idPaper));
        });
    });
  });
}
```

Elaborado por: El investigador

El presente método permite filtrar los inscritos validados de los inscritos no validados

Ilustración 321. Frontend – Filtro inscritos

```
filtroRegistro() {
  if (this.tipoRegistro === "Registros no validados") {
    this.inscripcionesFiltradas = this.inscripciones.filter(
      (element) => element.estado === "NV"
    );
  } else if (this.tipoRegistro === "Registros Validados") {
    this.inscripcionesFiltradas = this.inscripciones.filter(
      (element) => element.estado === "V"
    );
  } else if (this.tipoRegistro === "Todos los registros") {
    this.inscripcionesFiltradas = this.inscripciones;
  }
}
```

Elaborado por: El investigador

El presente método permite filtrar los inscritos que han subido su comprobante de los que no lo han subido.

Ilustración 322. Frontend – Filtro comprobantes

```
filtroComprobante() {  
  if (this.tipoComprobante === "Comprobantes subidos") {  
    this.inscripcionesFiltradas = this.inscripcionesFiltradas.filter(  
      (element) => element.comprobante !== null  
    );  
  }  
  if (this.tipoComprobante === "Sin comprobante") {  
    this.inscripcionesFiltradas = this.inscripcionesFiltradas.filter(  
      (element) => element.comprobante === null  
    );  
  }  
}
```

Elaborado por: El investigador

El presente método permite filtrar a los inscritos según su forma de pago.

Ilustración 323. Frontend – Filtro tipo de pago

```
filtroPago() {  
  this.inscripcionesFiltradas = this.inscripcionesFiltradas.filter(  
    (element) => element.tipo_Pago === this.tipoPago  
  );  
}
```

Elaborado por: El investigador

El presente método permite filtrar a los inscritos según su carrera.

Ilustración 324. Frontend – Filtro tipo de carrera

```
filtroCarrera() {  
  this.filtroInscripcion();  
  this.inscripcionesFiltradas = this.inscripcionesFiltradas.filter(  
    (element) => element.id_carrera_per === this.idCarrera  
  );  
}
```

Elaborado por: El investigador

El presente método permite filtrar a los inscritos según su semestre.

Ilustración 325. Frontend – Filtro tipo de semestre

```
filtroSemestre() {
  this.inscripcionesFiltradas = this.inscripcionesFiltradas.filter(
    (element) => element.id_semestre_per === this.idSemestre
  );
}
```

Elaborado por: El investigador

El presente método permite filtrar a los inscritos según su paralelo.

Ilustración 326. Frontend – Filtro tipo de paralelo

```
filtroParalelo() {
  this.inscripcionesFiltradas = this.inscripcionesFiltradas.filter(
    (element) => element.id_paralelo_per === this.idparalelo
  );
}
```

Elaborado por: El investigador

El presente método permite cambiar el estado de los inscritos de validados y no validados.

Ilustración 327. Frontend – Cambiar estado de los inscritos

```
cambiarEstado(inscrip: AdminsInscrip) {
  const fecha = new Date();
  if (inscrip.estado !== "V") {
    const estado = {
      estado: "V",
      fecha_validacion_deposito: fecha.getFullYear() + "-" + fecha.getMonth() + "-" + fecha.getDay(),
      Hora_Validacion_Deposito:
        fecha.getHours() +
        ":" +
        fecha.getMinutes() +
        ":" +
        fecha.getSeconds(),
    };
    inscrip.estado = "V";
    this._adminInscripService
      .update(inscrip.numero.toString(), estado)
      .subscribe((result) => {
        this.getInscripcionesValidadas();

        const mailVerificacion = {
          destinatario: inscrip.correo,
          edicion: this.informacionCongreso[0].nombre,
          idioma: inscrip.idioma
        };
        this._mailerService
          .mailVerificacion(mailVerificacion)
          .subscribe((result) => {});
      });
  } else if (inscrip.estado === "V") {
    inscrip.estado = "NV";
    const estado = {
      estado: "NV",
      fecha_validacion_deposito: fecha.getFullYear() + "-" + fecha.getMonth() + "-" + fecha.getDay(),
      Hora_Validacion_Deposito:
        fecha.getHours() +
        ":" +
        fecha.getMinutes() +
        ":" +
        fecha.getSeconds(),
    };
    this._adminInscripService
      .update(inscrip.numero.toString(), estado)
      .subscribe((result) => {
        this.getInscripcionesValidadas();
        this.getAutores();
      });
  }
}
```

Elaborado por: El investigador

El presente método permite seleccionar un archivo desde el ordenador o móvil.

Ilustración 328. Frontend – Seleccionar archivo

```
onFileSelected(event) {
  this.file = event.target.files[0];
  if (this.file.name !== null && this.file.name !== "") {
    this.textoPdf = this.file.name;
  } else {
    this.textoPdf = "";
    this.file = null;
  }
}
```

Elaborado por: El investigador

El presente método permite subir la factura de cada inscrito al sistema.

Ilustración 329. Frontend – Guardar factura

```
guardarPdf() {
  var data = new FormData();
  data.append("factura", this.file);

  this.inscripcionService
    .updateFactura(this.id_per_pert, data)
    .subscribe((result) => {
      (this.id_per_pert = null),
      this.Mensaje(
        "Factura actualziada",
        "Los datos se actualizaron exitosamente!"
      ),
      this.getInscripciones();
      this.cerrarModal();
    });
}
```

Elaborado por: El investigador

El presente método permite visualizar el modal para visualizar o subir las facturas.

Ilustración 330. Frontend – Abrir modal factura

```
openFactura(inscripcion: AdminsInscrip, content) {
  this.inscripcionActual = inscripcion;
  this.id_per_pert = inscripcion.id_per_pert;
  if (this.inscripcionActual.factura !== "" && this.inscripcionActual.factura !== null) {
    this.linkUrl(inscripcion.factura);
  }
  this.modalRef = this._modalService.open(content, { size: "xl" });
  this.modalRef.result.then(
    (result) => {
      this.resetVariables();
    },
    (reason) => {
      this.closeResult = `Dismissed ${this.getDismissReason(reason)}`;
      this.resetVariables();
    }
  );
}
```


Elaborado por: El investigador

El presente método permite subir las facturas de los inscritos por lotes.

Ilustración 331. Frontend – Subir factura por lotes

```
subirPdfPorLotes() {
  for (let factura of this.facturasPorLotes) {
    for (let inscripcion of this.inscripciones) {
      if (factura.name.split(".").slice(0, -1).join(".") === inscripcion.di) {
        var data = new FormData();
        data.append("factura", factura);
        this._inscripcionService
          .updateFactura(inscripcion.id_per_pert.toString(), data)
          .subscribe((result) => {});
      }
    }
  }
  this.facturasPorLotes = [];
  this.Mensaje("Operación Exitosa", "Se subieron todas las facturas");
  this.getInscripciones();
}
```

Elaborado por: El investigador

El presente método permite guardar a los patrocinadores del evento.

Ilustración 332. Frontend – Guardar patrocinadores

```
guardarPatrocinador() {
  if (
    this.nombrePatrocinador != "" &&
    this.file != undefined &&
    this.nombrePatrocinador != "" &&
    this.linkPatrocinador != ""
  ) {
    var data = new FormData();
    data.append("nombre", this.nombrePatrocinador);
    data.append("imagen", this.file);
    data.append("estado", this.estadoPatrocinador ? String(1) : String(0));
    data.append("link", this.linkPatrocinador);

    this._patrocinadoresService
      .insertPatrocinador(data)
      .subscribe((result) => {
        this.cerrarModal();
      });
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite actualizar el estado del patrocinador da activo o inactivo.

Ilustración 333. Frontend – Actualizar estado del patrocinador

```
actualizarEstadoPatrocinador(patrocinador: Patrocinador) {
  patrocinador.estado = !patrocinador.estado;
  this._patrocinadoresService
    .editPatrocinador(patrocinador.id, patrocinador)
    .subscribe((result) => {});
}
```

Elaborado por: El investigador

El presente método permite editar los datos de los patrocinadores.

