



Universidad del Azuay
Facultad de Ciencias de la Administración
Escuela de Ingeniería de Sistemas

**Desarrollo de aplicaciones web mediante HTML5 y la base de datos
NoSQL MONGO DB.**

**Tesis previo a la obtención del título de
Ingeniero en Sistemas.**

Autor: Henry Daniel Campoverde Borja

Director: Ing. Pablo Esquivel León

Cuenca, Ecuador

2012

DEDICATORIA:

A mis padres: Hernán y Ana, a mis hermanos: Hernán, Viviana, Geovanny e Ismael por su sacrificio y apoyo incondicional, a mis sobrinos: Paúl, Valeria y Sofía por darme la fuerza para seguir cada día adelante, y a toda la gente que ha sido parte de mi vida estudiantil durante todo el tiempo.

AGRADECIMIENTO:

Expreso mi agradecimiento a la Universidad del Azuay, a la Escuela de Ingeniería de Sistemas, a todos los colaboradores y cordiales amigos; de manera especial a mi director el Ing. Pablo Esquivel León, por su apoyo en el transcurso de mi formación académica, y por brindarme su amistad desinteresada.

RESUMEN

HTML desde su primera versión descrita tiempo atrás, ha ido evolucionando hasta llegar a su actual versión HTML5. Este proyecto de tesis está basado en aquella nueva versión, desarrollando así una aplicación web orientada a redes sociales, con el objetivo de exponer los cambios y mejoras que han surgido en HTML5, agregando funcionalidades de JavaScript que han surgido a la par con HTML5 y añadiendo una nueva alternativa para el almacenamiento de datos como lo es MongoDB, una base de datos NoSQL, que intenta solventar limitaciones del modelo relacional al momento de trabajar con una cantidad masiva de datos.

ABSTRACT

Since its first version described long ago, HTML has evolved to its present version, HTML5. This research project is based on that new version. We developed a web application oriented to social networks with the purpose of presenting the changes and improvements that have emerged from HTML5. We added JavaScript functionalities that have developed at the same time as HTML5 and added a new alternative for data storing, which is Mongo DB, and a NoSQL database that tries to solve the limitations of the relational model when it has to operate with a massive quantity of data.



Diana Lec Rodas
Translated by,
Diana Lec Rodas

ÍNDICE DE CONTENIDOS

Dedicatoria	ii
Agradecimiento	iii
Resumen	iv
Abstract	v
Índice de contenidos.....	1
Índice de ilustraciones.....	7
Índice de tablas.....	16
Introducción	18
CAPÍTULO 1 HTML5	20
1.1 Introducción.....	20
1.2 ¿Qué es html5?	20
1.3 Compatibilidad en los navegadores.....	21
1.3.1 Navegadores más comunes de escritorio.....	22
1.3.2 Navegadores más comunes en tabletas (tablets).....	22
1.3.3 Navegadores más comunes en móviles (mobiles).....	23
1.3.4 Navegadores más comunes en juegos (gaming).....	23
1.3.5 Navegadores más comunes en televisiones	24
1.4 Sintaxis html5.....	24
1.4.1 La declaración del tipo de documento (doctype).....	24
1.4.2 ¿Qué es un elemento?	25
1.4.3 ¿Qué son los atributos?	27
1.4.3.1 Atributos globales	28
1.4.3.1.1 Atributos esenciales.....	28
1.4.3.1.2 Atributos de eventos.....	29
1.5 Elementos html5.....	32
1.5.1 El elemento raíz	33
1.5.1.1 Html	33
1.5.2 Documento de metadatos.....	34
1.5.2.1 Head	34
1.5.2.2 Title.....	35
1.5.2.3 Base.....	36
1.5.2.4 Link.....	37
1.5.2.5 Meta	38
1.5.2.6 Style	39
1.5.3 Scripting.....	41
1.5.3.1 Script.....	41
1.5.3.2 Noscript.....	42
1.5.4 Secciones	43
1.5.4.1 Body.....	43
1.5.4.2 Article	46
1.5.4.3 Section.....	47
1.5.4.4 Nav.....	49
1.5.4.5 Aside	50
1.5.4.6 H1, h2, h3, h4, h5, h6.....	51

1.5.4.7 Hgroup	52
1.5.4.8 Header	53
1.5.4.9 Footer	54
1.5.4.10 Address	55
1.5.5 Agrupación de contenido.....	56
1.5.5.1 P	56
1.5.5.2 Hr	57
1.5.5.3 Pre	58
1.5.5.4 Blockquote	59
1.5.5.5 Ol.....	60
1.5.5.6 Ul.....	61
1.5.5.7 Li	62
1.5.5.8 Dl.....	62
1.5.5.9 Dt.....	63
1.5.5.10 Dd.....	64
1.5.5.11 Figure	65
1.5.5.12 Figcaption	66
1.5.5.13 Div.....	66
1.5.6 Semántica nivel-texto	67
1.5.6.1 A.....	67
1.5.6.2 Em.....	69
1.5.6.3 Strong.....	70
1.5.6.4 Small	70
1.5.6.5 S	71
1.5.6.6 Cite.....	72
1.5.6.7 Q.....	72
1.5.6.8 Dfn	73
1.5.6.9 Abbr	74
1.5.6.10 Time	75
1.5.6.11 Code	75
1.5.6.12 Var.....	76
1.5.6.13 Samp	77
1.5.6.14 Kbd.....	77
1.5.6.15 Sub	78
1.5.6.16 Sup	79
1.5.6.17 I.....	79
1.5.6.18 B.....	80
1.5.6.19 U.....	81
1.5.6.20 Mark.....	81
1.5.6.21 Ruby.....	82
1.5.6.22 Rt.....	83
1.5.6.23 Rp.....	83
1.5.6.24 Bdi.....	84
1.5.6.25 Bdo.....	85
1.5.6.26 Span.....	85
1.5.6.27 Br.....	86
1.5.6.28 Wbr	86
1.5.7 Ediciones.....	87
1.5.7.1 Ins.....	87
1.5.7.2 Del.....	88

1.5.8 Contenido incrustado (embebido).....	89
1.5.8.1 Img.....	89
1.5.8.2 Iframe.....	91
1.5.8.3 Embed.....	92
1.5.8.4 Object.....	93
1.5.8.5 Param.....	95
1.5.8.6 Video.....	96
1.5.8.7 Audio.....	98
1.5.8.8 Source.....	100
1.5.8.9 Track.....	101
1.5.8.10 Canvas.....	102
1.5.8.11 Map.....	103
1.5.8.12 Area.....	104
1.5.9 Datos tabulares.....	107
1.5.9.1 Table.....	107
1.5.9.2 Caption.....	108
1.5.9.3 Colgroup.....	109
1.5.9.4 Col.....	110
1.5.9.5 Tbody.....	111
1.5.9.6 Thead.....	112
1.5.9.7 Tfoot.....	113
1.5.9.8 Tr.....	113
1.5.9.9 Td.....	114
1.5.9.10 Th.....	115
1.5.10 Formularios.....	116
1.5.10.1 Form.....	116
1.5.10.2 Fieldset.....	118
1.5.10.3 Legend.....	119
1.5.10.4 Label.....	120
1.5.10.5 Input.....	121
1.5.10.5.1 Input (type=hidden).....	125
1.5.10.5.2 Input (type=text).....	126
1.5.10.5.3 Input (type=search).....	127
1.5.10.5.4 Input (type=tel).....	128
1.5.10.5.5 Input (type=url).....	129
1.5.10.5.6 Input (type=email).....	129
1.5.10.5.7 Input (type=password).....	130
1.5.10.5.8 Input (type=datetime).....	131
1.5.10.5.9 Input (type=date).....	132
1.5.10.5.10 Input (type=month).....	133
1.5.10.5.11 Input (type=week).....	134
1.5.10.5.12 Input (type=time).....	134
1.5.10.5.13 Input (type=datetime-local).....	135
1.5.10.5.14 Input (type=number).....	136
1.5.10.5.15 Input (type=range).....	137
1.5.10.5.16 Input (type=color).....	138
1.5.10.5.17 Input (type=checkbox).....	138
1.5.10.5.18 Input (type=radio).....	139
1.5.10.5.19 Input (type=file).....	140
1.5.10.5.20 Input (type=submit).....	140

1.5.10.5.21 Input (type=image).....	141
1.5.10.5.22 Input (type=reset).....	142
1.5.10.5.23 Input (type=button).....	142
1.5.10.6 Button.....	143
1.5.10.7 Select.....	144
1.5.10.8 Datalist.....	146
1.5.10.9 Optgroup.....	147
1.5.10.10 Option.....	148
1.5.10.11 Textarea.....	149
1.5.10.12 Keygen.....	151
1.5.10.13 Output.....	152
1.5.10.14 Progress.....	153
1.5.10.15 Meter.....	154
1.5.11 Elementos interactivos.....	155
1.5.11.1 Details.....	155
1.5.11.2 Summary.....	156
1.5.11.3 Command.....	156
1.5.11.4 Menu.....	158
CAPÍTULO 2 MONGODB.....	160
2.1 Introducción.....	160
2.2 ¿Qué es Nosql?.....	160
2.3 ¿Qué es MongoDB?.....	161
2.4 Instalación de MongoDB.....	166
2.4.1 Instalación en Windows.....	166
2.4.2 Instalación en Linux.....	172
2.5 Php con MongoDB.....	175
2.5.1 Instalación del controlador.....	175
2.5.1.1 Instalación en Windows.....	175
2.5.1.2 Instalación en Linux.....	176
2.5.2 Conexión.....	176
2.5.3 Seleccionar una base de datos y una colección.....	178
2.5.4 Operaciones de escritura.....	178
2.5.4.1 Insertar documentos.....	179
2.5.4.2 Actualizar documentos.....	181
2.5.4.2.1 Actualizaciones de modificación.....	182
2.5.4.2.2 Actualizaciones de reemplazo.....	189
2.5.4.3 Eliminar documentos.....	190
2.5.4.4 Eliminar colección.....	191
2.5.4.5 Eliminar base de datos.....	191
2.5.5 Consultas.....	191
2.5.6 Manejo de archivos.....	198
2.5.7 Seguridad.....	202
2.5.7.1 Ataques de inyección de código.....	202
2.5.7.2 Operaciones seguras.....	203
CAPÍTULO 3 HERRAMIENTAS UTILIZADAS EN LA APLICACIÓN.....	204
3.1 Introducción.....	204
3.2 Librerías Javascript.....	204
3.2.1 JQuery.....	204
3.2.2 JQuery UI.....	206
3.2.3 Tablesorter.....	206

3.2.4 Printarea.....	207
3.2.5 Keyboard.....	207
3.3 Bootstrap	208
3.4 API's Javascript	210
3.4.1 Audio y video	211
3.4.2 Canvas.....	212
3.4.3 Geolocation.....	222
3.4.4 Web storage	224
3.4.5 Web sql database	226
3.4.6 Drag and drop	230
3.4.7 Web workers.....	235
3.4.8 Webnotifications.....	236
3.5 Nodejs.....	240
CAPÍTULO 4 MANUAL DE USO DE LA APLICACIÓN WEB.....	242
4.1 Introducción.....	242
4.2 Descripción de la aplicación.....	242
4.3 Manual para usuarios.....	243
4.3.1 Página de inicio.....	243
4.3.1.1 Registrarse.....	244
4.3.1.2 Información sobre la aplicación.....	246
4.3.1.3 Iniciar sesión	248
4.3.2 Página principal de usuarios	250
4.3.2.1 Contenidos	250
4.3.2.2 Menú superior	252
4.3.2.2.1 Botón nombre de usuario	252
4.3.2.2.2 Botón Ver mi perfil	252
4.3.2.2.3 Buscador de usuarios ó personas.....	253
4.3.2.2.4 Botón inicio	255
4.3.2.2.5 Botón cerrar sesión.....	255
4.3.2.3 Menú izquierdo	255
4.3.2.3.1 Foto de perfil	257
4.3.2.3.2 Función Noticias	258
4.3.2.3.3 Función Mi cuenta.....	259
4.3.2.3.4 Función Personalización	264
4.3.2.3.5 Función Servicios.....	266
4.3.2.3.6 Función Fotos	268
4.3.2.3.7 Función Contactos.....	272
4.3.2.3.8 Función Mensajes.....	273
4.3.2.3.9 Función Ubicación	274
4.3.2.3.11 Función Entretenimiento	285
4.3.2.4 Barra de información	286
4.3.2.5 Herramienta de chat	287
4.3.3 Perfil visible a los demás usuarios.....	291
4.3.3.1 Botón agregar/eliminar contacto:.....	291
4.3.3.2 Foto de perfil:.....	292
4.3.3.3 Contenido:.....	292
4.3.3.4 Menú de navegación:	293
4.4 Manual para administrador.....	296
4.4.1 Página de inicio.....	296
4.4.1.1 Identificarse.....	296

4.4.2	Página principal	298
4.4.2.1	Tabla de reportes generada	298
4.4.2.2	Tipos de reportes	298
4.4.2.2.1	Reporte por usuario	299
4.4.2.2.2	Reporte por fecha específica	300
4.4.2.2.3	Reporte por mes y año específico	301
4.4.2.2.4	Reporte total	302
4.4.2.3	Cantidad de usuarios registrados	303
4.4.2.4	Salir del administrador	304
	Conclusiones	305
	Recomendaciones	306
	Bibliografía	307
	Referencias electrónicas:	307

ÍNDICE DE ILUSTRACIONES

Imagen 1. Doctype html5	25
Imagen 2. Árbol DOM HTML5	25
Imagen 3. Estructura de los Elementos HTML.....	28
Imagen 4. Ejemplo elemento html.	34
Imagen 5. Ejemplo elemento head.	35
Imagen 6. Ejemplo elemento title.	36
Imagen 7. Ejemplo elemento base.	37
Imagen 8. Ejemplo elemento link.	38
Imagen 9. Ejemplo elemento meta.	39
Imagen 10. Ejemplo elemento style.	40
Imagen 11. Ejemplo elemento script.....	42
Imagen 12. Ejemplo elemento noscript.....	43
Imagen 13. Ejemplo elemento body.....	45
Imagen 14. Ejemplo elemento article.....	47
Imagen 15. Ejemplo elemento section.	48
Imagen 16. Ejemplo elemento nav.....	49
Imagen 17. Ejemplo elemento aside.....	51
Imagen 18. Ejemplo elementos h1, h2, h3, h4, h5, h6.....	52
Imagen 19. Ejemplo elemento hgroup.	53
Imagen 20. Ejemplo elemento header.....	54
Imagen 21. Ejemplo elemento footer.....	55
Imagen 22. Ejemplo elemento address.....	56
Imagen 23. Ejemplo elemento p.....	57
Imagen 24. Ejemplo elemento hr.	58
Imagen 25. Ejemplo elemento pre.	59
Imagen 26. Ejemplo elemento blockquote.....	60
Imagen 27. Ejemplo elemento ol.	61
Imagen 28. Ejemplo elemento ul.	61
Imagen 29. Ejemplo elemento li.	62
Imagen 30. Ejemplo elemento dl.	63
Imagen 31. Ejemplo elemento dt.	64
Imagen 32. Ejemplo elemento dd.....	65
Imagen 33. Ejemplo elemento figure.....	66
Imagen 34. Ejemplo elemento div.	67
Imagen 35. Ejemplo elemento a.....	69
Imagen 36. Ejemplo elemento em.....	69
Imagen 37. Ejemplo elemento strong.....	70
Imagen 38. Ejemplo elemento small.....	71
Imagen 39. Ejemplo elemento s.....	71
Imagen 40. Ejemplo elemento cite.....	72
Imagen 41. Ejemplo elemento q.....	73
Imagen 42. Ejemplo elemento dfn.	74
Imagen 43. Ejemplo elemento abbr.	74
Imagen 44. Ejemplo elemento time.	75

Imagen 45. Ejemplo elemento code.....	76
Imagen 46. Ejemplo elemento var.	77
Imagen 47. Ejemplo elemento samp.	77
Imagen 48. Ejemplo elemento kbd.....	78
Imagen 49. Ejemplo elemento sub.	78
Imagen 50. Ejemplo elemento sup.	79
Imagen 51. Ejemplo elemento i.	80
Imagen 52. Ejemplo elemento b.....	81
Imagen 53. Ejemplo elemento u.....	81
Imagen 54. Ejemplo elemento mark.	82
Imagen 55. Ejemplo elemento ruby.	83
Imagen 56. Ejemplo elemento rp.	84
Imagen 57. Ejemplo elemento bdi.	84
Imagen 58. Ejemplo elemento bdo.....	85
Imagen 59. Ejemplo elemento span.	86
Imagen 60. Ejemplo elemento br.	86
Imagen 61. Ejemplo elemento wbr.	87
Imagen 62. Ejemplo elemento ins.....	88
Imagen 63. Ejemplo elemento del.....	89
Imagen 64. Ejemplo elemento img.	91
Imagen 65. Ejemplo elemento iframe.	92
Imagen 66. Ejemplo elemento embed.....	93
Imagen 67. Ejemplo elemento object.....	95
Imagen 68. Ejemplo elemento param.....	96
Imagen 69. Ejemplo elemento video.....	98
Imagen 70. Ejemplo elemento audio.....	100
Imagen 71. Ejemplo elemento source.	101
Imagen 72. Ejemplo elemento track.....	102
Imagen 73. Ejemplo elemento canvas.....	103
Imagen 74. Ejemplo elemento map.....	104
Imagen 75. Ejemplo elemento area.	106
Imagen 76. Ejemplo elemento table.....	108
Imagen 77. Ejemplo elemento caption.....	109
Imagen 78. Ejemplo elemento colgroup.	110
Imagen 79. Ejemplo elemento tbody.	111
Imagen 80. Ejemplo elemento thead.....	112
Imagen 81. Ejemplo elemento tr.	114
Imagen 82. Ejemplo elemento td.	115
Imagen 83. Ejemplo elemento th.	116
Imagen 84. Ejemplo elemento form.....	118
Imagen 85. Ejemplo elemento fieldset.....	119
Imagen 86. Ejemplo elemento legend.....	119
Imagen 87. Ejemplo elemento label.....	121
Imagen 88. Ejemplo elemento input tipo hidden.	126
Imagen 89. Ejemplo elemento input tipo text.	127
Imagen 90. Ejemplo elemento input tipo search.....	128
Imagen 91. Ejemplo elemento input tipo tel.	128
Imagen 92. Ejemplo elemento input tipo url.....	129
Imagen 93. Ejemplo elemento input tipo email.	130
Imagen 94. Ejemplo elemento input tipo password.....	131

Imagen 95. Ejemplo elemento input tipo datetime.	132
Imagen 96. Ejemplo elemento input tipo date.	133
Imagen 97. Ejemplo elemento input tipo month.	133
Imagen 98. Ejemplo elemento tipo week.	134
Imagen 99. Ejemplo elemento input tipo time.	135
Imagen 100. Ejemplo elemento input datetime-local.	136
Imagen 101. Ejemplo elemento input tipo number.	137
Imagen 102. Ejemplo elemento input tipo range.	137
Imagen 103. Ejemplo elemento input tipo color.	138
Imagen 104. Ejemplo elemento tipo checkbox.	139
Imagen 105. Ejemplo elemento input tipo radio.	139
Imagen 106. Ejemplo elemento tipo file.	140
Imagen 107. Ejemplo elemento input tipo submit.	141
Imagen 108. Ejemplo elemento input tipo image.	142
Imagen 109. Ejemplo elemento input tipo reset.	142
Imagen 110. Ejemplo elemento input tipo button.	143
Imagen 111. Ejemplo elemento button.	144
Imagen 112. Ejemplo elemento select.	146
Imagen 113. Ejemplo elemento datalist.	147
Imagen 114. Ejemplo elemento optgroup.	148
Imagen 115. Ejemplo elemento option.	149
Imagen 116. Ejemplo elemento textarea.	151
Imagen 117. Ejemplo elemento keygen.	152
Imagen 118. Ejemplo elemento output.	153
Imagen 119. Ejemplo elemento progress.	153
Imagen 120. Ejemplo elemento meter.	155
Imagen 121. Ejemplo elemento details.	156
Imagen 122. Ejemplo elemento command.	158
Imagen 123. Ejemplo elemento menu.	159
Imagen 124. Documento ejemplo de un producto en MongoDB.	163
Imagen 125. Comparación de un documento MongoDB frente a un modelo relacional.	164
Imagen 126. Documento de MongoDB con documentos o arrays dentro.	165
Imagen 127. Ubicación de carpeta extraída de MongoDB en Windows.	167
Imagen 128. Moviendo la carpeta extraída de MongoDB a otra ubicación en Windows.	167
Imagen 129. Creando la carpeta para almacenar los archivos de MongoDB en Windows.	168
Imagen 130. Cambiar a otra ruta alternativa la ubicación para almacenar los archivos de MongoDB en Windows.	168
Imagen 131. Iniciar MongoDB en Windows.	168
Imagen 132. Conectar con MongoDB en Windows.	169
Imagen 133. Guardar y consultar un documento en la base de datos por defecto “test”.	169
Imagen 134. Creación de un directorio para archivos de registro de MongoDB en Windows.	170
Imagen 135. Creación de un archivo de configuración para el logpath en Windows.	170
Imagen 136. Instalar el servicio de MongoDB en Windows.	171
Imagen 137. Ejecutar el servicio de MongoDB en Windows.	171