Ilustración 334. Frontend – Editar datos patrocinadores

```
    cambiarDatosPatrocinador() {  
      if (  
        this.nombrePatrocinador !== "" &&  
        this.nombrePatrocinador !== "" &&  
        this.linkPatrocinador !== ""  
      ) {  
        var data = new FormData();  
  
        data.append("nombre", this.nombrePatrocinador);  
        this.file !== undefined ? data.append("imagen", this.file) : "";  
        data.append("estado", this.estadoPatrocinador ? String(1) : String(0));  
        data.append("link", this.linkPatrocinador);  
  
        this._patrocinadoresService  
          .editPatrocinador(this.idPatrocinador, data)  
          .subscribe((result) => {  
            this.cerrarModal();  
          });  
      } else {  
        this.error = true;  
        setTimeout(() => {  
          this.error = false;  
        }, 5000);  
      }  
    }  
  }  
}
```

Elaborado por: El investigador

El presente método permite guardar la información del registro.

Ilustración 335. Frontend – Guardar información del registro

```
    guardarInfoRegistro() {  
      if (this.file !== undefined) {  
        var data = new FormData();  
        data.append("imagen", this.file);  
        data.append("estado", this.estadoInfoRegistro ? String(1) : String(0));  
  
        this._infoRegistroService.insertInfoRegistro(data).subscribe((result) => {  
          this.cerrarModal();  
          this.getInfoRegistro();  
        });  
      } else {  
        this.error = true;  
        setTimeout(() => {  
          this.error = false;  
        }, 5000);  
      }  
    }  
  }  
}
```

Elaborado por: El investigador

El presente método permite actualizar el estado de la información del registro.

Ilustración 336. Frontend – Actualizar estado registro

```
    actualizarEstadoInfoRegistro(item: InfoRegistro) {  
      item.estado = !item.estado;  
      this._infoRegistroService  
        .editInfoRegistro(item.id, item)  
        .subscribe((result) => {});  
    }  
  }  
}
```

Elaborado por: El investigador

El presente método permite editar los datos de la información del registro

Ilustración 337. Frontend – Editar datos del registro

```
cambiarDatosInfoRegistro() {  
  var data = new FormData();  
  
  this.file != undefined ? data.append("imagen", this.file) : "";  
  data.append("estado", this.estadoInfoRegistro ? String(1) : String(0));  
  
  this._infoRegistroService  
    .editInfoRegistro(this.idInfoRegistro, data)  
    .subscribe((result) => {  
      this.cerrarModal();  
      this.getInfoRegistro();  
    });  
}
```

Elaborado por: El investigador

El presente método permite guardar las ediciones en el sistema previo las validaciones correspondientes.

Ilustración 338. Frontend – Guardar edición

```
guardarEdicion() {  
  if (  
    this.tituloEdicion != "" &&  
    this.lugarEdicion != "" &&  
    this.modalidadEdicion != "" &&  
    this.horasEdicion != "" &&  
    this.fechaEdicion != "" &&  
    this.telefonoEdicion != "" &&  
    this.correoEdicion != "" &&  
    this.anioEdicion != ""  
  ) {  
    var data = new FormData();  
    data.append("titulo", this.tituloEdicion);  
    data.append("lugar", this.lugarEdicion);  
    data.append("modalidad", this.modalidadEdicion);  
    data.append("horas", this.horasEdicion);  
    data.append("fecha", this.fechaEdicion);  
    data.append("telefono", this.telefonoEdicion);  
    data.append("correo", this.correoEdicion);  
    data.append("web", this.webEdicion);  
    data.append("edicion", this.anioEdicion);  
  
    if (  
      this.file != undefined &&  
      this.file2 != undefined &&  
      this.idEdicion === null  
    ) {  
      data.append("imagen", this.file);  
      data.append("pdf", this.file2);  
      this._edicionService.insertEdicion(data).subscribe((result) => {  
        this.getEdiciones();  
        this.cerrarModal();  
      });  
    } else {  
      this.file != undefined && this.file != null  
        ? data.append("imagen", this.file)  
        : "";  
  
      this.file2 != undefined && this.file2 != null  
        ? data.append("pdf", this.file2)  
        : "";  
  
      this._edicionService  
        .editEdicion(this.idEdicion, data)  
        .subscribe((result) => {  
          this.getEdiciones();  
          this.cerrarModal();  
        });  
    }  
  } else {  
    this.error = true;  
    setTimeout(() => {  
      this.error = false;  
    }, 5000);  
  }  
}
```

Elaborado por: El investigador

El presente método permite guardar el nuevo tipo de paper previo las validaciones correspondientes.

Ilustración 339. Frontend – Guardar paper

```
guardarPaper() {
  if (this.nombrePaper !== "") {
    const paperGuardar: Paper = {
      tipo: this.nombrePaper,
      estado: this.estadoPaper,
    };
    if (this.idPaper !== null) {
      paperGuardar.id = this.idPaper;
      this._tipoPaperService.editPaper(paperGuardar).subscribe((result) => {
        this.getTipoPaper();
        this.cerrarModal();
      });
    } else {
      this._tipoPaperService.insertPaper(paperGuardar).subscribe((result) => {
        this.getTipoPaper();
        this.cerrarModal();
      });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite cambiar el estado del semestre de activo o inactivo.

Ilustración 340. Frontend – Actualizar estado del semestre

```
actualizarEstadoSemestre(item: Semestre) {
  item.estado = !item.estado;
  this._semestreService.editSemestre(item).subscribe();
}
```

Elaborado por: El investigador

El presente método permite cambiar el estado de la carrera de activa o inactiva.

Ilustración 341. Frontend – Actualizar estado de la carrera

```
actualizarEstadoCarrera(item: Carrera) {
  item.estado = !item.estado;
  this._carreraService.editCarrera(item).subscribe();
}
```

Elaborado por: El investigador

El presente método permite cambiar el estado de los tipos de inscripción de activo o inactivo.

Ilustración 342. Frontend – Actualizar estado tipo de inscripción

```
actualizarEstadoInscripcion(item: TipoInscripcion) {
  item.estado = !item.estado;
  this._tipoInscripcionService.editTipoInscripcion(item).subscribe();
}
```

Elaborado por: El investigador

El presente método permite guardar un nuevo tipo de inscripción o actualizarlo según corresponda.

Ilustración 343. Frontend – Guardar tipo de inscripción

```
guardarTipoInscripcion() {
  if ((this.nombreTipoInscripcion != "" , this.precioTipoInscripcion != "")) {
    const inscripcionGuardar: TipoInscripcion = {
      nom_inscr: this.nombreTipoInscripcion,
      costo: Number(this.precioTipoInscripcion),
      estado: this.estadoTipoInscripcion,
    };
    if (this.idTipoInscripcion === null) {
      this._tipoInscripcionService
        .insertTipoInscripcion(inscripcionGuardar)
        .subscribe((result) => {
          this.CargarTipoInscripcion();
          this.cerrarModal();
        });
    } else {
      inscripcionGuardar.id = Number(this.idTipoInscripcion);
      this._tipoInscripcionService
        .editTipoInscripcion(inscripcionGuardar)
        .subscribe((result) => {
          this.CargarTipoInscripcion();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite cambiar el estado de los idiomas.

Ilustración 344. Frontend – Actualizar estado del idioma

```
actualizarEstadoIdioma(item: Idioma) {
  item.estado = !item.estado;
  this._idiomaService.editIdioma(item).subscribe();
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar el idioma según corresponda.

Ilustración 345. Frontend – Guardar estado del idioma

```
guardarIdioma() {
  if (this.nombreIdioma != "") {
    const idiomaGuardar: Idioma = {
      idioma: this.nombreIdioma,
      estado: this.estadoIdioma,
    };
    if (this.idIdioma === null) {
      this._idiomaService.insertIdioma(idiomaGuardar).subscribe((result) => {
        this.getIdiomas();
        this.cerrarModal();
      });
    } else {
      idiomaGuardar.id = this.idIdioma;
      this._idiomaService.editIdioma(idiomaGuardar).subscribe((result) => {
        this.getIdiomas();
        this.cerrarModal();
      });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar las carreras según corresponda.