Imagen 138. Detener el servicio de MongoDB en Windows.....	171
Imagen 139. Quitar el servicio de MongoDB en Windows.	171
Imagen 140. Descarga de MongoDB para Linux 64-bits.	172
Imagen 141. Descarga de MongoDB para Linux 32-bits.	172
Imagen 142. Extracción de los archivos descargados de MongoDB en Linux.....	172
Imagen 143. Creación de directorio para guardar los datos de MongoDB en Linux.	173
Imagen 144. Cambiar de propietario del de directorio donde se guardarán los datos de MongoDB en Linux.	173
Imagen 145. Iniciar “mongod” de MongoDB en Linux.	173
Imagen 146. Conectarse a la instancia de MongoDB en Linux.....	174
Imagen 147. Conexión MongoDB simple en PHP.	176
Imagen 148. Conexión MongoDB a un host remoto en PHP.	177
Imagen 149. Conexión MongoDB a un host remoto con puerto determinado en PHP	177
Imagen 150. Conexión MongoDB a un host remoto con puerto determinado y autenticación en PHP.....	177
Imagen 151. Seleccionando una base de datos en MongoDB desde PHP.....	178
Imagen 152. Seleccionando una colección de una base de datos en MongoDB desde PHP.	178
Imagen 153. Insertar un documento en una base de datos en MongoDB desde PHP.	179
Imagen 154. Documento insertado en MongoDB.....	179
Imagen 155. Insertar un documento con ID específica, en una base de datos en MongoDB desde PHP.	180
Imagen 157. Insertar un documento en una base de datos en MongoDB con documentos o arrays anidados desde PHP.....	180
Imagen 156. Documento insertado en MongoDB con una ID específico.....	180
Imagen 158. Documento insertado en MongoDB con documentos anidados.	181
Imagen 159. Actualizar un documento con el operador \$inc en MongoDB.....	182
Imagen 160. Actualizar un documento con el operador \$inc, consultando por el Objeto ID de MongoDB.....	183
Imagen 161. Actualizar un documento con el operador \$inc, en documentos anidados de MongoDB.....	183
Imagen 162. Explicación de un documento anidado de MongoDB.....	184
Imagen 163. Actualizar un documento anidado sin conocer la posición index de MongoDB.....	184
Imagen 164. Documento ejemplo para actualizarlo con \$set en MongoDB.....	185
Imagen 165. Actualizar documento con el operador \$set en MongoDB.	185
Imagen 166. Documento ejemplo para agregar un campo con \$set en MongoDB.	186
Imagen 167. Actualizar un documento agregando un nuevo campo con \$set en MongoDB.....	186
Imagen 168. Documento después de agregarse un campo nuevo con \$set en MongoDB.....	186
Imagen 169. Documento para borrar un campo con \$unset en MongoDB.....	187
Imagen 170. Ejemplo para Borrar un campo con \$unset en MongoDB.	187
Imagen 171. Documento después de haber borrado un campo con \$unset en MongoDB.....	187
Imagen 172. Documento ejemplo para añadir un elemento a un array con \$push en MongoDB.....	187

Imagen 173. Actualizar un documento con \$push en MongoDB.....	188
Imagen 174. Resultado de un documento actualizado con \$push en MongoDB....	188
Imagen 175. Actualizar un documento con \$pull en MongoDB.	189
Imagen 176. Resultado de actualizar un documento con \$pull en MongoDB.....	189
Imagen 177. Documento para actualizar mediante actualizaciones de reemplazo en MongoDB.....	189
Imagen 178. Actualizar un documento mediante actualizaciones de reemplazo en MongoDB.....	190
Imagen 179. Documento actualizado mediante actualizaciones de reemplazo en MongoDB.....	190
Imagen 180. Eliminar un documento en MongoDB.	190
Imagen 181. Eliminar una colección en MongoDB.....	191
Imagen 182. Eliminar una base de datos en MongoDB.....	191
Imagen 183. Documentos para realizar ejemplos de consultas en MongoDB.....	192
Imagen 184. Consultar un solo documento en MongoDB.....	192
Imagen 185. Mostrar datos resultantes de una consulta con findOne() en MongoDB.	193
Imagen 186. Consultar datos de todos los documentos de una colección en MongoDB.....	193
Imagen 187. Mostrar datos resultantes de una consulta con find() en MongoDB..	194
Imagen 188. Mostrar datos resultantes de una consulta con find() con una condición en MongoDB.	195
Imagen 189. Mostrar datos resultantes de una consulta con find(), sacando un campo determinado en MongoDB.....	196
Imagen 190. Consulta de datos con find(), sacando varios campos determinados en MongoDB.....	196
Imagen 191. Almacenando un archivo en MongoDB.....	198
Imagen 192. Consultando un archivo por el ID en MongoDB.	199
Imagen 193. Documento HTML5 para escoger una foto y almacenarla en MongoDB.	200
Imagen 194. Seleccionar una foto para almacenarla en MongoDB.....	200
Imagen 195. Almacenar y mostrar en pantalla una foto en MongoDB.	201
Imagen 196. Foto almacenada en la base de datos y mostrada en pantalla en MongoDB.....	201
Imagen 197. Actualizar un documento de manera segura con “safe” en MongoDB.	203
Imagen 198. Ejemplo de inicio de jquery.	205
Imagen 199. Ejemplo de interacción con JQuery.	205
Imagen 200. Ejemplo de animación simple JQuery.	205
Imagen 201. Ejemplo de efecto con JQuery UI.	206
Imagen 202. Ejemplo ordenamiento de una tabla con TableSorter.	206
Imagen 203. Ejemplo de uso para imprimir un DIV con PrintArea.	207
Imagen 204. Resultado en el navegador de la librería Keyboard.	208
Imagen 205. Ejemplo crear un botón grande en BootStrap.	209
Imagen 206. Ejemplo crear carrusel con BootStrap.....	210
Imagen 207. Ejemplo de manejo de carrusel de BootStrap desde JavaScript.	210
Imagen 208. Ejemplo API Audio y Video.	211
Imagen 209. Ejemplo elemento canvas para dibujar con el API Canvas.....	212
Imagen 210. Codificación inicial con el API Canvas.	212
Imagen 211. Obtener contexto para dibujar con API Canvas.....	213

Imagen 212. Dibujar un rectángulo o cuadrado con API Canvas.....	213
Imagen 213. Resultado de la creación de 2 rectángulos con sus medidas mediante la API Canvas.	214
Imagen 214. Dibujar rectángulos con color de relleno en la Api Canvas.....	215
Imagen 215. Resultado en el navegador de la creación de 2 rectángulos con relleno en el API Canvas.	215
Imagen 216. Dibujar 2 rombos con caminos en el API Canvas.	217
Imagen 217. Resultado de dibujar 2 rombos en el navegador con el API Canvas.	218
Imagen 218. Creación de una animación de movimiento de un rectángulo con el API Canvas. Parte 1.....	219
Imagen 219. Creación de una animación de movimiento de un rectángulo con el API Canvas. Parte 2.....	220
Imagen 220. Resultado en el navegador de la animación de movimiento de un rectángulo con el API Canvas.	221
Imagen 221. Sintaxis para obtener la ubicación (Latitud y Longitud) de navegación mediante la API Geolocation.	222
Imagen 222. Seguridad del navegador para conceder el permiso de mostrar la ubicación, mediante Geolocalización.....	223
Imagen 223. Sintaxis para generar un mapa con las coordenadas obtenidas de la API Geolocation, mediante Google Maps.....	223
Imagen 224. Resultado en el navegador de la generación de un mapa con Google Maps.....	224
Imagen 225. Sintaxis para almacenar datos con LocalStorage.	225
Imagen 226. Sintaxis para obtener los datos almacenados con LocalStorage.	225
Imagen 227. Sintaxis para eliminar una Llave/Valor, almacenada con LocalStorage.	226
Imagen 228. Sintaxis para abrir o crear una base de datos local con el API Web SQL Database.	226
Imagen 229. Sintaxis de la función de transacción de una base de datos local con la API Web SQL Database.....	227
Imagen 230. Sintaxis para crear una tabla de una base de datos local con la API Web SQL Database.....	227
Imagen 231. Sintaxis para insertar un registro en una tabla de una base de datos local con la API Web SQL Database.	228
Imagen 232. Sintaxis para consultar registros en una tabla de una base de datos local con la API Web SQL Database.	228
Imagen 233. Sintaxis para modificar registros en una tabla con la API Web SQL Database.	229
Imagen 234. Sintaxis para eliminar registros en una tabla de una base de datos local con la API Web SQL Database.	230
Imagen 235. Sintaxis para que un elemento HTML sea draggable.	230
Imagen 236. Sintaxis para comenzar a arrastrar un elemento HTML con Drag and Drop.	231
Imagen 237. Función en JavaScript para comenzar a arrastrar un elemento HTML con Drag and Drop.	231
Imagen 238. Elemento destino donde un elemento se arrastrará con Drag and Drop.	232
Imagen 239. Elemento destino con sus atributos, para poder soltar los elementos arrastrados con Drag and Drop.	232

Imagen 240. Función en JavaScript para obtener el elemento arrastrado con Drag and Drop.	233
Imagen 241. Ejemplo de cómo arrastrar una foto desde la PC al navegador mediante Drag and Drop.	234
Imagen 242. Sintaxis para crear un Web Worker.	235
Imagen 243. Sintaxis para enviar un mensaje a un Web Worker.	236
Imagen 244. Sintaxis para enviar un mensaje desde un Web Worker.	236
Imagen 245. Sintaxis para recibir un mensaje de un Web Worker.	236
Imagen 246. Sintaxis para recibir un mensaje en el Web Worker.	236
Imagen 247. Sintaxis para pedir permiso al usuario de mostrar las Notificaciones con la API Web Notifications.	237
Imagen 248. Seguridad del navegador para conceder el permiso de mostrar Notificaciones con la API Web Notifications.	238
Imagen 249. Sintaxis para crear Notificaciones con la API Web Notifications.	239
Imagen 250. Resultado en la PC de una Notificación con la API Web Notifications.	240
Imagen 251. Ejemplo de un servidor http con NodeJS.	241
Imagen 252. Página principal de la aplicación web.	243
Imagen 253. Sección Registrarse en la aplicación web.	244
Imagen 254. Mensaje de finalización de registro en la aplicación web.	246
Imagen 255. Sección Información sobre la aplicación.	247
Imagen 256. Descripción breve de las características de la aplicación web.	247
Imagen 257. Video de demostración de utilización de la aplicación web.	248
Imagen 258. Sección Iniciar sesión.	249
Imagen 259. Página principal de usuarios de la aplicación web.	250
Imagen 260. Sección Contenidos de la aplicación web.	251
Imagen 261. Menú superior de la aplicación web.	252
Imagen 262. Ejemplo de usuario "Juan" al revisar su perfil en la aplicación web.	253
Imagen 263. Ejemplo de utilización del Buscador de usuarios en la aplicación web.	253
Imagen 264. Ejemplo de búsqueda de usuarios, mostrando el cuadro de descripción del usuario en la aplicación web.	254
Imagen 265. Ejemplo para revisar el perfil de un usuario encontrado con el Buscador de usuarios en la aplicación web.	255
Imagen 266. Menú Izquierdo de la aplicación web.	256
Imagen 267. Foto de perfil actual inicial de la aplicación web.	257
Imagen 268. Función Noticias de la aplicación web.	258
Imagen 269. Función Mi cuenta, sección Datos personales de la aplicación web.	259
Imagen 270. Ejemplo de modificación de un campo en la función Mi cuenta, sección Datos personales de la aplicación web.	260
Imagen 271. Función Mi cuenta, sección Configuración de la aplicación web.	260
Imagen 272. Habilitar/Deshabilitar notificaciones de escritorio en la aplicación web.	261
Imagen 273. Permiso para notificaciones de escritorio en la aplicación web.	261
Imagen 274. Acceso a cambiar contraseña de usuario de la aplicación web.	261
Imagen 275. Ventana para cambiar la contraseña del usuario de la aplicación web.	262
Imagen 276. Acceso a eliminar cuenta de usuario de la aplicación web.	262
Imagen 277. Ventana para eliminar cuenta del usuario de la aplicación web.	263
Imagen 278. Función Personalización de la aplicación web.	264

Imagen 279. Función Personalización, tema personalizado de la aplicación web..	265
Imagen 280. Función Servicios de la aplicación web.	266
Imagen 281. Ejemplo de agregar un servicio con la opción de eliminarlo en la aplicación web.....	267
Imagen 282. Función Fotos de la aplicación web.	268
Imagen 283. Ejemplo de elegir una foto para agregar a la lista de fotos de un usuario en la aplicación web.	269
Imagen 284. Ejemplo del progreso de subida de una foto en la aplicación web. ...	269
Imagen 285. Ejemplo de foto ya agregada en la lista de fotos del usuario en la aplicación web.....	270
Imagen 286. Ejemplo de 2 fotos agregadas en la lista de fotos del usuario en la aplicación web.....	270
Imagen 287. Ejemplo de cómo cambiar la foto de perfil en la aplicación web.	271
Imagen 288. Ejemplo de cómo se cambió una foto de perfil del usuario en la aplicación web.....	271
Imagen 289. Función Contactos de la aplicación web.	272
Imagen 290. Función Mensajes de la aplicación web.....	273
Imagen 291. Ejemplo de revisión de mensajes del usuario en la aplicación web...	273
Imagen 292. Función Ubicación de la aplicación web.	274
Imagen 293. Seguridad en el navegador al momento de Actualizar la ubicación del usuario en la aplicación web.	274
Imagen 294. Ejemplo al momento de actualizar la ubicación del usuario en la aplicación web.....	275
Imagen 295. Permiso para que todos los usuarios vean la ubicación geográfica del usuario en la aplicación web.	275
Imagen 296. Función Currículum de la aplicación web.	276
Imagen 297. Sección Datos personales del currículum del usuario en la aplicación web.	277
Imagen 298. Sección Estudios del currículum del usuario en la aplicación web. ..	278
Imagen 299. Sección Educación primaria del currículum del usuario en la aplicación web.	278
Imagen 300. Sección Educación secundaria del currículum del usuario en la aplicación web.....	279
Imagen 301. Sección Educación superior del currículum del usuario en la aplicación web.	280
Imagen 302. Sección Postgrados del currículum del usuario en la aplicación web.	281
Imagen 303. Sección Méritos del currículum del usuario en la aplicación web.....	282
Imagen 304. Sección Experiencia del currículum del usuario en la aplicación web.	282
Imagen 305. Sección Adjuntos del currículum en la aplicación web.	283
Imagen 306. Sección Mis archivos adjuntos del currículum en la aplicación web.	283
Imagen 307. Ejemplo de los archivos del usuario en el currículum de la aplicación web.	284
Imagen 308. Sección Agregar más archivos del currículum del usuario en la aplicación web.....	284
Imagen 309. Ejemplo de 2 archivos listos para agregar al currículum del usuario en la aplicación web.....	285
Imagen 310. Función Entretenimiento del usuario en la aplicación web.	286
Imagen 311. Barra de información de la aplicación web.	286

Imagen 312. Ejemplo de la barra de información cuando el usuario edite su teléfono en la aplicación web.....	287
Imagen 313. Ejemplo de la barra de información cuando el usuario reciba notificaciones en la aplicación web.....	287
Imagen 314. Herramienta de chat del usuario en la aplicación web.....	287
Imagen 315. Ejemplo de un contacto desconectado en la aplicación web.....	288
Imagen 316. Ejemplo de un contacto conectado en la aplicación web.....	288
Imagen 317. Ejemplo para elegir un contacto y enviarle un mensaje de chat en la aplicación web.....	289
Imagen 318. Ejemplo de envío de un mensaje de chat a un contacto en la aplicación web.....	289
Imagen 319. Ejemplo de aviso cuando al usuario le llegue un mensaje de chat en la aplicación web.....	290
Imagen 320. Ejemplo de cómo leer un mensaje que le ha llegado al usuario en la aplicación web.....	290
Imagen 321. Ejemplo de notificación de escritorio en la aplicación web.....	290
Imagen 322. Perfil visible a los demás usuarios en la aplicación web.....	291
Imagen 323. Botón eliminar contacto en la aplicación web.....	292
Imagen 324. Botón agregar contacto en la aplicación web.....	292
Imagen 325. Foto de perfil en el perfil del usuario en la aplicación web.....	292
Imagen 326. Perfil del usuario, sección Perfil de la aplicación web.....	293
Imagen 327. Perfil del usuario, sección Ubicación de la aplicación web.....	294
Imagen 328. Perfil del usuario, sección Fotos de la aplicación web.....	294
Imagen 329. Perfil del usuario, sección Currículum de la aplicación web.....	295
Imagen 330. Menú del currículum dl usuario en la aplicación web.....	295
Imagen 331. Página de inicio del administrador de la aplicación web.....	296
Imagen 332. Zona de identificación del administrador de la aplicación web.....	297
Imagen 333. Identificación del administrador con el teclado virtual en la aplicación web.....	297
Imagen 334. Página principal del administrador de la aplicación web.....	298
Imagen 335. Sección para escoger el tipo de reporte que generará el administrador de la aplicación web.....	299
Imagen 336. Reporte por usuario para el administrador en la aplicación web.....	299
Imagen 337. Ejemplo de reporte generado por el administrador de las actividades de un usuario en específico en la aplicación web.....	300
Imagen 338. Reporte por fecha específica de usuarios para el administrador en la aplicación web.....	300
Imagen 339. Ejemplo de reporte generado por el administrador de las actividades de usuarios en una fecha específica en la aplicación web.....	301
Imagen 340. Reporte por mes y año específico de usuarios para el administrador en la aplicación web.....	301
Imagen 341. Ejemplo de reporte generado por el administrador de las actividades de usuarios en un mes y año específico en la aplicación web.....	302
Imagen 342. Reporte total de usuarios para el administrador en la aplicación web.....	302
Imagen 343. Ejemplo de reporte generado por el administrador de todas las actividades de usuarios en la aplicación web.....	303
Imagen 344. Sección Cantidad de usuarios registrados en el administrador de la aplicación web.....	303
Imagen 345. Botón para salir del administrador en la aplicación web.....	304

ÍNDICE DE TABLAS

Tabla 1. Compatibilidad HTML5 en navegadores de escritorio.....	22
Tabla 2. Compatibilidad HTML5 de navegadores en tabletas.....	22
Tabla 3. Compatibilidad HTML5 de navegadores en móviles.....	23
Tabla 4. Compatibilidad HTML5 de navegadores en juegos.....	23
Tabla 5. Compatibilidad HTML5 de navegadores en televisiones.....	24
Tabla 6. Atributos globales HTML5.....	29
Tabla 7. Atributos de eventos globales HTML5.....	32
Tabla 8. Simbología de compatibilidad de los navegadores para cada elemento HTML5.....	33
Tabla 9. Atributos elemento html.....	33
Tabla 10. Atributos elemento base.....	36
Tabla 11. Atributos elemento link.....	38
Tabla 12. Atributos elemento meta.....	39
Tabla 13. Atributos elemento style.....	40
Tabla 14. Atributos etiqueta script.....	41
Tabla 15. Atributos elemento body.....	45
Tabla 16. Atributos elemento blockquote.....	59
Tabla 17. Atributos elemento ol.....	60
Tabla 19. Atributos elemento a.....	68
Tabla 20. Atributos elemento q.....	73
Tabla 21. Atributos elemento time.....	75
Tabla 22. Atributos elemento ins.....	88
Tabla 23. Atributos elemento del.....	89
Tabla 24. Atributos elemento img.....	90
Tabla 25. Atributos elemento iframe.....	92
Tabla 26. Atributos elemento embed.....	93
Tabla 27. Atributos elemento object.....	94
Tabla 28. Atributos elemento param.....	95
Tabla 29. Atributos elemento video.....	98
Tabla 30. Atributos elemento audio.....	100
Tabla 31. Atributos elemento source.....	100
Tabla 32. Atributos elemento track.....	102
Tabla 33. Atributos elemento canvas.....	103
Tabla 34. Atributos elemento map.....	104
Tabla 35. Atributos elemento area.....	106
Tabla 36. Atributos elemento table.....	107
Tabla 37. Atributos elemento colgroup.....	109
Tabla 38. Atributos elemento col.....	110
Tabla 39. Atributos elemento td.....	114
Tabla 40. Atributos elemento th.....	115
Tabla 41. Atributos elemento form.....	117
Tabla 42. Atributos elemento fieldset.....	118
Tabla 43. Atributos elemento label.....	120
Tabla 44. Atributos elemento input.....	125

Tabla 45. Atributos elemento button.....	144
Tabla 46. Atributos elemento select.....	145
Tabla 47. Atributos elemento optgroup.	147
Tabla 48. Atributos elemento option.....	148
Tabla 49. Atributos elemento textarea.	150
Tabla 50. Atributos elemento keygen.	151
Tabla 51. Atributos elemento output.....	152
Tabla 52. Atributos elemento progress.	153
Tabla 53. Atributos elemento meter.....	154
Tabla 54. Atributos elemento details.	155
Tabla 55. Atributos elemento command.	157
Tabla 56. Atributos elemento menu.	158
Tabla 57. Lista de archivos o extensiones .dll para instalar el controlador de PHP para mongodb.....	175
Tabla 58. Correlación entre sentencias de consulta SQL y mongodb.....	197
Tabla 59. Diferentes tipos de botones en bootstrap.	209
Tabla 60. Resultado de un registro agregado en una tabla SQL desde la API Web SQL Database.....	229

INTRODUCCIÓN

En la actualidad, con el paso de los días el uso que se le daba al internet hizo que surgieran nuevos retos, la web ya era simplemente documentos de texto, las personas la utilizan en la actualidad para enviar mensajes de correo electrónico, chatear, jugar y hasta para ver televisión.