Ilustración 346. Frontend – Guardar carrera

```
guardarCarrera() {
  if (this.nombreCarrera != "") {
    const carreraGuardar: Carrera = {
      carrera: this.nombreCarrera,
      estado: this.estadoCarrera,
    };
    if (this.idCarreraActual === null) {
      this._carreraService
        .insertCarrera(carreraGuardar)
        .subscribe((result) => {
          this.CargarCarreras();
          this.cerrarModal();
        });
    } else {
      carreraGuardar.id = this.idCarreraActual;
      this._carreraService.editCarrera(carreraGuardar).subscribe((result) => {
        this.CargarCarreras();
        this.cerrarModal();
      });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar el comité según corresponda.

Ilustración 347. Frontend – Guardar comité

```
guardarComite() {
  if (
    this.idTipoComite != null &&
    this.nombreComite != "" &&
    this.InstitutoComite != "" &&
    this.PaisComite != ""
  ) {
    var fecha = new Date();
    var year = fecha.getFullYear();
    const comiteGuardar = {
      id: null,
      id_tipo_per: this.idTipoComite,
      nombre: this.nombreComite,
      cargo: this.CargoComite,
      instituto: this.InstitutoComite,
      pais: this.PaisComite,
      edicion: year,
      validacion: this.validarComite,
    };
    if (this.IdComite === null) {
      this._comiteService.insertComite(comiteGuardar).subscribe((result) => {
        this.getComite();
        this.cerrarModal();
      });
    } else {
      comiteGuardar.id = this.idComite;
      this._comiteService.editComite(comiteGuardar).subscribe((result) => {
        this.getComite();
        this.cerrarModal();
      });
    } else {
      this.error = true;
      setTimeout(() => {
        this.error = false;
      }, 5000);
    }
  }
}
```

Elaborado por: El investigador

El presente método permite modificar los datos de los inscritos según sea necesario.

Ilustración 348. Frontend – Modificar inscripción

```
guardarInscripcion() {
  if (
    this.apellidosInscripcion != "" &&
    this.nombresInscripcion != "" &&
    this.correoInscripcion != "" &&
    this.celularInscripcion != "" &&
    this.direccionInscripcion != ""
  ) {
    let divNombre = this.nombresInscripcion.split(" ");
    let divApellido = this.apellidosInscripcion.split(" ");
    const personaGuardar = {
      nom_paterno: divNombre[0],
      nom_materno: divNombre[1],
      ape_paterno: divApellido[0],
      ape_materno: divApellido[1],
      direccion: this.direccionInscripcion,
      email: this.correoInscripcion,
      celular: this.celularInscripcion,
      id_tip_ins_pert: this.tipoInscripcionID,
    };
    const inscripcionGuardar = {
      estado: this.estadoInscripcion ? "V" : "N",
      num_deposito: this.numeroDepositoInscripcion,
    };
    this._inscripcionService
      .editInscripcion(
        this.idPersonaInscripcion.toString(),
        inscripcionGuardar
      )
      .subscribe(() => {
        this._personaService
          .editPersona(this.idPersonaInscripcion.toString(), personaGuardar)
          .subscribe(() => {
            this.cerrarModal();
            this.getInscripciones();
          });
      });
    } else {
      this.errorInscripcion = true;
      setTimeout(() => {
        this.errorInscripcion = false;
      }, 5000);
    }
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar los investigadores según corresponda.

Ilustración 349. Frontend – Guardar investigador

```
guardarInvestigador() {
  if (
    this.nombreInvestigador != "" &&
    this.paisInvestigador != "" &&
    this.Edicion != "" &&
    this.institucionInvestigador != ""
  ) {
    var data = new FormData();
    data.append("nombre", this.nombreInvestigador);
    data.append("pais", this.paisInvestigador);
    data.append("Edicion", this.Edicion);
    data.append("institucion", this.institucionInvestigador);
    if (this.idInvestigador === null && this.file != undefined) {
      if (this.file != undefined) {
        data.append("foto", this.file);
      }
      if (this.file2 != undefined) {
        data.append("pdf", this.file2);
      }
      this._investigadoresService
        .insertInvestigador(data)
        .subscribe(
          () => this.getInvestigadores(),
          this.cerrarModal(),
          this.resetVariablesInvestigador()
        );
    } else if (this.idInvestigador != null) {
      if (this.file != undefined) {
        data.append("foto", this.file);
      }
      if (this.file2 != undefined) {
        data.append("pdf", this.file2);
      }
      this._investigadoresService
        .editInvestigador(this.idInvestigador, data)
        .subscribe(
          () => this.getInvestigadores(),
          this.cerrarModal(),
          this.resetVariablesInvestigador()
        );
    } else {
      this.error = true;
      setTimeout(() => {
        this.error = false;
      }, 5000);
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permitirá generar los reportes en formato pdf.

Ilustración 350. Frontend – Obtener pdf

```
getPDF() {
  var HTML_Width = $(".canvas_div_pdf").width();
  var HTML_Height = $(".canvas_div_pdf").height();
  var top_left_margin = 0;
  var PDF_Width = HTML_Width + top_left_margin * 2;
  var PDF_Height = PDF_Width * 1.5 + top_left_margin * 2;
  var canvas_image_width = HTML_Width;
  var canvas_image_height = HTML_Height;

  var totalPDFPages = Math.ceil(HTML_Height / PDF_Height) - 1;

  html2canvas($(".canvas_div_pdf")[0], { allowTaint: true }).then(function (
    canvas
  ) {
    canvas.getContext("2d");

    var imgData = canvas.toDataURL("image/jpeg", 1.0);
    var pdf = new jsPDF("p", "pt", [PDF_Width, PDF_Height]);
    pdf.addImage(
      imgData,
      "JPG",
      top_left_margin,
      top_left_margin,
      canvas_image_width,
      canvas_image_height
    );

    for (var i = 1; i <= totalPDFPages; i++) {
      pdf.addPage(PDF_Width, PDF_Height);
      pdf.addImage(
        imgData,
        "JPG",
        top_left_margin,
        -(PDF_Height * i) + top_left_margin * 4,
        canvas_image_width,
        canvas_image_height
      );
    }

    pdf.save("HTML-Document.pdf");
  });
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar la información del registro según corresponda

Ilustración 351. Frontend – Guardar información congreso

```
guardarInformacionRegistro() {
  if (
    this.nombreCongreso != "" &&
    this.tituloCongreso != "" &&
    this.fechaCongreso != ""
  ) {
    var data = new FormData();
    data.append("nombre", this.nombreCongreso);
    data.append("titulo", this.tituloCongreso);
    data.append("fecha", this.fechaCongreso);
    if (this.file != undefined) {
      data.append("logo", this.file);
    }
    if (this.file2 != undefined) {
      data.append("favicon", this.file2);
    }
    this._informacionCongresoService
      .editInfo(this.idCongreso, data)
      .subscribe((result) => {
        this.getInformacionCongreso();
        this.cerrarModal();
      });
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar las imágenes de la galería de la portada.

Ilustración 352. Frontend – Guardar imagen de portada

```
guardarImagenPortada() {
  var data = new FormData();
  data.append("link", this.linkImgPortada);
  data.append("estado", this.estadoImgPortada ? "1" : "0");
  if (this.idImgPortada === null) {
    if (this.file != undefined) {
      data.append("imagen", this.file);
      this._imagenesPortada.insertImagenPortada(data).subscribe((result) => {
        this.getImagenesPortada();
        this.cerrarModal();
      });
    } else {
      this.error = true;
      setTimeout(() => {
        this.error = false;
      }, 5000);
    }
  } else {
    if (this.file != undefined) {
      data.append("imagen", this.file);
    }
    this._imagenesPortada
      .editImagenPortada(this.idImgPortada, data)
      .subscribe((result) => {
        this.getImagenesPortada();
        this.cerrarModal();
      });
  }
}
```

Elaborado por: El investigador

El presente método permite contar los inscritos según su tipo de inscripción.

Ilustración 353. Frontend – Contar tipos de inscritos

```
contarInscritos() {
  this.inscripciones.forEach((element) => {
    if (element.tipo_inscrito === "STUDENT") {
      this.totalEstudiantes++;
      if (element.estado === "V") {
        this.totalEstudiantesValidados++;
        if (element.tipo_pago === "CASH") {
          this.pagoEfectivo++;
        } else {
          this.pagoTarjeta++;
        }
      }
    } else if (element.tipo_inscrito === "PROFESSIONAL") {
      this.totalProfesionales++;
      if (element.estado === "V") {
        this.totalProfesionalesValidados++;
        if (element.tipo_pago === "CASH") {
          this.pagoEfectivo++;
        } else {
          this.pagoTarjeta++;
        }
      }
    } else {
      this.totalAutores++;
      if (element.estado === "V") {
        this.totalAutoresValidados++;
        if (element.tipo_pago === "CASH") {
          this.pagoEfectivo++;
        } else {
          this.pagoTarjeta++;
        }
      }
    }
  });
  this.inscritosValidados =
    this.totalEstudiantesValidados + this.totalProfesionalesValidados + this.totalAutoresValidados;
  //Calculano totales
  this.inscripcion.forEach((element) => {
    if (
      element.nom_inscr === "AUTHOR" ||
      element.nom_inscr === "AUTHOR UTA"
    ) {
      this.totalRecaudacionAutores =
        this.totalAutoresValidados * element.costo;
    } else if (element.nom_inscr === "PROFESSIONAL") {
      this.totalRecaudacionProfesionales =
        this.totalProfesionalesValidados * element.costo;
    } else {
      this.totalRecaudacionEstudiantes =
        this.totalEstudiantesValidados * element.costo;
    }
  });
}
```

Elaborado por: El investigador

El presente método permite guardar o editar las imágenes de la galería del lugar.

Ilustración 354. Frontend – Guardar galería lugar

```
guardarGaleriaLugar() {
  if (this.nombreGaleria !== "") {
    var data = new FormData();
    data.append("nombre", this.nombreGaleria);
    if (this.galeriaSeleccionada === undefined) {
      data.append("imagen", this.file);

      if (this.file !== undefined) {
        this._galeriaLugarService.insertGaleria(data).subscribe(() => {
          this.cerrarModal();
          this.getGaleriaLugar();
        });
      } else {
        this.error = true;
        setTimeout(() => {
          this.error = false;
        }, 5000);
      }
    } else {
      this.file !== undefined ? data.append("imagen", this.file) : "";
      this._galeriaLugarService
        .editGaleria(this.galeriaSeleccionada.id, data)
        .subscribe(() => {
          this.cerrarModal();
          this.getGaleriaLugar();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar las imágenes de la galería informativa.

Ilustración 355. Frontend – Guardar información de la galería

```
guardarGaleriaInfo() {
  if (this.galeriaSeleccionada === undefined) {
  }
  if (this.nombreGaleria !== "") {
    var data = new FormData();
    data.append("nombre", this.nombreGaleria);
    if (this.galeriaSeleccionada === undefined) {
      if (this.file !== undefined) {
        data.append("imagen", this.file);
        this._galeriaInformacionService.insertGaleria(data).subscribe(() => {
          this.cerrarModal();
          this.getGaleriaInformacion();
        });
      } else {
        this.error = true;
        setTimeout(() => {
          this.error = false;
        }, 5000);
      }
    } else {
      this.file !== undefined ? data.append("imagen", this.file) : "";
      this._galeriaInformacionService
        .editGaleria(this.galeriaSeleccionada.id, data)
        .subscribe(() => {
          this.cerrarModal();
          this.getGaleriaInformacion();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite editar la información de la presentación del evento.

Ilustración 356. Frontend – Guardar información de la presentación

```
guardarPresentacion() {
  if (this.textoPresentacion != "") {
    var data = new FormData();

    data.append("texto", this.textoPresentacion);
    this.file != undefined ? data.append("imagen_boton_1", this.file) : "";
    this.file2 != undefined ? data.append("imagen_boton_2", this.file2) : "";
    data.append("link_boton_1", this.link_boton_1);
    data.append("link_boton_2", this.link_boton_2);
    this._presentacionService
      .editPresentacion(1, data)
      .subscribe((result) => {
        this.getPresentation();
        this.cerrarModal();
      });
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite editar el lugar del evento.

Ilustración 357. Frontend – Guardar lugar del evento

```
guardarLugarDelEvento() {
  var data = new FormData();
  if (this.texto1LugarEvento != "" && this.texto2LugarEvento != "") {
    data.append("texto_1", this.texto1LugarEvento);
    data.append("texto_2", this.texto2LugarEvento);
    this.file != undefined ? data.append("imagen", this.file) : "";

    this._lugarDelEventoService
      .editLugarDelEvento(1, data)
      .subscribe((result) => {
        this.getLugarDelEvento();
        this.cerrarModal();
      });
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar la información turística según corresponda.

Ilustración 358. Frontend – Guardar información turística

```
guardarInformacionTuristica() {
  var data = new FormData();
  data.append("texto", this.textoInformacionTuristica);
  this.file != undefined ? data.append("boton1", this.file) : "";
  this.file2 != undefined ? data.append("boton2", this.file2) : "";
  this.link_boton_1 != "" ? data.append("link1", this.link_boton_1) : "";
  this.link_boton_2 != "" ? data.append("link2", this.link_boton_2) : "";
  if (this.informacionTuristicaSeleccionada === undefined) {
    if (this.textoInformacionTuristica != "") {
      this._informacionTuristica
        .insertInformacionTuristica(data)
        .subscribe((result) => {
          this.getInformacionTuristica();
          this.cerrarModal();
        });
    } else {
      this.error = true;
      setTimeout(() => {
        this.error = false;
      }, 5000);
    }
  } else {
    this._informacionTuristica
      .editInformacionTuristica(
        this.informacionTuristicaSeleccionada.id,
        data
      )
      .subscribe((result) => {
        this.getInformacionTuristica();
        this.cerrarModal();
      });
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar el programa según corresponda.

Ilustración 359. Frontend – Guardar programa

```
guardarPrograma() {
  if (this.descripcionPrograma != "") {
    var data = new FormData();
    data.append("descripcion", this.descripcionPrograma);
    this.file != undefined ? data.append("imagen", this.file) : "";
    this.file2 != undefined ? data.append("triptico", this.file2) : "";
    if (this.programaSeleccionado === undefined) {
      this._programaService.insertPrograma(data).subscribe((result) => {
        this.cerrarModal();
        this.getPrograma();
      });
    } else {
      this._programaService
        .editPrograma(this.programaSeleccionado.id, data)
        .subscribe((result) => {
          this.cerrarModal();
          this.getPrograma();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar la información de envío de trabajos según corresponda.

Ilustración 360. Frontend – Guardar envío de trabajos

```
guardarEnvioTrabajo() {
  if (
    this.textoEnvioTrabajos != "" &&
    this.botonEnvioTrabajos != "" &&
    this.linkEnvioTrabajos != ""
  ) {
    var data = new FormData();
    data.append("texto", this.textoEnvioTrabajos);
    this.file != undefined ? data.append("boton", this.file) : "";
    data.append("link", this.linkEnvioTrabajos);
    if (this.envioTrabajoSeleccionado === undefined) {
      this._envioTrabajosService
        .insertEnvioTrabajo(data)
        .subscribe((result) => {
          this.getEnvioTrabajos();
          this.cerrarModal();
        });
    } else {
      this._envioTrabajosService
        .editEnvioTrabajo(this.envioTrabajoSeleccionado.id, data)
        .subscribe((result) => {
          this.getEnvioTrabajos();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar la información de envío de trabajos según corresponda.