HTML ha ido actualizándose cada vez, llegando a su más reciente versión HTML5 que interactúa de una manera excelente entre páginas web y contenido media, de igual manera facilita la codificación en cuanto al diseño. Esta nueva versión de HTML resulta eficiente, por ejemplo mediante el uso del nuevo reproductor de video, que no requiere Flash o Adobe Player para reproducirse, simplemente requiere una etiqueta <video>.

Esta nueva versión es más ligera, sencilla y simple en su código, lo que permite que las páginas escritas en este lenguaje carguen más rápido en el navegador. Por tales motivos descritos anteriormente éste proyecto de tesis se basará en el lenguaje HTML5.

Desde algún tiempo atrás, “MySQL, una de las bases de datos más utilizadas, se ha venido dejando de un lado, en favor de una nueva alternativa como lo es NoSQL, porque las ventajas han sido grandes y resulta difícil no tomarlas en cuenta.”¹

NoSQL es nueva alternativa de manipulación de datos, que intentan mejorar las limitaciones del modelo relacional al momento de almacenar grandes cantidades de datos.

¹ DosIdeas. Leonardo De Seta. 9 de Julio de 2009. 25 de Mayo de 2012.
< <http://www.dosideas.com/noticias/base-de-datos/657-nosql-el-movimiento-en-contra-de-las-bases-de-datos.html>>.

El uso de las aplicaciones web es variado, existen aplicaciones para manejo de correo electrónico, ventas online, foros de discusión, juegos, redes sociales. Las redes sociales en internet se dedican a interactuar con varios usuarios, como es el caso de las redes de búsqueda de amistad o de compartir intereses, y muchas más.

Facebook una red social tan concurrente en la actualidad creó su almacén de datos NoSQL para llevar al cabo búsquedas en su sitio web, en vez de usar su base de datos MySQL existente. “De acuerdo al ingeniero de Facebook Avinash Lakshman, esa base de datos NoSQL puede escribir hasta 50GB de datos en disco en tan sólo 0.12 milisegundos, 2500 veces más rápido que MySQL”.²

El objetivo principal de este proyecto de tesis ha sido desarrollar una aplicación web sobre redes sociales, por lo que se ha utilizado MongoDB, una base de datos NoSQL que en comparación de las base de datos relacionales, resulta ágil, debido a que es orientada a documentos, que se almacenan en un formato BSON (versión modificada del JSON).

² [Static.last.fm](http://static.last.fm/johan/nosql-20090611/cassandra_nosql.ppt). Lakshman, Avinash. 11 de Junio de 2009. 29 de Abril de 2012
<http://static.last.fm/johan/nosql-20090611/cassandra_nosql.ppt>.

CAPÍTULO 1

HTML 5

1.1 Introducción

En este capítulo se desarrollará un manual de la nueva versión HTML5, tratando así sus nuevas etiquetas y su nueva estructura de un documento web.

W3C (World Wide Web Consortium) es el consorcio encargado del progreso de la especificación HTML.

1.2 ¿Qué es HTML5?

HTML5 es la actualización o la quinta revisión del lenguaje de hipertexto de la World Wide Web (WWW), HTML, el lenguaje en el que son creadas las páginas web.

HTML5 añade a sus etiquetas semántica, etiquetas que permiten describir cual es el significado, importancia o finalidad del contenido, pero que no producen ningún aspecto especial en la visualización, si no que a más de darle sentido a las etiquetas, están orientadas a buscadores web.

El lenguaje HTML, permaneció sin cambios desde su última revisión en el año 1999, quedando obsoleto en muchos aspectos. Entonces, por el año 2004, los principales desarrolladores de navegadores (Apple, Opera, Mozilla, entre otros) iniciaron un grupo, el cual estaba interesado en la evolución de la web, denominado WHATWG (Grupo de Trabajo de Tecnologías de Aplicaciones Web de Hipertexto). Como no podía ser de otra manera el W3C (World Wide Web) se unió a la iniciativa en 2006 y en el 2009 nace por fin HTML5.

Aprender HTML5 no es complicado, la manera de trabajar es la misma que con anteriores versiones, el uso de las etiquetas y atributos es idéntico, y muchas de las etiquetas se mantienen. Para sacarle el máximo provecho se podría manejar tanto

HTML5 con CSS3 y las APIs de JavaScript, tres tecnologías que asociándolas, darán como resultado una aplicación web muy atractiva.

1.3 Compatibilidad en los navegadores

Pueden existir problemas de compatibilidad con los navegadores que no soportan aún el 100% de HTML5, por lo tanto, antes de decidir implementar algo, es aconsejable informarse de la compatibilidad en los navegadores. Es cuestión de tiempo que HTML5 se convierta en un estándar y los navegadores permitan la utilización de todas sus etiquetas y funcionalidades que se pueden lograr en conjunto con las APIs de JavaScript y CSS3, como también los usuarios deben saber que mantener sus navegadores actualizados es primordial para disfrutar de HTML5.

Existen páginas web en donde se puede consultar de manera actualizada la compatibilidad de cada uno de los navegadores o browsers, ya sean de escritorio, tabletas, móviles, juegos o televisión. Una de las páginas más interesantes es www.html5test.com, en donde se puede consultar la compatibilidad de los navegadores actuales e inclusive compararlos con otros. Si un navegador pasa todas las pruebas, HTML5Test dará una puntuación máxima de 500 puntos, caso contrario dará los puntos correspondientes a su compatibilidad de funcionalidades HTML5 actuales.

A continuación se muestran tablas de los navegadores con las puntuaciones dadas por HTML5Test actualmente (Noviembre del 2012), siendo 500 la puntuación máxima:

1.3.1 Navegadores más comunes de Escritorio

Nombre	Puntos /500
Maxthon 3.4.5	457
Chrome 22	434
Opera 12.00	389
Safari 6.0	378
Firefox 16	372
Internet Explorer 10	320

Tabla 1. Compatibilidad HTML5 en navegadores de escritorio.
Fuente: THE HTML5 TEST. [Disponible en: www.html5test.com/results/desktop.html]

1.3.2 Navegadores más comunes en Tabletas (Tablets)

Nombre	Puntos /500
RIM Tablet OS 2.1	411
Opera Mobile 12.10	406
Chrome	390
Firefox Mobile 16	388
iOs 6.0	386
Silk 2.2	358
Internet Explorer 10	320
Android 4.0	297
webOS 3.0	224

Tabla 2. Compatibilidad HTML5 de navegadores en tabletas.
Fuente: THE HTML5 TEST. [Disponible en: www.html5test.com/results/tablet.html]

1.3.3 Navegadores más comunes en Móviles (Mobiles)

Nombre	Puntos /500
Opera Mobile 12.10	406
Chrome	390
Firefox Mobile 16	388
iOs 6.0	386
Windows Phone 8	320
Android 4.0	297
BlackBerry OS 7	288
Bada 2.0	283
Nokia Belle FP 1	252
Android 2.3	200

Tabla 3. Compatibilidad HTML5 de navegadores en móviles.

Fuente: THE HTML5 TEST. [Disponible en: www.html5test.com/results/mobile.html]

1.3.4 Navegadores más comunes en Juegos (Gaming)

Nombre del Juego	Nombre del navegador	Puntos /500
Xbox 360	Internet Explorer 9	120
Nintendo 3DS	NetFront	109
Nintendo Wii	Opera	94
Nintendo DSi	Opera	94
Sony Play Station Vita	NetFront	58

Tabla 4. Compatibilidad HTML5 de navegadores en juegos.

Fuente: THE HTML5 TEST. [Disponible en: www.html5test.com/results/gaming.html]

1.3.5 Navegadores más comunes en Televisiones

Nombre	Puntos /500
Sony Internet TV	342
Google TV	341
LG NetCast 2012	338
Panasonic Smart Viera	240
Boxee	222

Tabla 5. Compatibilidad HTML5 de navegadores en televisiones.

Fuente: THE HTML5 TEST. [Disponible en: www.html5test.com/results/television.html]

En la aplicación web desarrollada en este proyecto de tesis se ha utilizado el navegador Google Chrome 22 de escritorio, pues es uno de los navegadores con mayor compatibilidad y común en el medio, además es capaz de demostrar la mayoría de novedades de HTML con su nueva versión 5.

1.4 Sintaxis HTML5

1.4.1 La Declaración del tipo de documento (DOCTYPE)

Un DOCTYPE es un preámbulo necesario.

Los DOCTYPEs en los documentos HTML5 son necesarios por razones de compatibilidad. Cuando se omite, los navegadores tienden a utilizar un modo de representación distinta que sería incompatible con algunas especificaciones.

Incluyendo el DOCTYPE en un documento asegura que el navegador se rija a seguir las especificaciones pertinentes.

Un DOCTYPE normal debe constar de los siguientes componentes, en este orden:

- Una cadena “<!DOCTYPE”.
- Uno o más espacios.
- Una cadena “html”.

- Opcionalmente, uno o más espacios.
- Un carácter “>”.

En otras palabras, el siguiente ejemplo muestra un DOCTYPE:

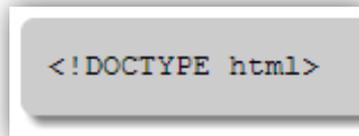


Imagen 1. DOCTYPE HTML5

Fuente: Consorcio W3C [Disponible en:
<http://dev.w3.org/html5/markup/syntax.html#doctype-syntax>]

A continuación se muestra una representación de un documento basado en el DOM (Modelo de Objetos del Documento) de la W3C, en el que se muestra un árbol que incluye los elementos: “html”, “head” y “body”, que se utilizaban normalmente en versiones anteriores de HTML5 y que se mantienen en su forma, donde sus etiquetas de inicio y fin se encuentran de manera implícita en el documento:

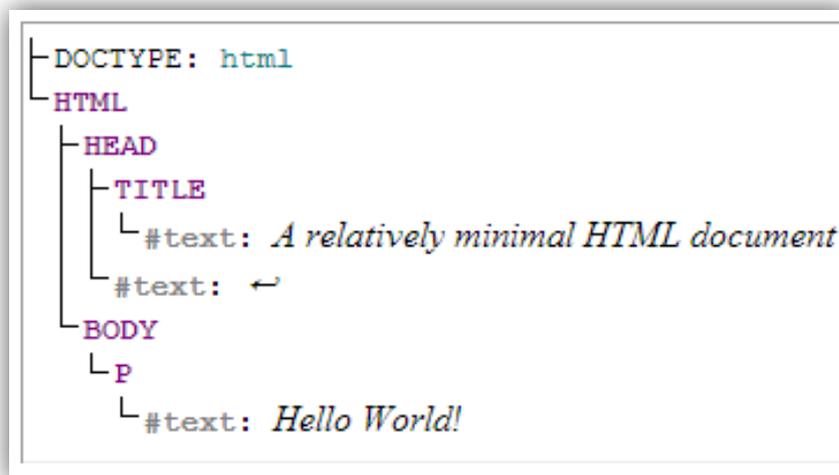


Imagen 2. Árbol DOM HTML5

Fuente: Consorcio W3C [Disponible en:
<http://dev.w3.org/html5/markup/documents.html#language-and-syntax>]

1.4.2 ¿Qué es un elemento?

Los elementos son la estructura básica de HTML, tienen dos propiedades básicas: atributos y contenido. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo, </nombre-

de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo, <nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>).

El contenido de un elemento son algunos elementos, datos de caracteres, y comentarios que contenga. Los atributos y sus valores no son considerados como el "contenido" de un elemento.

Un elemento vacío (void) es un elemento que no tiene contenido bajo ninguna circunstancia. Los elementos vacíos pueden tener atributos.

La siguiente es una lista completa de los elementos vacíos en HTML:

- area, base, br, col, command, embed, hr, img, input, keygen, link, meta, param, source, track, wbr.

En la siguiente lista se describen las reglas de sintaxis de los elementos.

- Las etiquetas se utilizan para delimitar el inicio y el final de los elementos en el marcado. Los elementos tienen una etiqueta de inicio para indicar dónde comenzar. Los elementos vacíos (void) no tienen una etiqueta de cierre para indicar dónde terminan.
- Los nombres de etiquetas se utilizan en las etiquetas de inicio y fin del elemento para indicar el nombre del elemento. Los elementos HTML tienen nombres que solamente usan caracteres en el rango de 0-9, a-z, y A-Z.
- Las etiquetas de inicio están compuestas de las siguientes partes, exactamente en el siguiente orden:
 - Un símbolo "<".
 - El nombre de etiqueta del elemento.
 - Opcionalmente, uno o más atributos, cada uno de los cuales debe ser precedido por uno o más caracteres de espacio.
 - Opcionalmente, uno o más caracteres de espacio.

- Opcionalmente, un carácter "/", que puede estar presente sólo si el elemento es un elemento vacío (void).
 - Un carácter ">".
- Las etiquetas finales están compuestas de las siguientes partes, exactamente en el siguiente orden:
 - Un símbolo "<".
 - A "/" carácter
 - El nombre de etiqueta del elemento.
 - Opcionalmente, uno o más caracteres de espacio.
 - Un carácter ">".

Los elementos vacíos (void) sólo tienen una etiqueta de inicio, las etiquetas de cierre no debe especificarse para los elementos vacíos.

1.4.3 ¿Qué son los atributos?

Los atributos son parte de un elemento y se expresan dentro de la etiqueta inicial del elemento. Los atributos tienen un nombre y un valor.

Nunca debe haber dos o más atributos con el mismo nombre en la misma etiqueta de inicio, cuyos nombres son insensibles a mayúscula y minúsculas.

Los nombres de los atributos deben constar de uno o más caracteres distintos de los caracteres de espacio, U +0000 NULL, "", "'", ">", "/", "=", los caracteres de control, y cualquier carácter que no sea definido por Unicode.

A continuación una imagen en resumen de una estructura de los elementos HTML, con sus etiquetas de inicio y final, su contenido y sus atributos:

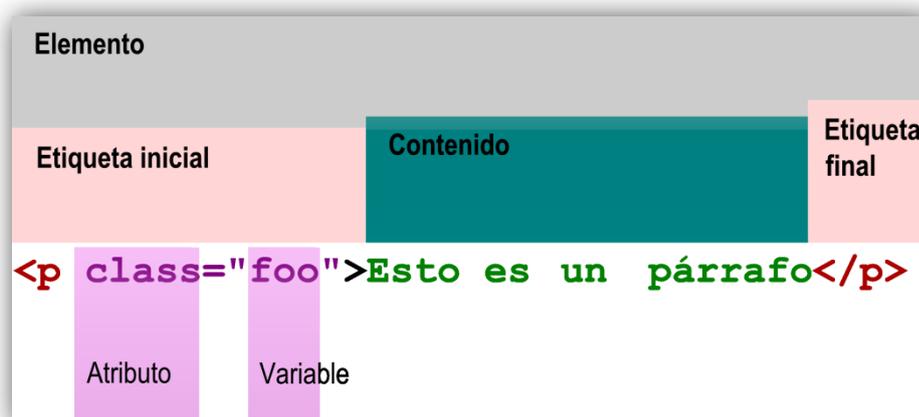


Imagen 3. Estructura de los Elementos HTML

Fuente: Consorcio W3C [Disponible en:

<http://dev.w3.org/html5/markingup/documents.html#language-and-syntax>]

1.4.3.1 Atributos globales

A continuación se muestra una lista de los atributos globales, es decir, que son considerados comunes a todos los elementos en el lenguaje HTML. Se dividen en 2 grupos: Atributos esenciales y Atributos de controlador de eventos

1.4.3.1.1 Atributos esenciales

- accesskey, class, contenteditable, contextmenu, dir, draggable, dropzone, hidden, id, lang, spellcheck, style, tabindex, title, translate.

Atributo	Descripción	Valores
accesskey	Especifica una tecla de acceso directo para activar/hacer focus en un elemento.	Algún carácter
Class	Especifica uno o más nombres de las clases de un elemento (se refiere a una clase en una hoja de estilo).	Nombre de clase
contenteditable	Especifica si el contenido de un elemento es editable o no.	true, false, inherit
contextmenu	Especifica un menú contextual para un elemento. El menú contextual aparece cuando un usuario hace clic en el elemento.	ID del elemento <menu>
Dir	Especifica la dirección del texto para el contenido de un elemento.	ltr, rtl, auto
draggable	Especifica si un elemento es arrastrable o no.	true, false, auto

dropzone	Especifica si los datos arrastrados son copiados, movidos o vinculados, cuando se dejan caer sobre un elemento.	copy, move, link
hidden	Especifica que un elemento no es visible aún, o no es, relevante.	Cadena vacía, sin valor, hidden
Id	Especifica un único 'id' a un elemento.	Nombre ID
Lang	Especifica el idioma del contenido del elemento.	Código de lenguaje ISO
spellcheck	Especifica si el elemento debe tener su ortografía y gramática comprobada o no.	true, false
Style	Especifica una línea de estilo CSS para un elemento.	Una o más propiedades de CSS
TabIndex	Especifica el orden de tabulación de un elemento.	Un número
Title	Especifica información adicional acerca de un elemento.	Cadena de texto
Translate	Se utiliza para especificar si los valores de un elemento y los valores de sus nodos hijos de texto se traducen cuando la página está traducándose, o si se dejan sin cambios.	yes, no

Tabla 6. Atributos globales HTML5

Fuente: W3Schools [Disponible en:

http://www.w3schools.com/tags/ref_standardattributes.asp]

1.4.3.1.2 Atributos de eventos

- onabort, onblur, oncancel, oncanplay, oncanplaythrough, onchange, onclick, onclose, oncontextmenu, oncuechange, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, ondurationchange, onemptied, onended, onerror, onfocus, oninput, oninvalid, onkeydown, onkeypress, onkeyup, onload, onloadeddata, onloadedmetadata, onloadstart, onmousedown, onmousemove, onmouseout, onmouseover, onmouseup, onmousewheel, onpause, onplay, onplaying, onprogress, onratechange, onreset, onscroll, onseeked, onseeking, onselect, onshow, onstalled, onsubmit, onsuspend, ontimeupdate, onvolumechange, onwaiting.

Atributo evento	Se ejecuta cuando...
Onabort	...el usuario detiene la obtención de datos media en descarga. Pero no debido a un error.
Onblur	...la ventana pierde el enfoque (focus).
Oncanplay	... un archivo está listo para reproducirse (cuando tenga suficiente almacenamiento temporal para empezar)
oncanplaythrough	... un archivo puede reproducirse todo su camino hasta el final, sin pausas para el almacenamiento temporal.
Onchange	...un elemento cambia.
Onclick	...se da click en el ratón sobre un elemento.
Oncontextmenu	...un menú contextual se activa.
Oncuechange	...cambia una pista (cue) de un elemento media.
Ondblclick	...se da doble click sobre un elemento.
Ondrag	...un elemento se arrastra.
Ondragend	...finaliza una operación de arrastre sobre un elemento.
Ondragenter	...un elemento ha sido arrastrado a un destino válido.
Ondragleave	...un elemento deja un destino válido.
Ondragover	...un elemento está siendo arrastrado sobre un destino válido.
Ondragstart	...se inicia en un elemento la operación de arrastre.
Ondrop	...un elemento arrastrado y soltado en un destino válido.
ondurationchange	...la longitud de los elementos media cambia.
Onemptied	...algo malo sucede en los elementos media y su archivo de repente no está disponible (se pierde la conexión inesperadamente).
Onended	...el archivo del elemento media llega a su final (evento útil para mostrar mensajes como “Gracias por escucharme”).
Onerror	...se produce un error cuando el archivo media está siendo cargado.
Onfocus	...un elemento recibe el enfoque (focus).
Oninput	...un elemento recibe datos introducidos por el usuario.
Oninvalid	...un elemento es inválido.
Onkeydown	...un usuario mantiene presionada una tecla.

Onkeypress	...un usuario pulsa una tecla.
Onkeyup	...un usuario suelta una tecla.
Onload	...termina de cargarse completamente una página web.
Onloadeddata	...se carga la información o datos de un elemento media.
onloadedmetadata	...los metadatos (como dimensiones y duración) de un elemento media se cargan.
Onloadstart	...el archivo del elemento media empieza a cargarse, antes de que algo esté actualmente cargado.
Onmousedown	...el botón del ratón se presiona hacia abajo sobre un elemento.
Onmousemove	...el puntero del ratón se mueve sobre un elemento.
Onmouseout	...el puntero de ratón se mueve fuera de un elemento.
Onmouseover	...el puntero del ratón se sitúa sobre un elemento.
Onmouseup	...el botón del mouse se suelta cuando está sobre un elemento.
Onmousewheel	...la rueda del ratón está girando.
Onpause	...cuando un elemento media está en pausa, ya sea provocado por el usuario o mediante programación.
Onplay	...un elemento media está listo para empezar a reproducirse.
Onplaying	...un elemento media actualmente han comenzado a reproducirse.
Onprogress	...el navegador está en proceso de obtención de los datos media.
Onratechange	...se cambia cada vez la velocidad de reproducción de un elemento media.
Onreset	...se hace click sobre el botón tipo reset de un formulario.
Onscroll	...la barra de desplazamiento de un elemento se está desplazando o en movimiento.
Onseeked	...se completa una operación de búsqueda en un elemento media.
Onseeking	...comienza una operación de búsqueda en un elemento media.
Onselect	...un usuario selecciona texto de un campo de texto cualquiera.
onstalled	...el navegador no es capaz de recuperar los datos media de

	un elemento media, por alguna razón.
Onsubmit	...se envía un formulario.
Onsuspend	...la búsqueda de datos media en un elemento media se detiene antes de que esté completamente cargado, por cualquier razón.
Ontimeupdate	...la posición de reproducción de un elemento media cambia (como cuando un usuario avanza rápidamente hacia un punto diferente en datos media).
onvolumechange	...se cambia el volumen de un elemento media (incluye también el ajuste de volumen a “mute”).
Onwaiting	...un elemento media ha detenido su reproducción, pero espera que se reanude.

Tabla 7. Atributos de eventos globales HTML5

Cabe recalcar que los atributos de la tabla 7, se aplican a todos los elementos, pero no son útiles en todos los elementos, por ejemplo solo los elementos media recibirán un evento “onvolumechange” por el usuario.

1.5 Elementos HTML5.

A continuación se detallan cada uno de los elementos que contiene HTML5, con sus respectivos atributos, no hay que olvidar que la mayoría de los elementos de versiones anteriores se mantienen.

Es importante mencionar que aparecen nuevos elementos para HTML5 y cada uno de estos difiere en compatibilidad con los navegadores web que existen, para ello se ha tomado como referencia cinco de los navegadores más comunes (Internet Explorer, Mozilla Firefox, Opera, Google Chrome, Safari) para mostrar en cada elemento HTML5 su existencia de compatibilidad o no.

A continuación se puede observar la nomenclatura que se usará para descubrir si un elemento HTML5 es compatible o no con los navegadores:

Nombre del navegador	Simbología Compatible	Simbología No Compatible
Internet Explorer 9		
Mozilla Firefox 17.0		
Opera 12.11		
Google Chrome 22		
Safari 5.34		

Tabla 8. Simbología de compatibilidad de los navegadores para cada elemento HTML5.

1.5.1 El elemento raíz

1.5.1.1 Html

Definición:

El elemento html representa la raíz de un documento HTML.

Los autores pueden especificar el atributo lang en el elemento raíz html, dando el idioma del documento.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Manifest	El atributo manifest da la dirección de manifiesto del documento de solicitud de caché, si es que existe	URL

Tabla 9. Atributos elemento html.

Compatible en los navegadores:



Ejemplo:

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Swapping Songs</title>
</head>
<body>
<h1>Swapping Songs</h1>
<p>Tonight I swapped some of the songs I wrote with some friends, who
gave me some of the songs they wrote. I love sharing my music.</p>
</body>
</html>
```

Imagen 4. Ejemplo elemento html.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-html-element>]

1.5.2 Documento de metadatos

1.5.2.1 Head

Definición:

El elemento head representa una colección de metadatos para el documento.

La colección de metadatos en un elemento head puede ser grande o pequeña.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <META CHARSET="UTF-8">
    <BASE HREF="http://www.example.com/">
    <TITLE>An application with a long head</TITLE>
    <LINK REL="STYLESHEET" HREF="default.css">
    <LINK REL="STYLESHEET ALTERNATE" HREF="big.css" TITLE="Big Text">
    <SCRIPT SRC="support.js"></SCRIPT>
    <META NAME="APPLICATION-NAME" CONTENT="Long headed application">
  </HEAD>
  <BODY>
    ...
```

Imagen 5. Ejemplo elemento head.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-head-element>]

1.5.2.2 Title

Definición:

El elemento de título representa el título del documento o el nombre. Los autores deberían utilizar títulos que identifican sus documentos, incluso cuando se utilizan fuera de contexto.

No debe haber más de un elemento de título por documento.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<title>Dances used during bee mating rituals</title>
...
<h1>The Dances</h1>
```

Imagen 6. Ejemplo elemento title.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-title-element>]

1.5.2.3 Base

Definición:

Define una URL base para todos los enlaces en una página

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Href	Especifica la URL base para todas las URL relativa en la página	URL
Target	Especifica el destino predeterminado para todos los hipervínculos y formularios en la página	_blank, _parent, _self, _top, Nombre de frame.

Tabla 10. Atributos elemento base.

Compatible en los navegadores:



Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is an example for the <code><base></code> element</title>
    <base href="http://www.example.com/news/index.html">
  </head>
  <body>
    <p>Visit the <a href="archives.html">archives</a>.</p>
  </body>
</html>
```

Imagen 7. Ejemplo elemento base.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-base-element>]

1.5.2.4 Link

Definición:

El elemento de enlace permite a los autores vincular su documento a otros recursos.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Href	Especifica la ubicación del documento vinculado	URL
Rel	Obligatorio. Especifica la relación entre el documento actual y el documento vinculado	alternate archives author bookmark external first help icon last license next nofollow noreferrer

		pingback prefetch prev search sidebar stylesheet tag up
Media	Especifica con qué dispositivo el documento vinculado se mostrará.	(<i>media_query</i>)
Hreflang	Especifica el idioma del texto del documento vinculado	Código de lenguaje (<i>lenguaje_code</i>).
Type	Especifica el tipo MIME del documento vinculado	Tipo MIME (<i>MIME_Type</i>).
Sizes	Especifica el tamaño del recurso vinculado. Sólo para rel = "icon"	Algún alto x ancho.

Tabla 11. Atributos elemento link.

Compatible en los navegadores:



Ejemplo:

```
<link rel=alternate href="/en/html" hreflang=en type=text/html title="English HTML">
<link rel=alternate href="/fr/html" hreflang=fr type=text/html title="French HTML">
```

Imagen 8. Ejemplo elemento link.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-link-element>]

1.5.2.5 Meta

Definición:

El elemento meta representa varios tipos de metadatos que no se pueden expresar con los elementos: “title”, “base”, “link”, “style”, y “script”.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Name	Especifica un nombre para los metadatos.	application-name, autor, description, generator, keywords.
http-equiv	Proporciona una cabecera HTTP de la información/valor del atributo content.	content-type, default-style, refresh.
Content	Da el valor asociado con el atributo http-equiv o el atributo name.	Texto
Charset	Especifica la codificación de caracteres del documento HTML	codificación de caracteres (<i>character_set</i>)

Tabla 12. Atributos elemento meta.

Compatible en los navegadores:



Ejemplo:

```
<meta name=generator content="Frontweaver 8.2">
```

Imagen 9. Ejemplo elemento meta.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-meta-element>]

1.5.2.6 Style

Definición:

El elemento style permite a los autores insertar información de estilo en sus documentos. El elemento style es una de varias entradas en el modelo de procesamiento de estilo. El elemento no representa contenido para el usuario.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Media	Especifica qué medio/dispositivo del recurso multimedia está optimizado para aquello.	(media_query).
scoped	Especifica que los estilos se aplican únicamente al elemento padre de este elemento y los elementos de ese elemento hijo.	Scoped
Type	Especifica el tipo MIME de la hoja de estilo	text/css

Tabla 13. Atributos elemento style.

Compatible en los navegadores:



Ejemplo:

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <title>My favorite book</title>
    <style>
      body { color: black; background: white; }
      em { font-style: normal; color: red; }
    </style>
  </head>
  <body>
    <p>My <em>favorite</em> book of all time has <em>got</em> to be
    <cite>A Cat's Life</cite>. It is a book by P. Rahmel that talks
    about the <i lang="la">Felis Catus</i> in modern human society.</p>
  </body>
</html>
```

Imagen 10. Ejemplo elemento style.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-style-element>]

1.5.3 Scripting

1.5.3.1 Script

Definición:

El elemento de script permite a los autores incluir secuencias de comandos dinámicos y bloques de datos en sus documentos. El elemento no representa contenido para el usuario.

Cuando se usa para incluir scripts dinámicos, las secuencias de comandos o bien pueden ser embebidos en línea o se pueden importar desde un archivo externo mediante el atributo src. Si el lenguaje no es "text/javascript", entonces el atributo type debe estar presente. Sea cual sea el lenguaje utilizado, el contenido del elemento script debe cumplir con los requisitos de la especificación de lenguaje.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Src	Especifica la dirección URL de un archivo de script externo	URL
Async	Especifica que el script se ejecuta de forma asíncrona (sólo para los scripts externos)	Async
Defer	Especifica que el script se ejecuta cuando la página ha terminado de analizar (sólo para los scripts externos)	Defer
Type	Especifica el tipo MIME de la secuencia de comandos (script)	Tipo de MIME (<i>MIME_type</i>)
charset	Especifica la codificación de caracteres utilizado en un archivo de script externo	Codificación de caracteres (<i>charset</i>).

Tabla 14. Atributos etiqueta script.

Compatible en los navegadores:



Ejemplo:

```
<script src="game-engine.js"></script>
<script type="text/x-game-map">
.....U.....e
o.....A.....e
.....A.....AAA.....e
.A..AAA...AAAAA...e
</script>
```

Imagen 11. Ejemplo elemento script.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-script-element>]

1.5.3.2 Noscript

Definición:

Elemento que define una sección sin scripts. El elemento no representa nada si el “scripting” está habilitado, pero si no está habilitado muestra su contenido.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<form action="calcSquare.php">
  <p>
    <label for=x>Number</label>:
    <input id="x" name="x" type="number">
  </p>
  <script>
    var x = document.getElementById('x');
    var output = document.createElement('p');
    output.textContent = 'Type a number; it will be squared right then!'
    x.form.appendChild(output);
    x.form.onsubmit = function () { return false; }
    x.oninput = function () {
      var v = x.valueAsNumber;
      output.textContent = v + ' squared is ' + v * v;
    };
  </script>
  <noscript>
    <input type=submit value="Calculate Square">
  </noscript>
</form>
```

Imagen 12. Ejemplo elemento noscript.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-noscript-element>]

1.5.4 Secciones

1.5.4.1 Body

Definición:

El elemento **body** representa el contenido principal del documento.

Conforme a los documentos, sólo hay un elemento **body**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Onafterprint	Evento que se ejecuta después de que el documento haya acabado de imprimirse	Script
onbeforeprint	Evento que se ejecuta antes de que el documento haya empezado a imprimirse	Script
Onbeforeunload	Evento que se ejecuta antes de que el usuario abandone el documento (la página se descarga).	Script
Onblur	Evento que se ejecuta cuando la ventana del navegador pierde el foco o enfoque (focus).	Script
Onerror	Evento que se ejecuta cuando el documento no se ha cargado correctamente.	Script
Onfocus	Evento que se ejecuta cuando la ventana del navegador queda seleccionada, recibe el foco.	Script
Onhashchange	Evento que se ejecuta cuando el usuario navega a una página anterior en el historial de navegación.	Script
Onload	Evento que se ejecuta cuando se carga el documento en la ventana del navegador.	Script
Onmessage	Evento que se ejecuta cuando se muestra un mensaje.	Script
Onoffline	Evento que se ejecuta cuando la página se desconecta.	Script
Online	Evento que se ejecuta cuando la página se pone en línea.	Script
Onpagehide	Evento que se ejecuta cuando el navegador deja una página almacenada en el historial de navegación.	Script
Onpageshow	Evento que se ejecuta cuando el	Script

	navegador entra en una página almacenada en el historial de navegación.	
Onpopstate	Evento que se ejecuta cuando cambia el historial de la ventana del navegador.	Script
Onresize	Evento que se ejecuta cuando la ventana cambia de tamaño.	Script
Onscroll	Evento que se ejecuta cuando un documento en la ventana del navegador se está desplazando.	Script
Onstorage	Evento que se ejecuta cuando el almacenamiento se activa.	Script
Onunload	Evento que se ejecuta cuando un usuario abandona la página.	Script

Tabla 15. Atributos elemento body.

Compatible en los navegadores:



Ejemplo:

```

<!DOCTYPE HTML>
<html>
<head>
<title>Online or offline?</title>
<script>
function update(online) {
document.getElementById('status').textContent =
online ? 'Online' : 'Offline';
}
</script>
</head>
<body ononline="update(true)"
onoffline="update(false)"
onload="update(navigator.onLine)">
<p>You are: <span id="status">(Unknown)</span></p>
</body>
</html>

```

Imagen 13. Ejemplo elemento body.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-body-element>]

1.5.4.2 Article

Definición:

El elemento article representa una composición auto-contenido en un documento, página, aplicación, o en el sitio y esto es, en principio, independientemente distribuible o reutilizable, por ejemplo podría ser un mensaje en el foro, un artículo de una revista o de un periódico, una entrada de blog, un comentario enviado por el usuario, un widget interactivo o gadget, o cualquier otro elemento independiente del contenido.

Cuando los elementos del artículo están anidados, los elementos del artículo interiores representan artículos que están en principio en relación con el contenido del artículo exterior. Por ejemplo, una entrada de blog en un sitio que acepta el envío de comentarios de usuarios, podrían representar los comentarios los elementos del artículo anidados en el elemento de artículo para la entrada de blog.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<article>
<header>
  <h1>The Very First Rule of Life</h1>
  <p><time datetime="2009-10-09">3 days ago</time></p>
</header>
<p>If there's a microphone anywhere near you, assume it's hot and
sending whatever you're saying to the world. Seriously.</p>
<p>...</p>
<section>
  <h1>Comments</h1>
  <article id="c1">
    <footer>
      <p>Posted by: George Washington</p>
      <p><time datetime="2009-10-10">15 minutes ago</time></p>
    </footer>
    <p>Yeah! Especially when talking about your lobbyist friends!</p>
  </article>
  <article id="c2">
    <footer>
      <p>Posted by: George Hammond</p>
      <p><time datetime="2009-10-10">5 minutes ago</time></p>
    </footer>
    <p>Hey, you have the same first name as me.</p>
  </article>
</section>
</article>
```

Imagen 14. Ejemplo elemento article.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-article-element>]

1.5.4.3 Section

Definición:

El elemento de sección representa una sección genérica de un documento o aplicación. Una sección, en este contexto, es una agrupación temática de los contenidos, por lo general con un título.

Ejemplos de secciones sería capítulos, las diferentes páginas con fichas en un cuadro de diálogo con fichas, o las secciones numeradas de una tesis. Un sitio Web en su página de inicio se puede dividir en secciones para una introducción, noticias e información de contacto.

Los autores pueden usar el elemento `article` en lugar del elemento `section` cuando tenga sentido syndicar el contenido del elemento.

El elemento `section` no es un elemento contenedor genérico. Cuando un elemento es necesario sólo para fines de estilo (styling) o para los scripts, a los autores se les aconseja utilizar el elemento `div` en su lugar. Una regla general es que el elemento `section` es apropiado sólo si los contenidos del elemento se enumeran explícitamente en el esquema del documento.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<article>
  <hgroup>
    <h1>Apples</h1>
    <h2>Tasty, delicious fruit!</h2>
  </hgroup>
  <p>The apple is the pomaceous fruit of the apple tree.</p>
  <section>
    <h1>Red Delicious</h1>
    <p>These bright red apples are the most common found in many
    supermarkets.</p>
  </section>
  <section>
    <h1>Granny Smith</h1>
    <p>These juicy, green apples make a great filling for
    apple pies.</p>
  </section>
</article>
```

Imagen 15. Ejemplo elemento `section`.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-section-element>]

1.5.4.4 Nav

Definición:

El elemento nav representa una sección de una página que enlaza a otras páginas o a partes dentro de la página: una sección con enlaces de navegación.

No todos los grupos de enlaces en una página necesita estar en un elemento nav. El elemento está destinado principalmente para las secciones que constan de bloques de navegación principales.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<body>
<header>
  <h1>Wake up sheeple!</h1>
  <p><a href="news.html">News</a> -
    <a href="blog.html">Blog</a> -
    <a href="forums.html">Forums</a></p>
  <p>Last Modified: 2009-04-01</p>
  <nav>
    <h1>Navigation</h1>
    <ul>
      <li><a href="articles.html">Index of all articles</a></li>
      <li><a href="today.html">Things sheeple need to wake up for today</a></li>
      <li><a href="successes.html">Sheeple we have managed to wake</a></li>
    </ul>
  </nav>
</header>
```

Imagen 16. Ejemplo elemento nav.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-nav-element>]

1.5.4.5 Aside

Definición:

El elemento aside representa una sección de una página que consta de contenido tangencialmente relacionado con el contenido alrededor del elemento aside, y que podría ser considerado por separado de ese contenido. Estas secciones se representan a menudo como barras laterales en la tipografía impresa.

El elemento puede ser utilizado para efectos tipográficos barras laterales, para la publicidad, para grupos de elementos de navegación, así como para otros contenidos que se considera por separado del contenido principal de la página.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<body>
<header>
  <h1>My wonderful blog</h1>
  <p>My tagline</p>
</header>
<aside>
  <!-- this aside contains two sections that are tangentially related
  to the page, namely, links to other blogs, and links to blog posts
  from this blog -->
  <nav>
    <h1>My blogroll</h1>
    <ul>
      <li><a href="http://blog.example.com/">Example Blog</a>
    </ul>
  </nav>
  <nav>
    <h1>Archives</h1>
    <ol reversed>
      <li><a href="/last-post">My last post</a>
      <li><a href="/first-post">My first post</a>
    </ol>
  </nav>
</aside>
<aside>
  <!-- this aside is tangentially related to the page also, it
  contains twitter messages from the blog author -->
  <h1>Twitter Feed</h1>
  <blockquote cite="http://twitter.example.net/t31351234">
    I'm on vacation, writing my blog.
  </blockquote>
  <blockquote cite="http://twitter.example.net/t31219752">
    I'm going to go on vacation soon.
  </blockquote>
</aside>
```

Imagen 17. Ejemplo elemento aside.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-aside-element>]

1.5.4.6 H1, h2, h3, h4, h5, h6

Definición:

Estos elementos representan el encabezado correspondiente a sus secciones.