Ilustración 361. Frontend – Guardar formatos de envío de trabajos

```
guardarEnvioTrabajoFormato() {
  if (
    this.textoEnvioTrabajos != "" &&
    this.botonEnvioTrabajos != "" &&
    this.linkEnvioTrabajos != ""
  ) {
    var data = new FormData();
    data.append("texto", this.textoEnvioTrabajos);
    this.file != undefined ? data.append("boton", this.file) : "";
    data.append("link", this.linkEnvioTrabajos);
    if (this.envioTrabajoSeleccionado === undefined) {
      this._envioTrabajosFormatosService
        .insertEnvioTrabajoFormato(data)
        .subscribe((result) => {
          this.getEnvioTrabajosFormatos();
          this.cerrarModal();
        });
    } else {
      this._envioTrabajosFormatosService
        .editEnvioTrabajoFormato(this.envioTrabajoSeleccionado.id, data)
        .subscribe((result) => {
          this.getEnvioTrabajosFormatos();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar las fechas de los envíos de trabajo.

Ilustración 362. Frontend – Guardar fechas de envíos de trabajo

```
guardarEnvioTrabajoFecha() {
  if (
    this.fechaEnvioTrabajoFecha != "" &&
    this.actividadEnvioTrabajoFecha != ""
  ) {
    const enviar = {
      fecha: this.fechaEnvioTrabajoFecha,
      actividad: this.actividadEnvioTrabajoFecha,
    };

    if (this.envioTrabajoFechaSeleccionado === undefined) {
      this._envioTrabajosFechasService
        .insertEnvioTrabajoFecha(enviar)
        .subscribe((result) => {
          this.getEnvioTrabajosFechas();
          this.cerrarModal();
        });
    } else {
      this._envioTrabajosFechasService
        .editEnvioTrabajoFecha(this.envioTrabajoFechaSeleccionado.id, enviar)
        .subscribe((result) => {
          this.getEnvioTrabajosFechas();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite cambiar el estado del comité de activo o inactivo.

Ilustración 363. Frontend – Cambiar estado del comité

```
cambiarEstadoComite(item: Comite) {
  const enviar = {
    id: item.id,
    validacion: !item.validacion,
  };
  this._comiteService.editComite(enviar).subscribe((result) => {
    this.getComite();
  });
}
```

Elaborado por: El investigador

El presente método permitirá guardar o editar la información del temario según corresponda.

Ilustración 364. Frontend – Guardar temario

```
guardarTemario() {
  if (this.tituloTemario != "") {
    const enviar = {
      titulo: this.tituloTemario,
    };

    if (this.temarioSeleccionado === undefined) {
      this._temarioService.insertTemario(enviar).subscribe((result) => {
        this.getTemarios();
        this.cerrarModal();
      });
    } else {
      this._temarioService
        .editTemarios(this.temarioSeleccionado.id, enviar)
        .subscribe((result) => {
          this.getTemarios();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar los temas del temario según corresponda.

Ilustración 365. Frontend – Guardar temas del temario

```
guardarTemarioTema() {
  if (this.tituloTemario != "") {
    const enviar = {
      id_tema_per: this.idTemarioPertenece,
      tema: this.tituloTemario,
    };

    if (this.temarioTemaSeleccionado === undefined) {
      this._temarioTemasServices
        .insertTemarioTemas(enviar)
        .subscribe((result) => {
          this.getTemariosTemas();
          this.cerrarModal();
        });
    } else {
      this._temarioTemasServices
        .editTemarioTemas(this.temarioTemaSeleccionado.id, enviar)
        .subscribe((result) => {
          this.getTemariosTemas();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o editar los detalles del cronograma según corresponda.

Ilustración 366. Frontend – Guardar detalles del cronograma

```
guardarCronogramaDetalle() {
  if (
    this.cronogramaDetalleActividad != "" &&
    this.cronogramaDetalleHora != ""
  ) {
    const enviar = {
      hora: this.cronogramaDetalleHora,
      actividad: this.cronogramaDetalleActividad,
      dia_per: this.cronogramaDiaSeleccionado.id,
    };
    if (this.cronogramaDetalleSeleccionado === undefined) {
      this._programaDetalleService
        .insertProgramaDetalle(enviar)
        .subscribe((result) => {
          this.getProgramaDetalle();
          this.cerrarModal();
        });
    } else {
      this._programaDetalleService
        .editProgramaDetalle(this.cronogramaDetalleSeleccionado.id, enviar)
        .subscribe((result) => {
          this.getProgramaDetalle();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o actualizar los días del cronograma según corresponda.

Ilustración 367. Frontend – Guardar día del cronograma

```
guardarCronogramaDia() {
  if (this.cronogramaDiaFecha != "") {
    if (this.cronogramaDiaSeleccionado === undefined) {
      const enviar = {
        fecha: this.cronogramaDiaFecha,
        jornada_per: this.cronogramaJornadaSeleccionado.id,
      };
      this._programaDiasService
        .insertProgramaDias(enviar)
        .subscribe((result) => {
          this.getProgramaDias();
          this.cerrarModal();
        });
    } else {
      const enviar = {
        fecha: this.cronogramaDiaFecha,
      };
      this._programaDiasService
        .editProgramaDias(this.cronogramaDiaSeleccionado.id, enviar)
        .subscribe((result) => {
          this.getProgramaDias();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar o editar la galería de inicio.

Ilustración 368. Frontend – Guardar galería del inicio

```
guardarInicioGaleria() {
  if (this.inicioGaleriaImagen != "") {
    var data = new FormData();
    data.append("link", this.inicioGaleriaLink);
    if (this.file != undefined) {
      data.append("imagen", this.file);
    }
    if (this.inicioGaleriaSeleccionado === undefined) {
      if (this.file != undefined) {
        this._inicioGaleriaService
          .insertInicioGaleria(data)
          .subscribe((result) => {
            this.getInicioGaleria();
            this.cerrarModal();
          });
      } else {
        this.error = true;
        setTimeout(() => {
          this.error = false;
        }, 5000);
      }
    } else {
      this._inicioGaleriaService
        .editInicioGaleria(this.inicioGaleriaSeleccionado.id, data)
        .subscribe((result) => {
          this.getInicioGaleria();
          this.cerrarModal();
        });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite enviar correos de forma individual.

Ilustración 369. Frontend – Enviar correo de forma individual

```
enviarCorreoIndividual() {
  if (this.mensaje != "" && this.asunto != "") {
    if (this.destinatario != "") {
      const email = {
        asunto: this.asunto,
        destinatario: this.destinatario,
        mensaje: this.mensaje,
      };
      this._mailerService.mailInscritos(email).subscribe((result) => {
        this.Mensaje("Operación exitosa", "Correo enviado correctamente");
        this.cerrarModal();
        this.asunto = "";
        this.mensaje = "";
        this.destinatario = "";
      });
    }
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite guardar nuevos usuarios.

Ilustración 370. Frontend – Guardar nuevo usuario

```
guardarNuevoUsuario() {
  let nuevo = true;
  if (this.clave !== this.repetirClave) {
    Swal.fire({
      title: "Error",
      text: "Las contraseñas no coinciden",
      icon: "error",
      confirmButtonText: "Aceptar",
      confirmButtonColor: "#7A1E19",
      iconColor: "#7A1E19",
      color: "#7A1E19",
    });
    return;
  }
  if (this.usuario !== "" && this.clave !== "" && this.idRolUsuario !== null) {
    const usuario = {
      usuario: this.usuario,
      pass: this.clave,
      id_tipo_per: this.idRolUsuario,
    };
    this.usuarios.forEach((element) => {
      element.usuario === this.usuario ? (nuevo = false) : "";
    });
    if (nuevo) {
      this._usuariosService.insertUsuario(usuario).subscribe((result) => {
        this.cerrarModal();
        this.resetVariablesUsuario();
        this.getUsuarios();
      });
    } else {
      Swal.fire({
        title: "Error",
        text: "El nombre de usuario ya existe",
        icon: "error",
        confirmButtonText: "Aceptar",
        confirmButtonColor: "#7A1E19",
        iconColor: "#7A1E19",
        color: "#7A1E19",
      });
    }
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite editar los usuarios del sistema.

Ilustración 371. Frontend – Editar usuario

```
editarUsuario() {
  if (this.usuario !== "" && this.idRolUsuario !== null) {
    const usuario = {
      id: this.idUsuarioSeleccionado,
      usuario: this.usuario,
      id_tipo_per: this.idRolUsuario,
    };
    this._usuariosService.editUsuario(usuario).subscribe(
      (result) => {
        this.cerrarModal();
        this.resetVariablesUsuario();
        this.getUsuarios();
      },
      (err) => {}
    );
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite enviar correos por lotes.