Estos elementos tienen un rango determinado por el número de su nombre. El elemento h1 se dice que tiene el rango más alto, el elemento h6 tiene el rango más bajo, y dos elementos con el mismo nombre tienen el mismo valor.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<body>
<h1>Let's call it a draw(ing surface)</h1>
<h2>Diving in</h2>
<h2>Simple shapes</h2>
<h2>Canvas coordinates</h2>
<h3>Canvas coordinates diagram</h3>
<h2>Paths</h2>
</body>
```

Imagen 18. Ejemplo elementos h1, h2, h3, h4, h5, h6.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-h1,-h2,-h3,-h4,-h5,-and-h6-elements>]

1.5.4.7 Hgroup

Definición:

El elemento hgroup representa el encabezado de una sección. El elemento se utiliza para agrupar un conjunto de elementos h1, h2, h3, h4, h5 y h6 cuando el título tiene varios niveles, tales como subtítulos, títulos alternativos.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<hgroup>
  <h1>The reality dysfunction</h1>
  <h2>Space is not the only void</h2>
</hgroup>

<hgroup>
  <h1>Dr. Strangelove</h1>
  <h2>Or: How I Learned to Stop Worrying and Love the Bomb</h2>
</hgroup>
```

Imagen 19. Ejemplo elemento hgroup.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-hgroup-element>]

1.5.4.8 Header

Definición:

El elemento header representa a un grupo de introducción o de ayudas navegación.

Un elemento header está destinado a contener por lo general la sección de encabezado (un elemento h1, h2, h3, h4, h5, h6 o un elemento hgroup), pero esto no es necesario. El elemento header también puede ser utilizado para envolver una sección de tabla de contenidos, un formulario de búsqueda, o cualquier logotipo pertinente.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<body>
<header>
  <h1>Little Green Guys With Guns</h1>
  <nav>
    <ul>
      <li><a href="/games">Games</a>
      <li><a href="/forum">Forum</a>
      <li><a href="/download">Download</a>
    </ul>
  </nav>
  <h2>Important News</h2> <!-- this starts a second subsection -->
  <!-- this is part of the subsection entitled "Important News" -->
  <p>To play today's games you will need to update your client.</p>
  <h2>Games</h2> <!-- this starts a third subsection -->
</header>
<p>You have three active games:</p>
<!-- this is still part of the subsection entitled "Games" -->
...
```

Imagen 20. Ejemplo elemento header.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-header-element>]

1.5.4.9 Footer

Definición:

El elemento footer representa un pie de página de su antecesor más próximo o seccionamiento contenido del elemento raíz. El elemento footer suele contener información sobre su sección, como quién la escribió, enlaces a documentos relacionados, datos de derechos de autor y similares.

Cuando el elemento de pie de página contiene secciones enteras, representan apéndices, índices, colofones largos, detallados acuerdos de licencia, y el contenido de este tipo.

Footer no necesariamente tiene que aparecer al final de una sección, aunque por lo general se lo hace.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<BODY>
<H1>The Ramblings of a Scientist</H1>
<ARTICLE>
  <H1>Episode 15</H1>
  <VIDEO SRC="/fm/015.ogv" CONTROLS PRELOAD>
  <P><A HREF="/fm/015.ogv">Download video</A>.</P>
</VIDEO>
  <FOOTER> <!-- footer for article -->
  <P>Published <TIME DATETIME="2009-10-21T18:26-07:00">on 2009/10/21 at 6:26pm</TIME></P>
</FOOTER>
</ARTICLE>
<ARTICLE>
  <H1>My Favorite Trains</H1>
  <P>I love my trains. My favorite train of all time is a K6f.</P>
  <P>It is fun to see them pull some coal cars because they look so
  dwarfed in comparison.</P>
  <FOOTER> <!-- footer for article -->
  <P>Published <TIME DATETIME="2009-09-15T14:54-07:00">on 2009/09/15 at 2:54pm</TIME></P>
</FOOTER>
</ARTICLE>
<FOOTER> <!-- site wide footer -->
<NAV>
  <P><A HREF="/credits.html">Credits</A> -
  <A HREF="/tos.html">Terms of Service</A> -
  <A HREF="/index.html">Blog Index</A></P>
</NAV>
<P>Copyright © 2009 Gordon Freeman</P>
</FOOTER>
</BODY>
```

Imagen 21. Ejemplo elemento footer.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-footer-element>]

1.5.4.10 Address

Definición:

El elemento address representa la información de contacto de su más cercano elemento article o body. Si ese es el elemento body, a continuación, la información de contacto se aplica al documento como un todo.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<footer>
  <address>
    For more details, contact
    <a href="mailto:js@example.com">John Smith</a>.
  </address>
  <p><small>© copyright 2038 Example Corp.</small></p>
</footer>
```

Imagen 22. Ejemplo elemento address.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-address-element>]

1.5.5 Agrupación de contenido

1.5.5.1 P

Definición:

El elemento **p** representa un párrafo.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<section>
  <!-- ... -->
  <footer>
    <p>Last modified: 2001-04-23</p>
    <address>Author: fred@example.com</address>
  </footer>
</section>
```

Imagen 23. Ejemplo elemento p.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-p-element>]

1.5.5.2 Hr

Definición:

El elemento **hr** representa un salto de párrafo de nivel temático, por ejemplo, un cambio de escena en una historia, o una transición a otro tema dentro de una sección de un libro de referencia.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<section>
  <h1>Communication</h1>
  <p>There are various methods of communication. This section
  covers a few of the important ones used by the project.</p>
  <hr>
  <p>Communication stones seem to come in pairs and have mysterious
  properties:</p>
  <ul>
    <li>They can transfer thoughts in two directions once activated
    if used alone.</li>
    <li>If used with another device, they can transfer one's
    consciousness to another body.</li>
    <li>If both stones are used with another device, the
    consciousnesses switch bodies.</li>
  </ul>
  <hr>
  <p>Radios use the electromagnetic spectrum in the meter range and
  longer.</p>
  <hr>
  <p>Signal flares use the electromagnetic spectrum in the
  nanometer range.</p>
</section>
```

Imagen 24. Ejemplo elemento hr.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-hr-element>]

1.5.5.3 Pre

Definición:

El elemento **pre** representa un bloque de texto con formato previo, en el que la estructura está representada por convenciones tipográficas en lugar de por los elementos.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<pre>                                maxling

it is with a                        heart
                                heavy

that i admit loss of a feline
                                so          loved

a friend lost to the
                                unknown

                                                                (night)

~cdr 11dec07</pre>
```

Imagen 25. Ejemplo elemento pre.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-pre-element>]

1.5.5.4 Blockquote

Definición:

El elemento blockquote representa una sección en la que se cita de otra fuente.

El contenido dentro de un blockquote debe ser citado de otra fuente, cuya dirección, si tiene uno, puede ser citado en el atributo **cite**.

Si el atributo **cite** está presente, debe ser un URL válido potencialmente rodeado por espacios. Para obtener el enlace citación correspondiente, el valor del atributo se debe resolver con respecto al elemento.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Cite	Especifica la URL (fuente de la cita)	URL

Tabla 16. Atributos elemento blockquote.

Compatible en los navegadores:



Ejemplo:

```
<blockquote>
  <p>I contend that we are both atheists. I just believe in one fewer
  god than you do. When you understand why you dismiss all the other
  possible gods, you will understand why I dismiss yours.</p>
</blockquote>
<p>— Stephen Roberts</p>
```

Imagen 26. Ejemplo elemento blockquote.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-base-element>]

1.5.5.5 Ol

Definición:

El elemento **ol** representa una lista de ítems, donde los ítems han sido intencionalmente ordenados, de tal manera que cambiando el orden cambiaría el significado del documento.

Los ítems de la lista son los nodos de elementos li secundarios del elemento ol, con el árbol ordenado.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Reversed	Especifica que el orden de la lista debe ser descendente (9,8,7 ...).	Reversed
Start	Especifica el valor inicial de una lista ordenada	Un número
Type	Especifica el tipo de marcador a utilizar en la lista	1, A, a, I, i.

Tabla 17. Atributos elemento ol.

Compatible en los navegadores:



Ejemplo:

```
<p>I have lived in the following countries (given in the order of when I first lived there):</p>
<ol>
  <li>United Kingdom
  <li>Switzerland
  <li>United States
  <li>Norway
</ol>
```

Imagen 27. Ejemplo elemento ol.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-ol-element>]

1.5.5.6 Ul

Definición:

El elemento **ul** representa una lista de elementos, donde el orden de los elementos no es importante, es decir, donde el cambio del orden no cambiaría materialmente el significado del documento.

Los elementos de la lista son los nodos de elementos **li** secundarios del elemento **ul**.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>I have lived in the following countries:</p>
<ul>
  <li>Norway
  <li>Switzerland
  <li>United Kingdom
  <li>United States
</ul>
```

Imagen 28. Ejemplo elemento ul.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-ul-element>]

1.5.5.7 Li

Definición:

El elemento **li** representa un ítem de una lista. Si su padre es un elemento **ul**, **ol**, o un elemento **menu**, entonces el elemento es un ítem de la lista del elemento padre, tal como se define por los elementos. De lo contrario, el ítem de la lista no ha definido la lista relacionada con relación a cualquier otro elemento **li**.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<ol reversed>
  <li><cite>Josie and the Pussycats</cite>, 2001</li>
  <li><cite lang="sh">Црна мачка, бели маџор</cite>, 1998</li>
  <li><cite>A Bug's Life</cite>, 1998</li>
  <li><cite>Toy Story</cite>, 1995</li>
  <li><cite>Monsters, Inc</cite>, 2001</li>
  <li><cite>Cars</cite>, 2006</li>
  <li><cite>Toy Story 2</cite>, 1999</li>
  <li><cite>Finding Nemo</cite>, 2003</li>
  <li><cite>The Incredibles</cite>, 2004</li>
  <li><cite>Ratatouille</cite>, 2007</li>
</ol>
```

Imagen 29. Ejemplo elemento li.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-li-element>]

1.5.5.8 Dl

Definición:

El elemento dl representa una lista de asociación, consta de cero o más grupos de nombre-valor (una lista descripción). Cada grupo debe constar de uno o más nombres (elementos dt) seguido por uno o más valores (elementos dd). Dentro de un solo elemento dl, no debe haber más de un elemento dt para cada nombre.

Los grupos nombre-valor pueden ser términos y definiciones, temas y valores de metadatos, preguntas y respuestas, o cualquier otro grupo de datos de nombre-valor.

Los valores dentro de un grupo son alternativas; varios párrafos que forman parte del mismo valor donde todo debe estar dentro del mismo elemento dd.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<dl>
  <dt> Authors
  <dd> John
  <dd> Luke
  <dt> Editor
  <dd> Frank
</dl>
```

Imagen 30. Ejemplo elemento dl.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-dl-element>]

1.5.5.9 Dt

Definición:

El elemento dt representa el término o nombre, que forma parte de un grupo término-descripción en una lista de descripciones (elemento dl).

El elemento dt en sí mismo, cuando se utiliza en un elemento dl, no indica que su contenido es un término que se define, pero esto se puede indicar mediante el elemento dfn.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<article>
<h1>FAQ</h1>
<dl>
  <dt>What do we want?</dt>
  <dd>Our data.</dd>
  <dt>When do we want it?</dt>
  <dd>Now.</dd>
  <dt>Where is it?</dt>
  <dd>We are not sure.</dd>
</dl>
</article>
```

Imagen 31. Ejemplo elemento dt.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-dt-element>]

1.5.5.10 Dd

Definición:

El elemento dd representa la descripción, la definición, o valor, parte de un grupo término-descripción en una lista de descripción (elemento dl).

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<dl>
  <dt><dfn>happiness</dfn></dt>
  <dd class="pronunciation">'hæ p. nes/</dd>
  <dd class="part-of-speech"><i><abbr>n.</abbr></i></dd>
  <dd>The state of being happy.</dd>
  <dd>Good fortune; success. <q>Oh <b>happiness</b>! It worked!</q></dd>
</dt><dfn>rejoice</dfn></dt>
  <dd class="pronunciation">/ri jɔis'</dd>
  <dd><i class="part-of-speech"><abbr>v.intr.</abbr></i> To be delighted oneself.</dd>
  <dd><i class="part-of-speech"><abbr>v.tr.</abbr></i> To cause one to be delighted.</dd>
</dl>
```

Imagen 32. Ejemplo elemento dd.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-dd-element>]

1.5.5.11 Figure

Definición:

El elemento **figure** representa un cierto contenido de flujo, opcionalmente con un título o leyenda, que es autónomo y típicamente se hace referencia como una sola unidad del flujo principal del documento.

El elemento puede así ser utilizado para anotar ilustraciones, diagramas, fotografías, los listados de código, etc, que se refiere a partir del contenido principal del documento, pero que podría, sin afectar el flujo del documento, ser alejado de dicho contenido principal, por ejemplo, a un lado de la página, a páginas dedicadas, o a un apéndice.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<figure>
  
  <figcaption>Bubbles at work</figcaption>
</figure>
```

Imagen 33. Ejemplo elemento figure.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-figure-element>]

1.5.5.12 Figcaption

Definición:

El elemento **figcaption** representa un subtítulo o leyenda para el resto de contenido del padre del elemento **figcaption**, el elemento **figure**.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

El ejemplo se puede observar en la imagen anterior (Imagen 33).

1.5.5.13 Div

Definición:

El elemento div no tiene un significado especial. Se puede utilizar con los atributos **class**, **lang**, y **title** para marcar semántica común a un grupo de elementos consecutivos.

A los autores se les recomienda ver el elemento **div** como un elemento de último recurso, ya cuando ningún otro elemento es adecuado. El uso de elementos más

apropiados en lugar del elemento **div** conduce a una mejor accesibilidad para los lectores y más fácil mantenimiento para los autores.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<article lang="en-US">
  <h1>My use of language and my cats</h1>
  <p>My cat's behavior hasn't changed much since her absence, except
  that she plays her new physique to the neighbors regularly, in an
  attempt to get pets.</p>
  <div lang="en-GB">
    <p>My other cat, coloured black and white, is a sweetie. He followed
    us to the pool today, walking down the pavement with us. Yesterday
    he apparently visited our neighbours. I wonder if he recognises that
    their flat is a mirror image of ours.</p>
    <p>Hm, I just noticed that in the last paragraph I used British
    English. But I'm supposed to write in American English. So I
    shouldn't say "pavement" or "flat" or "colour"...</p>
  </div>
  <p>I should say "sidewalk" and "apartment" and "color"!</p>
</article>
```

Imagen 34. Ejemplo elemento div.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-div-element>]

1.5.6 Semántica Nivel-Texto

1.5.6.1 A

Definición:

Si el elemento tiene un atributo href, entonces representa un hipervínculo (un ancla de hipertexto) marcada por su contenido.

Los atributos **target**, **rel**, **media**, **hreflang**, y **type** deben ser omitidos si el atributo **href** no está presente.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Href	Especifica la dirección URL de la página a donde el enlace va.	URL
Target	Especifica dónde abrir el documento vinculado	_blank, _parent, _self, _top, Nombre de frame.
Rel	Especifica la relación entre el documento actual y el documento vinculado	alternate, autor, bookmark, help, license, next, nofollow, noreferrer, prefetch, prev, search, tag.
Media	Especifica con qué dispositivo el documento vinculado se mostrará.	(<i>media_query</i>)
Hreflang	Especifica el idioma del texto del documento vinculado	Código de lenguaje (<i>lenguaje_code</i>).
Type	Especifica el tipo MIME del documento vinculado	Tipo MIME (<i>MIME_Type</i>).

Tabla 18. Atributos elemento a.

Compatible en los navegadores:



Ejemplo:

```
<nav>
<ul>
<li> <a href="/">Home</a> </li>
<li> <a href="/news">News</a> </li>
<li> <a>Examples</a> </li>
<li> <a href="/legal">Legal</a> </li>
</ul>
</nav>
```

Imagen 35. Ejemplo elemento a.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-a-element>]

1.5.6.2 Em

Definición:

El elemento **em** representa énfasis de su contenido.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p><em>Cats</em> are cute animals.</p>
```

Imagen 36. Ejemplo elemento em.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-em-element>]

1.5.6.3 Strong

Definición:

El elemento **strong** representa una gran importancia para su contenido.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p><strong>Warning.</strong> This dungeon is dangerous.  
<strong>Avoid the ducks.</strong> Take any gold you find.  
<strong><strong>Do not take any of the diamonds</strong>,&br/>they are explosive and <strong>will destroy anything within  
ten meters.</strong></strong> You have been warned.</p>
```

Imagen 37. Ejemplo elemento strong.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-strong-element>]

1.5.6.4 Small

Definición:

El elemento **small** representa comentarios adversos como letra pequeña.

Small típicamente incluye avisos legales, advertencias, restricciones legales o derechos de autor. El elemento **small** a veces también se usa para la atribución o para satisfacer los requisitos de licencia.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>Example Corp today announced record profits for the  
second quarter <small>(Full Disclosure: Foo News is a subsidiary of  
Example Corp)</small>, leading to speculation about a third quarter  
merger with Demo Group.</p>
```

Imagen 38. Ejemplo elemento small.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-small-element>]

1.5.6.5 S

Definición:

El elemento s representa el texto que ya no es correcto, exacto o pertinente.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>Buy our Iced Tea and Lemonade!</p>  
<p><s>Recommended retail price: $3.99 per bottle</s></p>  
<p><strong>Now selling for just $2.99 a bottle!</strong></p>
```

Imagen 39. Ejemplo elemento s.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-s-element>]

1.5.6.6 Cite

Definición:

El elemento **cite** representa el título de una obra (por ejemplo, un libro, un artículo, un ensayo, un poema, una partitura, una canción, un guión, una película, un programa de televisión, un juego, una escultura, una pintura, una obra de teatro, una ópera, un musical, una exposición, un caso legal, etc.). Esto puede ser un trabajo que se cita o se hace referencia en detalle (es decir, una cita), o simplemente puede ser una obra que se menciona de pasada.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>According to the Wikipedia article <cite>HTML</cite>, as it stood in mid-February 2008, leaving attribute values unquoted is unsafe. This is obviously an over-simplification.</p>
```

Imagen 40. Ejemplo elemento cite.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-cite-element>]

1.5.6.7 Q

Definición:

El elemento **q** representa algunos contenidos de fraseo citados de otra fuente.

El uso de elementos **q** para marcar citas es completamente opcional; utilizando puntuaciones como las comillas sin elementos **q**, es tan correcto de igual manera.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Cite	Especifica la dirección URL de origen de la cita	URL

Tabla 19. Atributos elemento q.

Compatible en los navegadores:



Ejemplo:

```
<p>In <cite>Example One</cite>, he writes <q>The man said <q>Things that are impossible just take longer</q>. I disagreed with him</q>. Well, I disagree even more!</p>
```

Imagen 41. Ejemplo elemento q.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-q-element>]

1.5.6.8 Dfn

Definición:

El elemento **dfn** representa la instancia de definición de un término.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>The <dfn><abbr title="Garage Door Opener">GDO</abbr></dfn>
is a device that allows off-world teams to open the iris.</p>
<!-- ... later in the document: -->
<p>Teal'c activated his <abbr title="Garage Door Opener">GDO</abbr>
and so Hammond ordered the iris to be opened.</p>
```

Imagen 42. Ejemplo elemento dfn.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-dfn-element>]

1.5.6.9 Abbr

Definición:

El elemento **abbr** representa una abreviatura o sigla, opcionalmente con su expansión. El atributo **title** puede ser utilizado para proporcionar una expansión de la abreviatura. El atributo, si se especifica, debe contener la expansión de la abreviatura, y nada más.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>The <a href="#whatwg"><abbr
title="Web Hypertext Application Technology Working Group">WHATWG</abbr></a>
community does not have much representation from Asia.</p>
```

Imagen 43. Ejemplo elemento abbr.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-abbr-element>]

1.5.6.10 Time

Definición:

El elemento **time** representa su contenido, junto con una forma legible por la máquina de los contenidos en el atributo **datetime**. El tipo de contenido se limita a los diversos tipos de fechas, horas, compensaciones horarias y duraciones.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
datetime	Da la fecha/hora que se especifica. De lo contrario, la fecha/hora está dada por los contenidos del elemento	Formato de fecha/hora (<i>datetime</i>)

Tabla 20. Atributos elemento time.