Ilustración 372. Frontend – Enviar correo por lotes

```
enviarCorreoPorLotes() {
  if (this.mensaje != "" && this.asunto != "") {
    try {
      this.inscripcionesFiltradas.forEach((element) => {
        if (element.correo != "") {
          const email = {
            asunto: this.asunto,
            destinatario: element.correo,
            mensaje: this.mensaje,
          };
          this._mailerService.mailInscritos(email).subscribe((result) => {});
        }
      });
      this.Mensaje("Operación exitosa", "Correo enviado correctamente");
      this.cerrarModal();
    } catch (error) {}
  } else {
    this.error = true;
    setTimeout(() => {
      this.error = false;
    }, 5000);
  }
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de participación.

Ilustración 373. Frontend – Generar certificado de participación

```
generarCertificadosParticipacion(inscripcion: AdminsInscrip) {
  const enviar = {
    id: inscripcion.numero,
    nombre: inscripcion.nombres + " " + inscripcion.apellidos,
    cedula: inscripcion.di,
    correo: inscripcion.correo
  };
  this._certificadosPService
    .crearCertificado(enviar)
    .subscribe((result) => {});
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de participación por lotes

Ilustración 374. Frontend – Generar certificados de participación por lotes

```
generarTodosCertificadosParticipacion() {
  this.inscripcionesValidadas.forEach((element) => {
    const enviar = {
      id: element.numero,
      nombre: element.nombres + " " + element.apellidos,
      cedula: element.di,
      correo:element.correo
    };
    this._certificadosPService
      .crearCertificado(enviar)
      .subscribe((result) => {});
  });
  setTimeout(() => {
    this.getInscripciones();
  }, 5000);
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de asistencia.

Ilustración 375. Frontend – Generar certificado de asistencia

```
generarCertificadosAsistencia(inscripcion: AdminsInscrip) {
  const enviar = {
    id: inscripcion.numero,
    nombre: inscripcion.nombres + " " + inscripcion.apellidos,
    cedula: inscripcion.di,
    correo:inscripcion.correo
  };
  this._certificadosAsistenicaService
    .crearCertificado(enviar)
    .subscribe((result) => {});
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de asistencia por lotes.

Ilustración 376. Frontend – Generar certificados de asistencia por lotes

```
generarTodosCertificadosAsistencia() {
  this.inscripcionesValidadas.forEach((element) => {
    const enviar = {
      id: element.numero,
      nombre: element.nombres + " " + element.apellidos,
      cedula: element.di,
      correo: element.correo
    };
    this._certificadosAsistenciaService
      .crearCertificado(enviar)
      .subscribe((result) => {});
  });
  setTimeout(() => {
    this.getInscripciones();
  }, 5000);
}
```

Elaborado por: El investigador

El presente método permite generar los certificados del comité.

Ilustración 377. Frontend – Generar certificado comité

```
generarCertificadosComite(comite: Comite) {
  const enviar = {
    id: comite.id,
    nombre: comite.nombre,
  };
  this._certificadosCService
    .crearCertificado(enviar)
    .subscribe((result) => {
      this.getComite();
    });
}
```

Elaborado por: El investigador

El presente método permite generar certificados de los expositores.

Ilustración 378. Frontend – Generar certificado expositor

```
generarCertificadoExpositor(expositor: Expositor) {  
  const enviar = {  
    id: expositor.id,  
    id_paper:expositor.id_paper,  
    nombre: expositor.nombre,  
    titulo: expositor.titulo,  
  };  
  this._expositorService.generarCertificado(enviar).subscribe((result) => {});  
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de los expositores por lotes.

Ilustración 379. Frontend – Generar certificado de expositor

```
generarTodoCertificadoExpositor() {  
  this.expositores.forEach((element) => {  
    console.log(element.nombre)  
    const enviar = {  
      id: element.id,  
      id_paper:element.id_paper,  
      nombre: element.nombre,  
      titulo: element.titulo,  
    };  
    this._expositorService.generarCertificado(enviar).subscribe((result) => {});  
  });  
  setTimeout(() => {  
    this.getExpositores();  
  }, 5000);  
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de autores.

Ilustración 380. Frontend – Generar certificado de autor

```
generarCertificadoAutor(autor: Autores) {  
  let nombres=""  
  let cont =0;  
  autor.autores.forEach((element)=>{  
    nombres +=element.Nombre1 + " " + element.Apellido1;  
  
    if(cont !=autor.autores.length-1){  
      nombres+=", "  
    }  
    cont++;  
  });  
  
  const enviar = {  
    id: autor.paper_id,  
    nombre: nombres,  
    titulo: autor.titulo  
  };  
  this._autoresService.editAutor(enviar).subscribe((result) => {});  
}
```

Elaborado por: El investigador

El presente método permite generar certificados de autores por lotes.

Ilustración 381. Frontend – Generar certificados de autores por lotes

```
generarTodoCertificadoAutor() {
  this.autores.forEach((autor) => {
    let nombres=""
    let cont =0;
    autor.autores.forEach((element)=>{
      nombres +=element.Nombre1 + " " + element.Apellido1;

      if(cont !=autor.autores.length-1){
        nombres+=", "
      }
      cont++;
    });

    const enviar = {
      id: autor.paper_id,
      nombre: nombres,
      titulo: autor.titulo
    };
    this._autoresService.editAutor(enviar).subscribe((result) => {});
  });
  setTimeout(() => {
    this.getAutores();
  }, 5000);
}
```

Elaborado por: El investigador

El presente método permite generar los certificados de comité por lotes.

Ilustración 382. Frontend – Generar certificados de comité por lotes

```
generarTodosCertificadosComite() {
  this.comite.forEach((element) => {
    if (element.validacion) {
      const enviar = {
        id: element.id,
        nombre: element.nombre,
      };
      this._certificadosCService
        .crearCertificado(enviar)
        .subscribe((result) => {});
    }
  });
  setTimeout(() => {
    this.getComite();
  }, 5000);
}
```

Elaborado por: El investigador

3.2.4 Fase IV: Pruebas

Las pruebas son una herramienta esencial en el desarrollo de software, ya que indica si el software en desarrollo está en condiciones óptimas para su implementación en un entorno de producción. Estas pruebas son fundamentales para que tanto los desarrolladores como los usuarios finales queden satisfechos con el trabajo realizado y sepan si el software se adhiere a las pautas descritas en la historia del usuario.

Tabla 71. Prueba de aceptación 001

PRUEBA DE ACEPTACIÓN
Número: 001
Identificador de Historia: H009
Denominación: Inicio se sesión
Descripción: Los usuarios del sistema podrán acceder a la aplicación mediante autenticación de usuario y contraseña.
Condición de ejecución: La aplicación mostrará una pantalla de inicio de sesión con los respectivos campos necesarios como usuario y contraseña.
Interfaz: La pantalla validara los campos necesarios para el inicio de sesión.
Resultado esperado: El usuario es verificado y accede al módulo de configuración.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 72. Prueba de aceptación 002

PRUEBA DE ACEPTACIÓN
Número: 002
Identificador de Historia: H010
Denominación: Consulta de información.

Descripción: Los usuarios del sistema podrán acceder a una pantalla con diferentes módulos en los cuales contarán con toda la información correspondiente al congreso CSEI22.
Condición de ejecución: El sistema mostrar la información que le corresponda en cada módulo.
Interfaz: La pantalla estará compuesta por un menú con diversos módulos cada uno con su respectiva información.
Resultado esperado: El usuario accede a la información del módulo y además puede navegar entre las diferentes pestañas.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 73. Prueba de aceptación 003

PRUEBA DE ACEPTACIÓN
Número: 003
Identificador de Historia: H011
Denominación: Registro de participante.
Descripción: Los participantes del congreso podrán acceder a un formulario con los respectivos campos para su registro
Condición de ejecución: La pantalla mostrada los campos necesarios para realizar la inscripción en el congreso CSEI22.
Interfaz: La pantalla estará compuesta por 4 formularios diferentes cada uno requiriendo su respectiva información
Resultado esperado: El usuario tiene acceso al registro y puede completar todos los campos para completar su inscripción.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 74. Prueba de aceptación 004

PRUEBA DE ACEPTACIÓN

Número: 004
Identificador de Historia: H012
Denominación: Validación de registro.
Descripción: Los participantes del congreso podrán acceder al módulo de validación ingresando su número de cedula.
Condición de ejecución: La pantalla mostrará el formulario de validación una vez se ingrese el número de cedula o pasaporte luego tendrá que llenar el formulario de validación para poder guardar la información
Interfaz: La pantalla validara los datos ingresados en los formularios.
Resultado esperado: El usuario es verificado, se le permite cargar su comprobante de pago y llenar el detalle de este validando su registro.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 75. Prueba de aceptación 005

PRUEBA DE ACEPTACIÓN
Número: 005
Identificador de Historia: H013
Denominación: Consulta de facturas.
Descripción: Los participantes del congreso podrán acceder al módulo de facturas ingresando su número de cedula.
Condición de ejecución: La pantalla mostrará el formulario de facturas una vez se halla ingresado el número de cedula o pasaporte, en esta pantalla podrá visualizar su factura
Interfaz: La pantalla validara los datos ingresados en los formularios y presentara la respectiva factura del participante
Resultado esperado: El usuario es verificado, puede visualizar su factura y descargar esta.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 76. Prueba de aceptación 006

PRUEBA DE ACEPTACIÓN
Número: 006
Identificador de Historia: H014
Denominación: Consulta de certificados.
Descripción: Los participantes del congreso podrán acceder al módulo de certificados ingresando su número de cedula.
Condición de ejecución: La pantalla mostrará el formulario de certificados una vez se halla ingresado el número de cedula o pasaporte, en esta pantalla podrá visualizar sus respectivos certificados
Interfaz: La pantalla validara los datos ingresados en los formularios y presentara los respectivos certificados del participante
Resultado esperado: El usuario es verificado, puede visualizar sus certificados y descargar estos.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Tabla 77. Prueba de aceptación 007

PRUEBA DE ACEPTACIÓN
Número: 007
Identificador de Historia: H015
Denominación: Configuración de información CSEI.
Descripción: Los usuarios del sistema podrán gestionar toda la información de la página informativa y todos los módulos que contiene la aplicación.
Condición de ejecución: La aplicación mostrar aun pantalla con diferentes submenús desde los cuales se podrá controlar la configuración de cada sección del sistema.

Interfaz: Cada uno de los submódulos validaran los datos en su respectivo formulario.
Resultado esperado: El usuario puede insertar, actualizar y eliminar la información de cada uno de los submódulos de configuración. El usuario puede validar inscritos, comunicarse con ellos, generar certificados y reportes.
Resultado de la prueba: Prueba aceptada satisfactoriamente.

Elaborado por: El investigador

Descripción de procesos con la implementación del sitio web.

Con la implementación del sitio web, los procesos involucrados en el desarrollo del CSEI son los presentados en la siguiente tabla.

Tabla 78: Cuadro de actividades y procesos del CSEI con la implementación del sitio web.

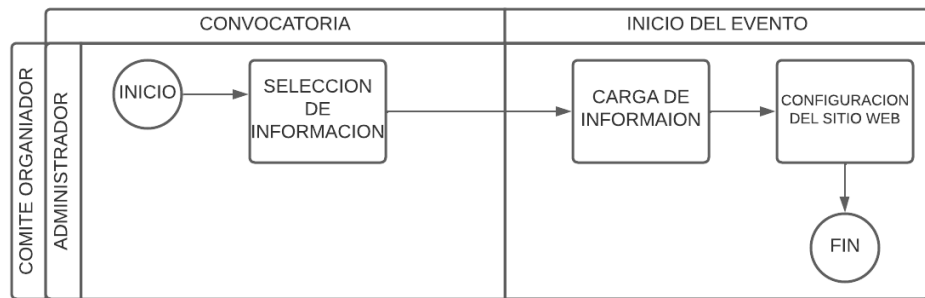
ACTIVIDAD	PROCESOS
Convocatoria.	1. Mediante la notificación de aprobación y presentación del manuscrito final del congreso CSEI. Los organizadores del evento realizan la convocatoria para participar en el congreso a través de la página web, cargando la información del congreso a través del panel de configuración del sitio web.
Publicación del evento.	1. Los miembros del comité organizador seleccionan la información que será presentará en la página web. 2. Cargan la información al sitio web mediante el panel de configuración de la página informativa.
Inicio Congreso	1. Se activa el sistema de registro en el apartado de configuraciones del sitio web.
Inscripciones	1. Los usuarios acceden a la página informativa para ser redirigidos hacia el sistema de registro que está integrado en el mismo sitio web.

<p>Registro</p>	<ol style="list-style-type: none"> 1. Los usuarios ingresan al sistema de registro. 2. Ingreso de cedula para verificar si existe o no en los registros de la base de datos, si existe los datos se auto completan, caso contrario al usuario se le permite rellenar todos los campos para la inscripción. Los usuarios realizan su registro llenando los datos solicitados acorde a su tipo de participante. Aceptan su registro y finalizan con este para posteriormente visualizar la información de pago misma que también se envía por correo electrónico al participante. 3. En caso de existir un error por parte del participante este podrá volver a ingresar al sistema de registro para modificar sus datos.
<p>Validación</p>	<ol style="list-style-type: none"> 1. Los administradores habilitan la opción de validación. 2. Los usuarios acceden al apartado de validación del sistema de registro. 3. Ingresan su documento de identificación y realizan la validación de su registro subiendo la imagen, numero del comprobante de pago y una descripción de este. 4. Los usuarios reciben un correo electrónico en el cual se indica que su solicitud será aprobada.
<p>Facturación</p>	<ol style="list-style-type: none"> 1. Un representante de la EP UTA ingresa a la parte de validación de registros del sistema de registro y procede a validar a los usuarios registrados que ya registren su comprobante de pago cargado. 2. Revisan uno a uno los comprobantes de pago y en caso de un error por parte del participante, se comunican mediante un botón de correo electrónico con este. En caso de que la información este correcta el representante de EP UTA valida al usuario. 3. Una vez que se valide el pago de un participante se factura mediante un sistema propio de la EP UTA y se carga la factura a través del sistema,

	la carga de facturas se puede realizar de forma individual o por lotes.
Cierre del congreso	1. Los organizadores del congreso deshabilitan la opción de registro y validación.
Emisión de certificados	1. El administrador mediante el apartado de certificados genera los certificados para todos los involucrados en el evento CSEI.
Generación de reportes	1. El comité organizador mediante el apartado de reportes genera reportes generales y listados.