Compatible en los navegadores:



Ejemplo:

```
Today is <time datetime="2011-11-12">Tuesday</time>.
```

Imagen 44. Ejemplo elemento time.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-time-element>]

1.5.6.11 Code

Definición:

El elemento **code** representa un fragmento de código informático. Esto podría ser un nombre de elemento XML, un nombre de archivo, un programa informático, o cualquier otra cadena que una computadora podría reconocer.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<pre><code class="language-pascal">var i: Integer;  
begin  
  i := 1;  
end.</code></pre>
```

Imagen 45. Ejemplo elemento code.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-code-element>]

1.5.6.12 Var

Definición:

El elemento **var** representa una variable. Esta podría ser una variable real en una expresión matemática o contexto de la programación, un identificador que representa una constante, un símbolo que indique una cantidad física, un parámetro de función, o simplemente ser un término que se utiliza como un marcador de posición.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>If there are <var>n</var> pipes leading to the ice  
cream factory then I expect at <em>least</em> <var>n</var>  
flavors of ice cream to be available for purchase!</p>
```

Imagen 46. Ejemplo elemento var.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-var-element>]

1.5.6.13 Samp

Definición:

El elemento **samp** representa (muestra) la salida (output) de un programa o sistema informático.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>The computer said <samp>Too much cheese in tray  
two</samp> but I didn't know what that meant.</p>
```

Imagen 47. Ejemplo elemento samp.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-samp-element>]

1.5.6.14 Kbd

Definición:

El elemento **kbd** representa la entrada del usuario (típicamente la entrada de teclado, aunque también puede ser utilizado para representar otra entrada, tales como comandos de voz).

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>To make George eat an apple, select <kbd>File | Eat Apple...</kbd></p>
```

Imagen 48. Ejemplo elemento kbd.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-kbd-element>]

1.5.6.15 Sub

Definición:

El elemento **sub** define texto subíndice. El texto de subíndice aparece por debajo de la línea de base. El texto de subíndice se puede utilizar para las fórmulas químicas, como H₂O.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>The coordinate of the <var>i</var>th point is  
(<var>x<sub><var>i</var></sub></var>, <var>y<sub><var>i</var></sub></var>)  
For example, the 10th point has coordinate  
(<var>x<sub>10</sub></var>, <var>y<sub>10</sub></var>).</p>
```

Imagen 49. Ejemplo elemento sub.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-sub-and-sup-elements>]

1.5.6.16 Sup

Definición:

El elemento **sup** define el texto de superíndice. El texto superíndice aparece por encima de la línea de base. Superíndice texto puede ser utilizado para notas al pie, como WWW^[1].

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<var>E</var>=<var>m</var><var>c</var><sup>2</sup>  
f(<var>x</var>, <var>n</var>) = log<sub>4</sub><var>x</var><sup><var>n</var></sup>
```

Imagen 50. Ejemplo elemento sup.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-sub-and-sup-elements>]

1.5.6.17 I

Definición:

El elemento **i** define una parte de texto en una voz alternativa o estado de ánimo. El contenido del elemento normalmente se muestra en cursiva.

Este elemento se puede utilizar para indicar un término técnico, una frase de otra lengua, un pensamiento, o un nombre de barco, etc.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>The <i class="taxonomy">Felis silvestris catus</i> is cute.</p>  
<p>The term <i>prose content</i> is defined above.</p>  
<p>There is a certain <i lang="fr">je ne sais quoi</i> in the air.</p>
```

Imagen 51. Ejemplo elemento i.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-i-element>]

1.5.6.18 B

Definición:

El elemento **b** representa un fragmento de texto al que se llama la atención con fines utilitarios sin transmitir ninguna importancia extra y sin que ello implique una voz alternativa o estado de ánimo, tales como palabras clave en un resumen del documento, los nombres de productos en sus revisiones, palabras recurribles en texto interactivo impulsado por software.

Se debe utilizar como último recurso, cuando no hay otra etiqueta que sea la más apropiada.

El contenido del elemento normalmente se muestra en negrita.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>The <b>frobonitor</b> and <b>barbinator</b> components are fried.</p>
```

Imagen 52. Ejemplo elemento b.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-b-element>]

1.5.6.19 U

Definición:

El elemento **u** representa un texto que debe ser estilísticamente diferente de texto normal, tales como palabras mal escritas o nombres propios en chino.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>This is a <u>parragraph</u>.</p>
```

Imagen 53. Ejemplo elemento u.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_u.asp]

1.5.6.20 Mark

Definición:

El elemento **mark** representa un texto marcado o resaltado.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>I also have some <mark>kitten</mark>s who are visiting me  
these days. They're really cute. I think they like my garden! Maybe I  
should adopt a <mark>kitten</mark>.</p>
```

Imagen 54. Ejemplo elemento mark.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-mark-element>]

1.5.6.21 Ruby

Definición:

El elemento **ruby** representa una anotación ruby. Las anotaciones ruby se utilizan para la tipografía en el Este de Asia, para mostrar la pronunciación de los caracteres de Asia oriental.

Utilizar el elemento **ruby** junto con el **tr** y/o elementos **rp**: El elemento **ruby** consta de uno o más caracteres que necesitan una explicación/pronunciación, y un elemento **rt** que da esa información, y un elemento opcional **rp** que define lo que deben mostrar los navegadores que no soportan anotaciones ruby.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
...<ruby>汉<rt>hàn</rt>字<rt>zì</rt></ruby>...
```

Imagen 55. Ejemplo elemento ruby.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-ruby-element>]

1.5.6.22 Rt

Definición:

El elemento **rt** define una explicación o la pronunciación de caracteres (para la tipografía del Este de Asia) en una anotación de ruby.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

El ejemplo se puede observar en la imagen anterior (imagen 55).

1.5.6.23 Rp

Definición:

El elemento **rp** define lo que se mostrará si un navegador no es compatible con las anotaciones de ruby.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
...  
<ruby>漢<rp> (</rp><rt>かん</rt><rp>) </rp>字<rp> (</rp><rt>じ</rt><rp>) </rp></ruby>  
...
```

Imagen 56. Ejemplo elemento rp.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-rp-element>]

1.5.6.24 Bdi

Definición:

El elemento **bdi** representa un fragmento de texto que va a ser aislado de su entorno para los fines de formato de texto bidireccional.

BDI representa Bi-direccional aislamiento.

El atributo global **dir** está predeterminado en “auto” en este elemento.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<ul>  
<li>User <bdi>jcranmer</bdi>: 12 posts.  
<li>User <bdi>hober</bdi>: 5 posts.  
<li>User <bdi>إيان</bdi>: 3 posts.  
</ul>
```

Imagen 57. Ejemplo elemento bdi.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-bdi-element>]

1.5.6.25 Bdo

Definición:

El elemento **bdo** representa direccionalidad explícita de control de formato de texto para sus hijos. Esto permite a los autores anular el algoritmo bidireccional de Unicode (BDI) especificando explícitamente una dirección.

Los autores deben especificar el atributo **dir** en este elemento, con el valor **ltr** para especificar una redefinición de izquierda a derecha y con el valor **rtl** para especificar una redefinición de derecha a izquierda.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<bdo dir="rtl">  
This text will go right-to-left.  
</bdo>
```

Imagen 58. Ejemplo elemento bdo.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_bdo.asp]

1.5.6.26 Span

Definición:

El elemento **span** no significa nada por sí mismo, pero puede ser útil cuando se utiliza junto con los atributos globales, por ejemplo, **class**, **lang**, o **dir**. Esto representa a sus hijos.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>My mother has <span style="color:blue">blue</span> eyes.</p>
```

Imagen 59. Ejemplo elemento span.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_span.asp]

1.5.6.27 **Br**

Definición:

El elemento **br** representa un salto de línea.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>P. Sherman<br>42 Wallaby Way<br>Sydney</p>
```

Imagen 60. Ejemplo elemento br.

Fuente: W3 Schools [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-br-element>]

1.5.6.28 **Wbr**

Definición:

La oportunidad de salto de palabras (**wbr**) especifica donde, en un texto debería agregarse un salto de línea.

Consejo: Cuando una palabra es demasiado larga, o se tiene miedo de que el navegador rompa las líneas en el lugar equivocado, se puede utilizar el elemento **wbr** para agregar palabras de oportunidades de quiebre (**WBR**).

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<p>  
To learn AJAX, you must be familiar with the XML<wbr>Http<wbr>Request  
Object.  
</p>
```

Imagen 61. Ejemplo elemento wbr.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_wbr.asp]

1.5.7 Ediciones

1.5.7.1 Ins

Definición:

El elemento **ins** representa una adición al documento una inserción de un texto en un documento.

Los navegadores por lo general trazan una línea como de subrayado cuando se inserta un texto con el elemento **ins**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Cite	Especifica una URL a un documento que explica la razón por la que el texto ha sido insertado/cambiado	URL
datetime	Especifica la fecha y hora en que se insertó/cambió el texto.	YYYY-MM-DDThh:mm:ssTZD

Tabla 21. Atributos elemento ins.

Compatible en los navegadores:



Ejemplo:

```
<aside>
  <ins datetime="2005-03-16 00:00Z">
    <p> I like fruit. </p>
  </ins>
  <ins datetime="2005-03-16 00:00Z">
    Apples are <em>tasty</em>.
  </ins>
  <ins datetime="2007-12-19 00:00Z">
    So are pears.
  </ins>
</aside>
```

Imagen 62. Ejemplo elemento ins.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-ins-element>]

1.5.7.2 Del

Definición:

El elemento **del** representa un texto que ha sido borrado de un documento.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Cite	Especifica una URL a un documento que explica la razón por la que el texto ha sido insertado/cambiado	URL
datetime	Especifica la fecha y hora en que se insertó/cambió el texto.	YYYY-MM-DDThh:mm:ssTZD

Tabla 22. Atributos elemento del.

Compatible en los navegadores:



Ejemplo:

```
<h1>To Do</h1>
<ul>
  <li>Empty the dishwasher</li>
  <li><del datetime="2009-10-11T01:25-07:00">Watch Walter Lewin's lectures</del></li>
  <li><del datetime="2009-10-10T23:38-07:00">Download more tracks</del></li>
  <li>Buy a printer</li>
</ul>
```

Imagen 63. Ejemplo elemento del.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-del-element>]

1.5.8 Contenido incrustado (embebido)

1.5.8.1 Img

Definición:

Un elemento **img** representa una imagen.

La imagen dada por los atributos **src** es el contenido incrustado, el valor del atributo **alt** proporciona contenido equivalente para aquellos que no pueden procesar las imágenes o que han deshabilitado la carga de imágenes.

El atributo **src** debe estar presente, y debe contener una dirección URL válida que no esté vacía, potencialmente rodeado de espacios hacen referencia recursos no-interactivos.

Los requisitos anteriores implican que las imágenes pueden ser mapas de bits estáticos (por ejemplo, los archivos PNG GIF, JPEG), documentos vectoriales de una sola página (archivos PDF de una sola página, los archivos XML con un elemento raíz SVG), mapas de bits animados (APNGs, GIFs animados), vector de dibujos animados gráficos (XML archivos con un elemento raíz SVG que utilizan animación SMIL declarativa), y así sucesivamente. Sin embargo, estas definiciones excluyen archivos SVG con scripts, archivos de multipágina PDF, archivos interactivos MNG, documentos HTML, documentos de texto sin formato, etc.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Alt	Especifica un texto alternativo para una imagen	texto
Src	Especifica la dirección URL de una imagen	URL
Crossorigin	Su propósito es permitir que las imágenes de los sitios de terceros permitan acceso a orígenes múltiples (cross-origin), para usar con canvas .	Anonymous, use-credentials.
Usemap	Especifica una imagen como un mapa de imagen del lado del cliente.	Nombre del mapa (#mapname)
Ismap	Especifica una imagen como un mapa de imagen del lado del servidor.	ismap
Width	Especifica el ancho de una imagen	Tamaño en Pixeles
Height	Especifica el alto de una imagen	Tamaño en Pixeles

Tabla 23. Atributos elemento img.

Compatible en los navegadores:



Ejemplo:

```
<p>Carouge has a coat of arms.</p>
<p></p>
<p>The coat of arms depicts a lion, sitting in front of a tree.
It is used as decoration all over the town.</p>
```

Imagen 64. Ejemplo elemento img.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-img-element>]

1.5.8.2 Iframe

Definición:

El elemento iframe representa un contexto de navegación anidada.

Especifica un marco flotante, un marco flotante se utiliza para insertar otro documento dentro del documento HTML actual.

Atributos:

✓ Atributos globales

Nombre	Descripción	Valores
Src	Especifica la dirección del documento para insertar en el <iframe>	URL
Srcdoc	Especifica el contenido HTML de la página para mostrar en el <iframe>	Código HTML
Name	Especifica el nombre de un <iframe>	Un nombre
Sandbox	Permite un conjunto de restricciones adicionales para el contenido en el <iframe>	"" , allow-forms, allow-same-origin, allow-scripts,

		allow-top-navigation.
Seamless	Especifica que el <iframe> debe parecer que es una parte del documento que contiene.	Seamless
Width	Especifica el ancho de un <iframe>	Tamaño en pixeles
Height	Especifica el alto de un <iframe>	Tamaño en pixeles

Tabla 24. Atributos elemento iframe.

Compatible en los navegadores:



Ejemplo:

```
<article>
  <footer> Five minutes ago, <a href="/users/ch">ch</a> wrote: </footer>
  <iframe seamless sandbox srcdoc="<p>hey that's earl's table.
  <p>you should get earl&amp;me on the next cover."></iframe>
</article>
```

Imagen 65. Ejemplo elemento iframe.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-iframe-element>]

1.5.8.3 Embed

Definición:

El elemento **embed** representa un punto de integración para una aplicación externa (normalmente no HTML) o contenido interactivo.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Height	Especifica la altura del contenido integrado (<embed>)	Tamaño en pixeles
Src	Especifica la dirección del archivo externo para empotrar	URL
Type	Especifica el tipo MIME del contenido integrado	Tipo de MIME (<i>MIME_type</i>)
Width	Especifica el ancho del contenido integrado (<embed>)	Tamaño en pixeles

Tabla 25. Atributos elemento embed.

Compatible en los navegadores:



Ejemplo:

```
<embed src="catgame.swf">
```

Imagen 66. Ejemplo elemento embed.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-embed-element>]

1.5.8.4 Object

Definición:

El elemento **object** puede representar un recurso externo, que, dependiendo del tipo de recurso, tanto será tratado como una imagen, como un contexto de navegación anidada, o un recurso externo a ser procesados por un plugin.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Data	Especifica la dirección URL del recurso a ser utilizado por el objeto	URL
Type	Especifica el tipo MIME de los datos especificados en el atributo data	Tipo de MIME (<i>MIME_type</i>)
typemustmatch	Indica que el recurso especificado por el atributo data es sólo para ser usado si el valor del atributo type y el tipo de contenido del recurso mencionado antes coinciden o son iguales. El atributo typemustmatch no debe especificarse a menos que el atributo data y el atributo type estén presentes.	typemustmatch
Name	Especifica el nombre del objeto	Un nombre
Usemap	Especifica el nombre de un mapa de imagen del lado del cliente para ser utilizado con el objeto	Nombre del mapa (<i>#mapname</i>)
Form	Especifica uno o más <i>forms</i> al que pertenece el objeto.	ID del form (<i>form_id</i>)
Width	Especifica el ancho del objeto	Tamaño en pixeles
Height	Especifica el alto del objeto	Tamaño en pixeles

Tabla 26. Atributos elemento object.

Compatible en los navegadores:



Ejemplo:

```
<figure>
  <object type="application/x-java-applet">
    <param name="code" value="MyJavaClass">
      <p>You do not have Java available, or it is disabled.</p>
    </object>
  <figcaption>My Java Clock</figcaption>
</figure>
```

Imagen 67. Ejemplo elemento object.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-object-element>]

1.5.8.5 Param

Definición:

El elemento **param** define los parámetros para plugins invocados por elementos objeto. No representa nada en sí mismo.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Name	Especifica el nombre del parámetro (param)	Un nombre
Value	Especifica el valor del parámetro	Un valor

Tabla 27. Atributos elemento param.

Compatible en los navegadores:



Ejemplo:

```
<object type="application/vnd.o3d.auto">
  <param name="o3d_features" value="FloatingPointTextures">
  
  <p>To see the teapot actually rendered by O3D on your
  computer, please download and install the <a
  href="http://code.google.com/apis/o3d/docs/gettingstarted.html#install">O3D plugin</a>.</p>
</object>
```

Imagen 68. Ejemplo elemento param.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-param-element>]

1.5.8.6 Video

Definición:

Un elemento **video** se utiliza para la reproducción de vídeos o películas y archivos de audio con subtítulos.

El contenido puede ser proporcionado dentro del elemento de vídeo. El navegador no muestra ese contenido para el usuario, sino que está pensado para los navegadores web más antiguos que no soportan vídeo, por lo que los plugins de vídeo antiguos puede intentar reproducirse, o para mostrar texto a los usuarios de estos navegadores antiguos informándoles de cómo acceder a el contenido de vídeo.

Subtítulos pueden ser proporcionados, ya sean incrustados en la secuencia de vídeo o como archivos externos usando el elemento de pista (**track**).

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Src	Especifica la dirección URL del archivo de vídeo	URL
Crossorigin	Es un atributo de ajustes de CORS. CORS es sinónimo de intercambio de recursos de orígenes múltiples. El propósito del atributo crossorigin es que le permite configurar las peticiones CORS de datos obtenidos del elemento.	Anonymous, Use-credentials.
Poster	Especifica una imagen que se muestra mientras el vídeo se carga, o hasta que el usuario pulse el botón de reproducción.	URL
Preload	Especifica si y cómo el autor piensa que el video debe ser cargado cuando se carga la página.	Auto, Metadata, None.
Autoplay	Especifica que el vídeo comenzará a reproducirse tan pronto como esté listo.	autoplay
Mediagroup	Para sincronizar la reproducción de vídeos (o elementos multimedia). Permite especificar los elementos media para enlazar. El valor es una cadena de texto, por ejemplo: mediagroup=movie. Elementos videos/media con el mismo valor que enlaza automáticamente el navegador. Un ejemplo de donde el atributo MediaGroup se podría utilizar es donde usted necesita para superponer una pista intérprete de lenguaje de un archivo de película en la parte superior de otro.	Cadena de texto
Loop	Especifica que el vídeo comenzará	Loop

	otra vez, cada vez que se terminó	
Muted	Especifica que la salida de audio del video debería ser silenciado	Muted
Controls	Especifica los controles de vídeo que se debe mostrar (como botón de reproducción/pausa, etc).	Controls
Width	Establece el ancho del reproductor de vídeo	Tamaño en pixeles
Height	Establece el alto del reproductor de vídeo	Tamaño en pixeles

Tabla 28. Atributos elemento video.

Compatible en los navegadores:



Ejemplo:

```
<p><video src="tgif.vid" autoplay controls onerror="failed(event)"></video></p>
<p><a href="tgif.vid">Download the video file</a>.</p>
```

Imagen 69. Ejemplo elemento video.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-video-element>]

1.5.8.7 Audio

Definición:

Un elemento **audio** representa un flujo de sonido o de audio.

El contenido puede ser proporcionado en el interior del elemento **audio**. Los navegadores no muestran ese contenido para el usuario, sino que está pensado para los navegadores web antiguos que no son compatibles con audio, por lo que los plugins de audio que se ponga dentro del elemento **audio** pueden intentar reproducirse, o para mostrar texto a los usuarios de estos navegadores antiguos informándoles de cómo acceder a el contenido de audio.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Src	Especifica la dirección URL del archivo de sonido	URL
Crossorigin	Es un atributo de ajustes de CORS. CORS es sinónimo de intercambio de recursos de orígenes múltiples. El propósito del atributo crossorigin es que le permite configurar las peticiones CORS de datos obtenidos del elemento.	Anonymous, Use-credentials.
Preload	Especifica si y cómo el autor piensa que el audio debe ser cargado cuando se carga la página.	Auto, Metadata, None.
Autoplay	Especifica que el audio comenzará a reproducirse tan pronto como esté listo.	autoplay
Mediagroup	Para sincronizar la reproducción de audios (o elementos multimedia). Permite especificar los elementos media para enlazar. El valor es una cadena de texto, por ejemplo: mediagroup=movie. Elementos videos/media con el mismo valor que enlaza automáticamente el navegador. Un ejemplo de donde el atributo MediaGroup se podría utilizar es donde usted necesita para superponer una pista intérprete de lenguaje de un archivo de película en la parte superior de otro.	Cadena de texto
Loop	Especifica que el vídeo comenzará otra vez, cada vez que se terminó	Loop

Muted	Especifica que la salida de audio del video debería ser silenciado	Muted
Controls	Especifica los controles de vídeo que se debe mostrar (como botón de reproducción/pausa, etc).	Controls

Tabla 29. Atributos elemento audio.

Compatible en los navegadores:



Ejemplo:

```
<audio controls="controls">
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mp3">
  Your browser does not support the audio tag.
</audio>
```

Imagen 70. Ejemplo elemento audio.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_audio.asp]

1.5.8.8 Source

Definición:

El elemento **source** permite a los autores especificar múltiples alternativas de recursos para elementos media. No representa nada en sí mismo.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Src	Especifica la dirección URL del archivo multimedia.	URL
Type	Especifica el tipo MIME del recurso multimedia	Tipo de MIME (<i>MIME_type</i>)
media	Especifica el tipo de recurso multimedia	(<i>media_query</i>)

Tabla 30. Atributos elemento source.

Compatible en los navegadores:



Ejemplo:

```
<video controls autoplay>
  <source src='video.mp4' type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
  <source src='video.ogv' type='video/ogg; codecs="theora, vorbis"'
        onerror="fallback(parentNode)">
  ...
</video>
```

Imagen 71. Ejemplo elemento source.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-source-element>]

1.5.8.9 Track

Definición:

El elemento **track** permite a los autores especificar explícitamente externas programadas pistas de texto para los elementos media. No representa nada en sí mismo.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Kind	Especifica el tipo de pista de texto	captions, chapters, descriptions, metadata, subtitles.
Src	Obligatorio. Especifica la dirección URL del archivo de pista	URL
Srclang	Especifica el lenguaje de los datos de la pista de texto (necesario si kind = "subtitles"). El valor debe ser una dirección válida etiqueta de lenguaje BCP47.	Código de lenguaje (<i>language_code</i>)

Label	Especifica el título de la pista de texto	texto
Default	Especifica que la pista está habilitado si las preferencias de los usuarios no indican que otra pista sería más apropiado.	default

Tabla 31. Atributos elemento track.

Compatible en los navegadores:



Ejemplo:

```
<video src="brave.webm">
  <track kind=subtitles src=brave.en.vtt srclang=en label="English">
  <track kind=captions src=brave.en.hoh.vtt srclang=en label="English for the Hard of Hearing">
  <track kind=subtitles src=brave.fr.vtt srclang=fr lang=fr label="Français">
  <track kind=subtitles src=brave.de.vtt srclang=de lang=de label="Deutsch">
</video>
```

Imagen 72. Ejemplo elemento track.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-track-element>]

1.5.8.10 Canvas

Definición:

El elemento **canvas** se utiliza para dibujar gráficos, gráficos de juego, arte, u otras imágenes visuales, sobre la marcha, a través de secuencias de comandos (normalmente JavaScript).

El elemento **canvas** es sólo un contenedor de gráficos, se debe utilizar un script para extraer realmente los gráficos.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Width	Especifica el ancho del canvas.	Tamaño en pixeles
Height	Especifica el alto del canvas.	Tamaño en pixeles

Tabla 32. Atributos elemento canvas.

Compatible en los navegadores:



Ejemplo:

```
<canvas id="myCanvas"></canvas>

<script type="text/javascript">
var canvas=document.getElementById('myCanvas');
var ctx=canvas.getContext('2d');
ctx.fillStyle='#FF0000';
ctx.fillRect(0,0,80,100);
</script>
```

Imagen 73. Ejemplo elemento canvas.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_canvas.asp]

1.5.8.11 Map

Definición:

El elemento **map**, junto con algún elemento **area**, define un mapa de imagen. El elemento representa a sus hijos.

El atributo **name** del elemento es requerido y se asocia con el atributo **usemap** del elemento **img** y crea una relación entre la imagen y el mapa.

El elemento **map** contiene una serie de elementos **area**, que define las áreas seleccionables en el mapa de imágenes.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Name	Obligatorio. Especifica el nombre de una imagen de mapa.	Nombre del mapa (<i>mapname</i>)

Tabla 33. Atributos elemento map.

Compatible en los navegadores:



Ejemplo:

```

<!DOCTYPE HTML>
<TITLE>Babies™: Toys</TITLE>
<HEADER>
  <H1>Toys</H1>
  <IMG SRC="/images/menu.gif"
      ALT="Babies™ navigation menu. Select a department to go to its page."
      USEMAP="#NAV">
</HEADER>
...
<FOOTER>
  <MAP NAME="NAV">
    <P>
      <A HREF="/clothes/">Clothes</A>
      <AREA ALT="Clothes" COORDS="0,0,100,50" HREF="/clothes/"> |
      <A HREF="/toys/">Toys</A>
      <AREA ALT="Toys" COORDS="0,0,100,50" HREF="/toys/"> |
      <A HREF="/food/">Food</A>
      <AREA ALT="Food" COORDS="0,0,100,50" HREF="/food/"> |
      <A HREF="/books/">Books</A>
      <AREA ALT="Books" COORDS="0,0,100,50" HREF="/books/">
    </MAP>
  </FOOTER>

```

Imagen 74. Ejemplo elemento map.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-map-element>]

1.5.8.12 Area

Definición:

El elemento **area** define un área dentro de una imagen de mapa (una imagen del mapa es una imagen con áreas seleccionables).

El elemento **area** siempre está anidado dentro de un elemento **map**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Alt	Especifica un texto alternativo para la zona. Requerido si el atributo href está presente	Texto
Coords	Especifica las coordenadas del área	Según el shape: rect: left, top, right, bottom. circle: center-x, center-y, radius. poly: x1, y1, x2, y2, ...
shape	Especifica la forma (shape) del área.	Default, Rect, Circle, Poly.
Href	Especifica el destino del hipervínculo para el área	URL
target	Especifica dónde abrir la dirección URL de destino	_blank, _parent, _self, _top, Nombre de un frame.
Rel	Especifica la relación entre el documento actual y la URL de destino	alternate, autor, bookmark, help, license, next, nofollow,

		noreferrer, prefetch, prev, search, tag.
media	Especifica qué medio/dispositivo del recurso multimedia está optimizado para aquello.	(media_query).
hreflang	Especifica el idioma o lenguaje de la URL de destino	Código de lenguaje o idioma (language_code)
Type	Especifica el tipo MIME de la dirección URL de destino	Tipo de MIME (MIME_type)

Tabla 34. Atributos elemento area.

Compatible en los navegadores:



Ejemplo:

```
<p>
Please select a shape:

<area shape=rect coords="50,50,100,100"> <!-- the hole in the red box -->
<area shape=rect coords="25,25,125,125" href="red.html" alt="Red box.">
<area shape=circle coords="200,75,50" href="green.html" alt="Green circle.">
<area shape=poly coords="325,25,262,125,388,125" href="blue.html" alt="Blue triangle.">
<area shape=poly coords="450,25,435,60,400,75,435,90,450,125,465,90,500,75,465,60"
href="yellow.html" alt="Yellow star.">
</map>
</p>
```

Imagen 75. Ejemplo elemento area.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-area-element>]

1.5.9 Datos tabulares

1.5.9.1 Table

Definición:

El elemento **table** representa los datos con más de una dimensión, en la forma de una tabla.

El elemento **table** participa en el modelo de tabla. Las tablas tienen filas, columnas y celdas dadas por sus descendientes. Las filas y columnas forman una rejilla; celdas de una tabla que debe cubrir completamente la red sin solapamiento.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Border	Especifica si las celdas de la tabla debe tener un borde o no.	1

Tabla 35. Atributos elemento table.

Compatible en los navegadores:



Ejemplo:

```
<table>
<colgroup><col><col><col>
<colgroup><col><col><col>
<colgroup><col><col><col>
<tbody>
<tr> <td> 1 <td>   <td> 3 <td> 6 <td>   <td> 4 <td> 7 <td>   <td> 9
<tr> <td>   <td> 2 <td>   <td>   <td> 9 <td>   <td>   <td> 1 <td>
<tr> <td> 7 <td>   <td>   <td>   <td>   <td>   <td>   <td>   <td> 6
<tbody>
<tr> <td> 2 <td>   <td> 4 <td>   <td> 3 <td>   <td> 9 <td>   <td> 8
<tr> <td>   <td>   <td>   <td>   <td>   <td>   <td>   <td>   <td>
<tr> <td> 5 <td>   <td>   <td> 9 <td>   <td> 7 <td>   <td>   <td> 1
<tbody>
<tr> <td> 6 <td>   <td>   <td>   <td> 5 <td>   <td>   <td>   <td> 2
<tr> <td>   <td>   <td>   <td>   <td> 7 <td>   <td>   <td>   <td>
<tr> <td> 9 <td>   <td>   <td> 8 <td>   <td> 2 <td>   <td>   <td> 5
</table>
```

Imagen 76. Ejemplo elemento table.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-table-element>]

1.5.9.2 Caption

Definición:

El elemento **caption** representa el título de la tabla que es su padre, si tiene un padre y ese es un elemento **table**.

El elemento **caption** participa en el modelo de tabla.

Debe insertarse inmediatamente después de del elemento **table**.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<table border="1">
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```

Imagen 77. Ejemplo elemento caption.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_caption.asp]

1.5.9.3 Colgroup

Definición:

El elemento **colgroup** representa un grupo de una o más columnas de la tabla que es su padre, si tiene un padre y ese es un elemento **table**.

Si el elemento **colgroup** no contiene elementos **col**, entonces el elemento debería tener el atributo **span**, cuyo valor válido debe ser un número entero no negativo mayor que cero.

El elemento **colgroup** y su atributo **span** participan en el modelo de tabla.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Span	Especifica el número de columnas que debería abarcar un grupo de columnas.	Un número.

Tabla 36. Atributos elemento colgroup.

Compatible en los navegadores:



Ejemplo:

```
<table border="1">
  <colgroup>
    <col span="2" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
</table>
```

Imagen 78. Ejemplo elemento colgroup.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_colgroup.asp]

1.5.9.4 Col

Definición:

Si un elemento **col** tuviera un padre y ese fuera un elemento **colgroup** y este último fuera hijo de un elemento padre **table**, entonces el elemento **col** representa una o más columnas del grupo de columnas representadas por **colgroup**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Span	Especifica el número de columnas que debería abarcar un grupo de columnas.	Un número.

Tabla 37. Atributos elemento col.

Compatible en los navegadores:



Ejemplo:

El ejemplo se puede observar en la imagen anterior (imagen 78).

1.5.9.5 Tbody

Definición:

El elemento **tbody** representa un bloque de filas que se componen de un cuerpo de datos para un elemento **table** padre, si el elemento **tbody** tuviera un padre y sea un **table**.

El elemento **tbody** participa en el modelo de tabla.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<table border="1">
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```

Imagen 79. Ejemplo elemento tbody.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_tbody.asp]

1.5.9.6 Thead

Definición:

El elemento **thead** representa el bloque de filas que se componen de las etiquetas (labels) de columna (cabeceras) para el elemento **table** padre, si el elemento **thead** tuviera un padre y este sea un elemento **table**.

El elemento **thead** participa en el modelo de tabla.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<table border="1">
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>$180</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>January</td>
      <td>$100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>$80</td>
    </tr>
  </tbody>
</table>
```

Imagen 80. Ejemplo elemento thead.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_thead.asp]

1.5.9.7 Tfoot

Definición:

El elemento **tfoot** representa un bloque de filas que se componen de los sumarios de columna (pies de página) para el elemento **table** padre, si el elemento **tfoot** tuviera un padre y un elemento **table**.

El elemento **tfoot** participa en el modelo de tabla.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

El ejemplo se puede observar en la imagen anterior (imagen 80).

1.5.9.8 Tr

Definición:

El elemento **tr** representa una fila de celdas en una tabla.

El elemento **tr** participa en el modelo de tabla.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<table border="1">
<tr>
<th>Month</th>
<th>Savings</th>
</tr>
<tr>
<td>January</td>
<td>$100</td>
</tr>
</table>
```

Imagen 81. Ejemplo elemento tr.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_tr.asp]

1.5.9.9 Td

Definición:

El elemento **td** representa una celda de datos en una tabla.

El elemento **td** y sus atributos **colspan**, **rowspan** y **headers** participan en el modelo de tabla.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Colspan	Especifica el número de columnas que una celda debe abarcar	Un número.
Rowspan	Establece el número de filas que una celda debe abarcar	Un número
Headers	Especifica una lista separada por espacios de celdas de encabezado que contienen información acerca de esta celda. El valor debe coincidir con el identificador de la celda de cabecera (que se establece mediante el atributo id). Este atributo es útil para navegadores no visuales.	Id de celda de encabezado (<i>header_id</i>)

Tabla 38. Atributos elemento td.

Compatible en los navegadores:



Ejemplo:

```
<table border="1">
  <tr>
    <td>Cell A</td>
    <td>Cell B</td>
  </tr>
</table>
```

Imagen 82. Ejemplo elemento td.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_td.asp]

1.5.9.10 Th

Definición:

El elemento **th** representa una celda de encabezado en una tabla.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Colspan	Especifica el número de columnas que una celda de encabezado debe abarcar	Un número
rowspan	Especifica el número de filas que una celda de encabezado debe abarcar	Un número
Headers	Especifica una lista separada por espacios de celdas de encabezado que contienen información acerca de esta celda. El valor debe coincidir con el identificador de la celda de cabecera (que se establece mediante el atributo id). Este atributo es útil para navegadores no visuales.	Id de celda de encabezado (<i>header_id</i>)
Scope	Especifica si una celda de encabezado es un encabezado de una columna, fila o grupo de columnas o filas.	Col, colgroup, row, rowgroup.
Abbr	Especifica una versión abreviada del contenido de una celda de encabezado	Texto

Tabla 39. Atributos elemento th.

Compatible en los navegadores:



Ejemplo:

```
<table>
<thead>
  <tr> <th> ID <th> Measurement <th> Average <th> Maximum
</thead>
<tbody>
  <tr> <td> <th scope=rowgroup> Cats <td> <td>
  <tr> <td> 93 <th scope=row> Legs <td> 3.5 <td> 4
  <tr> <td> 10 <th scope=row> Tails <td> 1 <td> 1
</tbody>
<tbody>
  <tr> <td> <th scope=rowgroup> English speakers <td> <td>
  <tr> <td> 32 <th scope=row> Legs <td> 2.67 <td> 4
  <tr> <td> 35 <th scope=row> Tails <td> 0.33 <td> 1
</tbody>
</table>
```

Imagen 83. Ejemplo elemento th.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-th-element>]

1.5.10 Formularios

1.5.10.1 Form

Definición:

El elemento **form** representa una colección de elementos asociados a formularios, algunos de los cuales pueden representar valores editables que se pueden enviar a un servidor para su procesamiento.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Accept-charset	Especifica la codificación de caracteres que se van a utilizar para el envío del formulario	codificación de caracteres (<i>character_set</i>)
Action	Especifica dónde enviar el formulario de datos (<i>form-data</i>) cuando se envía un formulario	URL
Autocomplete	Especifica si un formulario debe tener autocompletar activado o desactivado.	On, off.
Enctype	Especifica cómo se codificará la los datos del formulario (<i>form-data</i>) cuando se envía al servidor (sólo para method = "post")	application/x-www-form-urlencoded, multipart/form-data, text/plain.
Method	Especifica el método HTTP a utilizar al enviar los datos de un formulario (<i>form-data</i>)	Get, post.
Name	Especifica el nombre de un formulario	Texto
Novalidate	Especifica que el formulario no debe ser validado cuando se trate de enviar el formulario (<i>submitted</i>)	Novalidate
Target	Especifica dónde se debe mostrar la respuesta que se recibe después de enviar el formulario	_blank, _self, _parent, _top.

Tabla 40. Atributos elemento form.

Compatible en los navegadores:



Ejemplo:

```
<form action="http://www.google.com/search" method="get">
  <label>Google: <input type="search" name="q"></label> <input type="submit" value="Search...">
</form>
<form action="http://www.bing.com/search" method="get">
  <label>Bing: <input type="search" name="q"></label> <input type="submit" value="Search...">
</form>
```

Imagen 84. Ejemplo elemento form.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-form-element>]

1.5.10.2 Fieldset

Definición:

El elemento **fieldset** representa un conjunto de controles de formulario opcionalmente agrupados bajo un nombre común.

El nombre del grupo está dado por el primer elemento **legend** que es un hijo del elemento **fieldset**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Disabled	Especifica que un grupo de elementos de formulario relacionados deben ser desactivados	Disabled
Form	Especifica uno o más formularios al que pertenece el fieldset .	ID del formulario (<i>form_id</i>)
Name	Especifica el nombre del fieldset	Texto

Tabla 41. Atributos elemento fieldset.

Compatible en los navegadores:



Ejemplo:

```
<fieldset name="clubfields" disabled>
<legend> <label>
  <input type=checkbox name=club onchange="form.clubfields.disabled = !checked">
  Use Club Card
</label> </legend>
<p><label>Name on card: <input name=clubname required></label></p>
<p><label>Card number: <input name=clubnum required pattern="[-0-9]+"></label></p>
<p><label>Expiry date: <input name=clubexp type=month></label></p>
</fieldset>
```

Imagen 85. Ejemplo elemento fieldset.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-fieldset-element>]

1.5.10.3 Legend

Definición:

El elemento **legend** representa un título o leyenda para los elementos **fieldset**.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<form>
  <fieldset>
    <legend>Personalia:</legend>
    Name: <input type="text" size="30"><br>
    Email: <input type="text" size="30"><br>
    Date of birth: <input type="text" size="10">
  </fieldset>
</form>
```

Imagen 86. Ejemplo elemento legend.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_legend.asp]

1.5.10.4 Label

Definición:

El elemento **label** define una etiqueta o título para un control de formulario. El título debe estar asociado a ese control del formulario.

El elemento **label** no hace nada especial para el usuario. Sin embargo, proporciona una mejora de uso del ratón para los usuarios, ya que si el usuario hace clic en el texto dentro del elemento **label**, cambia o se alterna con el control asociado.

El atributo **for** del elemento **label** debe ser igual al atributo **id** del control relacionado para atarlos o asociarlos juntos.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Form	Especifica uno o más formularios al que pertenece el label .	ID del formulario (<i>form_id</i>)
For	Especifica el control de formulario a la que este elemento pertenece. Este valor debe ser el mismo que el valor del ID del control asociado.	ID del elemento o control de formulario asociado (<i>element_id</i>)

Tabla 42. Atributos elemento label.