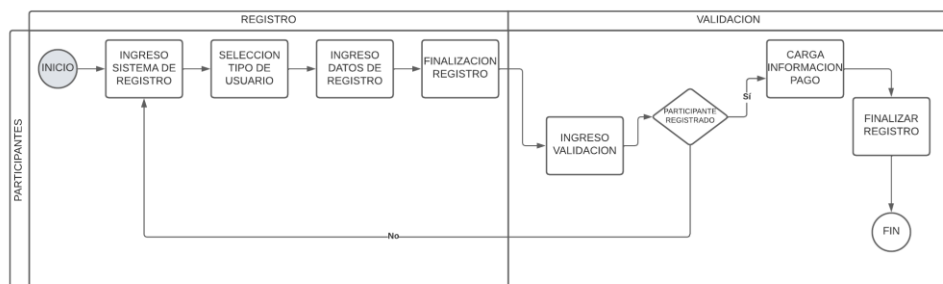
Elaborado por: El investigador

Ilustración 383: Inicio CSEI 2022



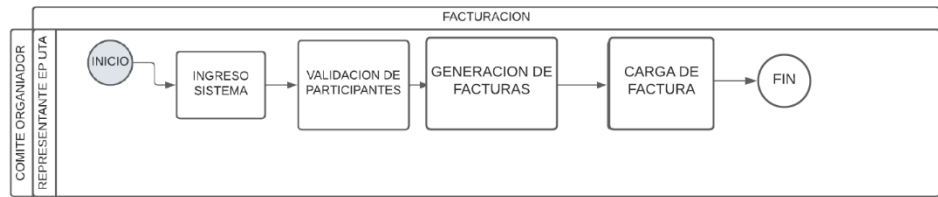
Elaborado por: El investigador

Ilustración 384: Registro y Validación CSEI 2022



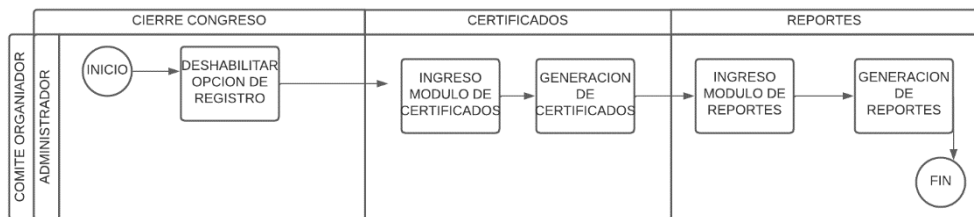
Elaborado por: El investigador

Ilustración 385: facturación CSEI 2022



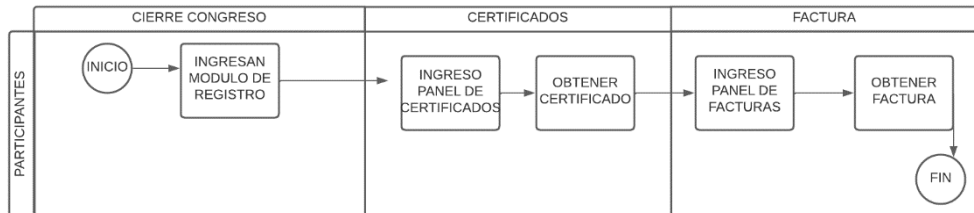
Elaborado por: El investigador

Ilustración 386: Fin CSEI 2022 (Procesos Administrativos)



Elaborado por: El investigador

Ilustración 387: Fin CSEI 2022 (Procesos Participantes)



Elaborado por: El investigador

CAPÍTULO IV

4 Conclusiones y recomendaciones

4.1. Conclusiones

- Al analizar los procesos y necesidades de los involucrados en la organización del congreso internacional CSEI, así como de los interesados en asistir a este, se pudo determinar los requerimientos funcionales para el sitio web, con lo que se pudo implementar herramientas que automaticen los procesos que son parte del desarrollo del evento.
- Se investigó la aplicación de metodologías ágiles en el desarrollo de aplicaciones web, finalmente seleccionando XP como marco para el desarrollo de este sitio web debido a su adaptabilidad ante este tipo de proyectos y el bajo riesgo de errores que es característico de esta metodología.
- Se estudiaron diferentes tecnologías para el desarrollo del sitio web, de las que se seleccionó Angular como framework para la creación del front-end y Node.js para el back-end ya que estas son compatibles entre sí y cuentan con características que las hacen superiores a otras alternativas open source.
- La implementación del sitio Web se ha realizado de forma satisfactoria, funcionando durante el desarrollo del congreso internacional CSEI edición 2022.

4.2 Recomendaciones

- Se recomienda la utilización de navegadores actuales para tener una mejor experiencia al interactuar con el sitio web.
- Se sugiere que en cada edición del congreso CSEI se de mantenimiento al servidor y se mantenga actualizadas las versiones de Angular, MySQL y Node.js.
- Adicionalmente, se recomienda que en cada edición del congreso CSEI se realice un respaldo de la base de datos y se de mantenimiento a esta.

BIBLIOGRAFÍA

- [1] R. Valarezo, COMPARACIÓN DE TENDENCIAS TECNOLÓGICAS EN APLICACIONES WEB, 2018.
- [2] E. Wohlgethan, Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js, 2018.
- [3] M. Navajas, Evaluación de la facilidad de aprendizaje de frameworks JavaScript: Backbone, Angular y Ember, 2019.
- [4] I. Mina, Desarrollo de aplicaciones web y móvil para la gestión de publicaciones científicas”, 2021.
- [5] C. QUISPE, “Sistema web para el seguimiento y control de documentos para la gestión administrativa en la red de salud Puno, PUNO, 2019.
- [6] A. Mera, “Aplicación web para gestionar los procesos administrativos de las Asociaciones Agropecuarias del cantón Chone de la provincia de Manabí”, Porto Viejo, 2019.
- [7] F. Vallejo, “Aplicación Multiplataforma (Web / Android) para la gestión y control del Proceso de Titulación de los estudiantes de la Facultad de Tecnologías de la Información, Telecomunicaciones e Industrial, Ambato, 2020.
- [8] P. Urrutia, International Conference on Computer Science, Electronics and Industrial Engineering, Ambato, 2019.
- [9] IEEE, 729-1993, 2019.
- [10] Free software fundatios, Software, 2019.
- [11] SoftwareLab, ¿Qué es un sitio web y una página web?, 2022.
- [12] Ivan de Souza, Entiende las diferencias entre Front-End y Back-end en el ambiente de los sitios web, 2021.
- [13] barzana, Páginas web dinámicas, 2021.
- [14] Onion, Ceros & Unos, 2020.
- [15] R. Arjonilla, BackEnd, 2020.
- [16] Millenium, Sitios Web, 2020.

- [17] Mora, Sistemas web, 2010.
- [18] Báez, Sistemas de Internet, 2013 .
- [19] IDA, Diferencias entre una aplicación web y un sitio web, 2016.
- [20] IBM, Servicios web, 2019.
- [21] IBM, SOAP, 2021.
- [22] RedHat, Diferencias entre REST y SOAP, 2019.
- [23] N. A. Vázquez, «Sitios web y aplicaciones nativas para móviles en bibliotecas,» , 2015. [En línea]. Available: <https://gredos.usal.es/handle/10366/128006>. [Último acceso: 5 1 2023].
- [24] I. RAMÍREZ, ¿Qué es una Aplicación Web Progresiva o PWA?, 2021.
- [25] Native Team, ¿Qué Es Una PWA? Y ¿Cuáles Son Sus Características?, 2020.
- [26] Desarrollo web, PWA, 2019.
- [27] MDN, Service Worker API, 2020.
- [28] programador clic , Arquitectura de PWA, 2019.
- [29] Digitalgrowth, Progressive Web Apps (PWA): Qué son y cómo funcionan, 2022.
- [30] vuejs.org, What is Vue?, 2020.
- [31] Tecnologías Web, Lenguajes para aplicaciones web, 2017.
- [32] Desarrolloweb, Qué es React. Por qué usar React, 2019.
- [33] J. Torres, ¿Qué es el DOM? El significado del Modelo de Objeto de Documento en JavaScript, 2020.
- [34] ICONIC FRAMEWORK, The mobile SDK for the Web., 2021.
- [35] Flutter.dev, Build apps for any screen, 2022.
- [36] P. Vinodhini, AN INTERPRETATION OF DART PROGRAMMING LANGUAGE, 2021.
- [37] Angular, Introducción a la Documentación de Angular, 2021.
- [38] V. Aristizabal, Arquitectura en Angular, 2021.
- [39] typescriptlang, What is TypeScript?, 2021.
- [40] oracle, ¿Qué es el lenguaje de consulta estructurada (SQL)?, 2022.

- [41] MICROSOFT, Access SQL: conceptos básicos, vocabulario y sintaxis, 2022.
- [42] MySQL, MySQL Standard Edition, 2022.
- [43] B. d. Austro, Conoce RedyPAgo, 2022.
- [44] International Journal of Interactive Mobile Technologies, Agile Software Development: Methodologies and Trends, 2020.
- [45] IONOS, Páginas web: creación y gestión - IONOS Digital Guide, 2022.
- [46] scrum, Welcome to the Home of Scrum!, 2022.
- [47] CSEI, Congresos internacionas CSEI, 2022.
- [48] S. J. Reiser, La medicina y el imperio de la tecnología / Medicine and the impire of the technology, Mexico D.F.: Fondo de Cultura Economica , 1990.