Compatible en los navegadores:



Ejemplo:

```
<form action="demo_form.asp">
  <label for="male">Male</label>
  <input type="radio" name="sex" id="male" value="male"><br>
  <label for="female">Female</label>
  <input type="radio" name="sex" id="female" value="female"><br>
  <input type="submit" value="Submit">
</form>
```

Imagen 87. Ejemplo elemento label.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_label.asp]

1.5.10.5 Input

Definición:

El elemento **input** representa un tipo de campo de datos, por lo general con un control de formulario (form), para permitir que el usuario edite los datos.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Accept	Especifica los tipos de archivo que el servidor acepta (solo para type="file")	Audio/*, Video/*, Image/*, Tipo MIME (<i>MIME_type</i>).
Alt	Especifica un texto alternativo para las imágenes (sólo para type = "image")	Texto
Autocomplete	Evita que el navegador de el uso de "autocompletar" en este control. Puede ser útil para los datos confidenciales, como datos bancarios, etc	On, off.
Autofocus	Especifica que un elemento <input> automáticamente debe recibir el foco cuando se carga la página	autofocus

Checked	Especifica que un elemento <input> debe ser pre-seleccionado cuando se cargue la página (para type = "checkbox" o type = "radio")	Checked
Dirname	Determina la dirección del texto presentado en el campo de entrada.	El valor de este atributo puede ser una cadena de texto, como un nombre para el campo. Por ejemplo, si se especifica el atributo como dirname = "text_dir", una vez que el formulario ha sido enviado, los datos podrían tener este aspecto: text_dir = ltr.
Disabled	Especifica que un elemento <input> deben ser desactivados	Disabled
Form	Especifica uno o más formularios al que pertenece el input .	ID del formulario (<i>form_id</i>)
Formaction	Especifica la dirección URL del archivo que va a procesar el control de entrada cuando se envía el formulario (para type = "submit" y tipo "image" =)	URL
Formenctype	Especifica cómo el formulario de datos se debería codificar antes de enviarlo a un servidor. Sólo para type = "submit" y type =	application/x-www-form-urlencoded, multipart/form-

	“image”	data, text/plain.
Formmethod	Define el método HTTP para enviar datos a la URL de la acción (para type = "submit" y type = "image")	Get, post
Formnovalidate	Define que los elementos del formulario no deben ser validados cuando se intente enviar los datos al servidor (submitted)	Formnovalidate
Formtarget	Especifica dónde se debe mostrar la respuesta que se recibe después de enviar el formulario (para type = "submit" y type = "image")	_blank, _selft, _parent, _top, Nombre de un frame (<i>frame_name</i>)
Height	Especifica el alto de un elemento <input> (sólo para type = "image")	Tamaño en píxeles
List	Se refiere a un elemento de <datalist> que contiene opciones predefinidas para un elemento <input>	ID de un elemento datalist (<i>datalist_id</i>)
Max	Especifica el valor máximo para un elemento <input>	Una válida cadena de fecha (date), Un número.
Maxlength	Especifica el número máximo de caracteres permitido en un elemento <input>	Un número
Min	Especifica el valor mínimo para un elemento <input>	Una válida cadena de fecha (date), Un número.
Multiple	Especifica que el usuario puede introducir más de un valor en un elemento <input>	Multiple

Name	Especifica el nombre de un elemento <input>	Un nombre
Pattern	Especifica una expresión regular con la que el valor de un elemento <input> es chequeado o comprobado su validez (<i>checked</i>)	Expresión regular (<i>regexp</i>)
Placeholder	Especifica una pista corta que describe el valor esperado de un elemento <input>	Texto
Readonly	Especifica que un campo de entrada (input) es de sólo lectura	Readonly
Required	Especifica que un campo de entrada (input) debe ser completado antes de enviar el formulario.	Required
Size	Especifica el ancho, en caracteres, de un elemento <input>	Un número
Src	Especifica la dirección URL de la imagen que desea utilizar como un botón de envío (sólo para type = "image")	URL
Step	Especifica los intervalos de números legales (permitidos) para un campo de entrada.	Un número
Type	Especifica el tipo de elemento <input> para mostrar.	Hidden, Text, Search, Tel, url, email, password, datetime, date, month, week, time, datetime-local,

		number, range, color, checkbox, radio, file, submit, image, reset, button.
Value	Especifica el valor de un elemento <input>	Texto
Width	Especifica el ancho de un elemento <input> (sólo para type = "image")	Tamaño en píxeles.

Tabla 43. Atributos elemento input.

Como se observa en la tabla anterior, existe el atributo **type** que controla el tipo de datos del elemento. Dependiendo del tipo de datos del elemento se podrá o no utilizar algunos atributos para poder controlar el elemento.

No todos los principales navegadores soportan todos los tipos de **input** nuevos. Sin embargo, se puede usar esos **inputs** en los que si son compatibles, y si no lo son, se comportarán como campos de texto normales.

A continuación se detalla cada tipo de datos del elemento **input**, con los atributos que se puede utilizar dependiendo de su tipo:

1.5.10.5.1 Input (type=hidden)

Definición:

Define un campo oculto (no visible para un usuario). Un campo oculto suele almacenar un valor predeterminado, o puede obtener su valor por un script.

Atributos

- ✓ Atributos globales
- ✓ Value

Compatibilidad en los navegadores



Ejemplo:

```
<input type="hidden" name="country" value="Norway">
```

Imagen 88. Ejemplo elemento input tipo hidden.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.2 Input (type=text)

Definición:

Default. Define un campo de entrada texto de una sola línea (ancho predeterminado es de 20 caracteres).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ Dirname
- ✓ List
- ✓ Maxlength
- ✓ Pattern
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Size

Compatibilidad en los navegadores



Ejemplo:

```
First name: <input type="text" name="fname"><br>  
Last name: <input type="text" name="lname"><br>
```

Imagen 89. Ejemplo elemento input tipo text.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.3 Input (type=search)

Definición:

El tipo **search** se utiliza para campos de búsqueda (un campo de búsqueda se comporta como un campo de texto normal).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ Dirname
- ✓ List
- ✓ Maxlength
- ✓ Pattern
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Size

Compatibilidad en los navegadores



Ejemplo:

```
Search Google: <input type="search" name="googlesearch">
```

Imagen 90. Ejemplo elemento input tipo search.

Fuente: W3 Schools [Disponible en:

http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.4 Input (type=tel)

Definición:

Define un campo para introducir un número de teléfono.

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Maxlength
- ✓ Pattern
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Size

Compatibilidad en los navegadores



Ejemplo:

```
Telephone: <input type="tel" name="usrtel">
```

Imagen 91. Ejemplo elemento input tipo tel.

Fuente: W3 Schools [Disponible en:

http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.5 Input (type=url)

Definición:

Define un campo para introducir una URL.

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Maxlength
- ✓ Pattern
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Size

Compatibilidad en los navegadores



Ejemplo:

```
Add your homepage: <input type="url" name="homepage">
```

Imagen 92. Ejemplo elemento input tipo url.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.6 Input (type=email)

Definición:

Define un campo para una dirección de e-mail.

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Maxlength
- ✓ Multiple
- ✓ Pattern
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Size

Compatibilidad en los navegadores



Ejemplo:

```
E-mail: <input type="email" name="usremail">
```

Imagen 93. Ejemplo elemento input tipo email.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.7 Input (type=password)

Definición:

Define un campo de contraseña (caracteres se enmascaran).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List

- ✓ Maxlength
- ✓ Pattern
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Size

Compatibilidad en los navegadores



Ejemplo:

```
<input type="password" name="pwd">
```

Imagen 94. Ejemplo elemento input tipo password.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.8 Input (type=datetime)

Definición:

Define un control de fecha y hora (año, mes, día, hora, minuto, segundo y fracción de segundo, basado en la zona horaria UTC).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Readonly
- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Birthday (date and time): <input type="datetime" name="bdaytime">
```

Imagen 95. Ejemplo elemento input tipo datetime.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.9 input (type=date)

Definición:

Define un control de fecha (año, mes y día (no hay tiempo)).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Readonly
- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Birthday: <input type="date" name="bday">
```

Imagen 96. Ejemplo elemento input tipo date.

Fuente: W3 Schools [Disponible en:

http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.10 input (type=month)

Definición:

Define un control de un mes y un año (sin zona horaria).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Readonly
- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Birthday (month and year): <input type="month" name="bdaymonth">
```

Imagen 97. Ejemplo elemento input tipo month.

Fuente: W3 Schools [Disponible en:

http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.11 input (type=week)

Definición:

Define un control de una semana y un año (no zona horaria).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Readonly
- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Select a week: <input type="week" name="week_year">
```

Imagen 98. Ejemplo elemento tipo week.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.12 input (type=time)

Definición:

Define un control para ingresar una hora (sin zona horaria)

Atributos

- ✓ Atributos globales

- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Readonly
- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Select a time: <input type="time" name="usr_time">
```

Imagen 99. Ejemplo elemento input tipo time.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.13 input (type=datetime-local)

Definición:

Define un control de fecha y hora (año, mes, día, hora, minuto, segundo y fracciones de segundo (sin zona horaria)).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Readonly

- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Birthday (date and time): <input type="datetime-local" name="bdaytime">
```

Imagen 100. Ejemplo elemento input datetime-local.

Fuente: W3 Schools [Disponible en:

http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.14 input (type=number)

Definición:

Define un control para introducir un número.

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Placeholder
- ✓ Readonly
- ✓ Required
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5">
```

Imagen 101. Ejemplo elemento input tipo number.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.15 input (type=range)

Definición:

Define un control para introducir un número cuyo valor exacto no es importante (como un control deslizante).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List
- ✓ Max
- ✓ Min
- ✓ Step

Compatibilidad en los navegadores



Ejemplo:

```
<input type="range" name="points" min="1" max="10">
```

Imagen 102. Ejemplo elemento input tipo range.

Fuente: W3 Schools [Disponible en:
http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.16 input (type=color)

Definición:

Define un control de selector de colores.

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Autocomplete
- ✓ List

Compatibilidad en los navegadores



Ejemplo:

```
Select your favorite color: <input type="color" name="favcolor">
```

Imagen 103. Ejemplo elemento input tipo color.

Fuente: W3 Schools [Disponible en:

http://www.w3schools.com/html/html5_form_input_types.asp]

1.5.10.5.17 input (type=checkbox)

Definición:

Define una casilla de verificación.

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Checked
- ✓ Required

Compatibilidad en los navegadores



Ejemplo:

```
<input type="checkbox" name="vehicle" value="Bike"> I have a bike<br>
<input type="checkbox" name="vehicle" value="Car"> I have a car
```

Imagen 104. Ejemplo elemento tipo checkbox.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.18 input (type=radio)

Definición:

Define un botón de opción (radio button).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Checked
- ✓ Required

Compatibilidad en los navegadores



Ejemplo:

```
<input type="radio" name="gender" value="male"> Male<br>
<input type="radio" name="gender" value="female"> Female
```

Imagen 105. Ejemplo elemento input tipo radio.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.19 input (type=file)

Definición:

Define un campo de selección de archivos y un botón "Examinar" (para la carga de archivos).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Accept
- ✓ Multiple
- ✓ Required

Compatibilidad en los navegadores



Ejemplo:

```
Select a file: <input type="file" name="img">
```

Imagen 106. Ejemplo elemento tipo file.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.20 input (type=submit)

Definición:

Define un botón de envío (para enviar los datos de un formulario al servidor).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Formaction
- ✓ Formenctype
- ✓ Formmethod

- ✓ Formnovalidate
- ✓ Formtarget

Compatibilidad en los navegadores



Ejemplo:

```
<input type="submit">
```

Imagen 107. Ejemplo elemento input tipo submit.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.21 input (type=image)

Definición:

Define una imagen como botón submit. (input type=submit).

Atributos

- ✓ Atributos globales
- ✓ Value
- ✓ Alt
- ✓ Formaction
- ✓ Formenctype
- ✓ Formmethod
- ✓ Formnovalidate
- ✓ Formtarget
- ✓ Height
- ✓ Src
- ✓ Width

Compatibilidad en los navegadores



Ejemplo:

```
<input type="image" src="img_submit.gif" alt="Submit">
```

Imagen 108. Ejemplo elemento input tipo image.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.22 input (type=reset)

Definición:

Define un botón de reinicio (restablece todos los valores del formulario a valores por defecto)

Atributos

- ✓ Atributos globales
- ✓ Value

Compatibilidad en los navegadores



Ejemplo:

```
<input type="reset">
```

Imagen 109. Ejemplo elemento input tipo reset.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.5.23 input (type=button)

Definición:

Define un botón para pulsarlo (en su mayoría se utiliza con un JavaScript para activar una secuencia de comandos).

Atributos

- ✓ Atributos globales
- ✓ Value

Compatibilidad en los navegadores



Ejemplo:

```
<input type="button" value="Click me" onclick="msg()">
```

Imagen 110. Ejemplo elemento input tipo button.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/att_input_type.asp]

1.5.10.6 button

Definición:

El elemento **button** representa un botón etiquetado por su contenido. El elemento es un botón.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Autofocus	Especifica que un botón automáticamente obtendrá el foco (focus) cuando se carga la página.	Autofocus
Disabled	Especifica que un botón se debe desactivar	Disabled
Form	Especifica uno o más formularios al que pertenece el button .	ID del formulario (<i>form_id</i>)
Formaction	Especifica dónde enviar el formulario de datos cuando se trata de enviar un formulario. Sólo para <code>type = "submit"</code>	URL
Formenctype	Especifica cómo el formulario de datos se debería codificar antes de enviarlo a un servidor.	application/x-www-form-urlencoded,

	Sólo para type = "submit"	multipart/form-data, text/plain.
Formmethod	Especifica cómo enviar el formulario de datos (el método HTTP a utilizar). Sólo para type= "submit".	Get, post
Formnovalidate	Especifica que el tipo de datos no debe ser validado en el intento de enviar un formulario al servidor. Sólo para type= "submit".	Formnovalidate
Formtarget	Especifica dónde se debe mostrar la respuesta después de enviar el formulario. Sólo para type= "submit".	_blank, _self, _parent, _top, Nombre de un frame (<i>frame_name</i>)
Name	Especifica un nombre para el botón.	Un nombre
Type	Especifica el tipo de botón.	Button, Reset, Submit.
Value	Especifica un valor inicial para el button .	Texto

Tabla 44. Atributos elemento button.

Compatible en los navegadores:



Ejemplo:

```
<button type=button
  onclick="alert('This 15-20 minute piece was composed by George Gershwin.')">
  Show hint
</button>
```

Imagen 111. Ejemplo elemento button.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-button-element>]

1.5.10.7 Select

Definición:

El elemento **select** representa un control para seleccionar entre un conjunto de opciones.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Autofocus	Especifica que la lista desplegable automáticamente debe recibir el foco (focus) cuando se carga la página.	Autofocus
Disabled	Especifica que una lista desplegable se debe desactivar.	Disabled
Form	Especifica uno o más formularios al que pertenece el select .	ID de formulario (<i>form_id</i>)
Multiple	Especifica que las opciones se pueden seleccionar varias a la vez	Multiple
Name	Define un nombre para la lista desplegable	Un nombre
Required	El usuario deberá seleccionar un valor antes de enviar el formulario.	Required
Size	Define el número de opciones visibles en una lista desplegable	Un número

Tabla 45. Atributos elemento select.

Compatible en los navegadores:



Ejemplo:

```
<p>
  <label for="unitttype">Select unit type:</label>
  <select id="unitttype" name="unitttype">
    <option value="1"> Miner </option>
    <option value="2"> Puffer </option>
    <option value="3" selected> Snipey </option>
    <option value="4"> Max </option>
    <option value="5"> Firebot </option>
  </select>
</p>
```

Imagen 112. Ejemplo elemento select.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-select-element>]

1.5.10.8 Datalist

Definición:

El elemento **datalist** representa una lista de opciones predefinidas, compuestas por un elemento **option** para un elemento **input**.

El elemento **datalist** se utiliza para proporcionar un "autocompletar" en función de los elementos **input**. Los usuarios podrán ver una lista desplegable de opciones predefinidas como entrada de datos.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

```
<input list="browsers">

<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Imagen 113. Ejemplo elemento datalist.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_datalist.asp]

1.5.10.9 Optgroup

Definición:

El elemento **optgroup** representa un grupo de elementos **option** con una etiqueta común.

El elemento de grupo de elementos **option** se compone de los elementos **option** que son hijos del elemento **optgroup**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Disabled	Especifica que una opción de grupo (<i>option-group</i>) debe ser desactivado	Disabled
Label	Especifica una etiqueta para una opción de grupo (<i>option-group</i>)	Texto

Tabla 46. Atributos elemento optgroup.

Compatible en los navegadores:



Ejemplo:

```
<select>
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

Imagen 114. Ejemplo elemento optgroup.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_optgroup.asp]

1.5.10.10 Option

Definición:

El elemento **option** representa una opción en un elemento **select** o como parte de una lista de sugerencias en un elemento **datalist**.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Disabled	Especifica que un option se desactivará	disabled
Label	Especifica una etiqueta corta para un option	texto
Selected	Especifica que el option debe ser pre-seleccionado cuando se cargue la página.	Selected
Value	Especifica el valor que se envía a un servidor.	value

Tabla 47. Atributos elemento option.

Compatible en los navegadores:



Ejemplo:

```
<select>
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="opel">Opel</option>
  <option value="audi">Audi</option>
</select>
```

Imagen 115. Ejemplo elemento option.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_option.asp]

1.5.10.11 Textarea

Definición:

El elemento **textarea** representa un control de texto de varias líneas de entrada.

Un área de texto (**textarea**) puede contener un número ilimitado de caracteres.

El tamaño de un **textarea** puede ser especificado por los atributos cols y rows, o mejor aún, a través de CSS y propiedades de ancho.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Autofocus	Especifica que un área de texto (textarea) debería automáticamente obtener el foco (focus) cuando se carga la página	Autofocus
Cols	Especifica el ancho visible de un área de texto.	Un número
Dirname	Especifica el nombre de un campo de entrada (input) que contiene la dirección del texto del área de texto. El valor del campo de entrada (input) que contiene la dirección del	Nombre de un campo de entrada (nombre de input)

	texto debe ser "ltr" o "rtl".	
Disabled	Especifica que un área de texto debe ser desactivada o deshabilitada.	Disabled
Form	Especifica uno o más formularios al que pertenece el textarea .	ID de formulario (<i>form_id</i>)
Maxlength	Especifica el número máximo de caracteres permitidos en el área de texto	Un número
Name	Especifica el nombre de un área de texto	Texto
Placeholder	Especifica una pista corta que describe el valor esperado de un área de texto	Texto
Readonly	Especifica que un área de texto debe ser de sólo lectura	Readonly
Required	Especifica que un área de texto se necesita o debe ser llenado obligatoriamente.	Required
Rows	Especifica el número de líneas visibles en un área de texto	Un número
Wrap	Especifica cómo el texto del textarea debe envolver a la línea siguiente.	Hard, soft

Tabla 48. Atributos elemento textarea.

Compatible en los navegadores:



Ejemplo:

```
<textarea rows="4" cols="50">
At w3schools.com you will learn how to make a website. We offer free
tutorials in all web development technologies.
</textarea>
```

Imagen 116. Ejemplo elemento textarea.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_textarea.asp]

1.5.10.12 Keygen

Definición:

El elemento **keygen** representa un control de generador de par de claves (key-pair). Cuando el control en un formulario es enviado, una clave privada se almacena en el almacén de claves local, y la clave pública se empaqueta y se envía al servidor.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Autofocus	Especifica que un elemento keygen automáticamente se enfoque (focus) cuando se carga la página.	Autofocus
Challenge	Especifica que el valor del elemento keygen debe ser cambiado cuando se trate de enviar el formulario (submitted).	Challenge
Disabled	Especifica que un elemento keygen debe ser desactivado.	Disabled
Form	Especifica uno o más formularios al que pertenece el keygen .	ID de formulario (<i>form_id</i>)
Keytype	Especifica el algoritmo de seguridad de la clave	Rsa, dsa, ec.
Name	Especifica el nombre del elemento keygen .	Un nombre

Tabla 49. Atributos elemento keygen.

Compatible en los navegadores:



Ejemplo:

```
<form action="processkey.cgi" method="post" enctype="multipart/form-data">
  <p><keygen name="key"></p>
  <p><input type="submit" value="Submit key..."></p>
</form>
```

Imagen 117. Ejemplo elemento keygen.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-keygen-element>]

1.5.10.13 Output

Definición:

El elemento **output** representa el resultado de un cálculo.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
For	Especifica la relación entre el resultado del cálculo, y los elementos utilizados en el cálculo	ID del elemento/s relacionado (<i>element_id</i>)
Form	Especifica uno o más formularios al que pertenece el output .	ID de formulario (<i>form_id</i>)
Name	Especifica el nombre del elemento output .	

Tabla 50. Atributos elemento output.

Compatible en los navegadores:



Ejemplo:

```
<form oninput="x.value=parseInt (a.value)+parseInt (b.value) ">0
  <input type="range" name="a" value="50">100
  +<input type="number" name="b" value="50">
  =<output name="x" for="a b"></output>
</form>
```

Imagen 118. Ejemplo elemento output.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_output.asp]

1.5.10.14 Progress

Definición:

El elemento **progress** representa el progreso finalización de una tarea. El progreso puede ser o no indeterminado, lo que indica que se están haciendo progresos, pero no está claro cuánto todavía queda por hacer antes de que se complete la tarea.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Value	Especifica cuánto de la tarea se ha completado.	Un número
Max	Especifica la cantidad de trabajo que la tarea requiere en total	Un número

Tabla 51. Atributos elemento progress.

Compatible en los navegadores:



Ejemplo:

```
<progress value="22" max="100"></progress>
```

Imagen 119. Ejemplo elemento progress.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_progress.asp]

1.5.10.15 Meter

Definición:

El elemento **meter** representa una medida escalar dentro de un rango conocido, o un valor fraccionario, por ejemplo de uso de disco, la relevancia del resultado de una consulta, o la fracción de población votante que ha seleccionado a un candidato en particular, etc.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Value	Obligatorio. Especifica el valor actual de la medición (meter)	Un número
Min	Especifica el valor mínimo del rango.	Un número
Max	Especifica el valor máximo del rango.	Un número
Low	Especifica el rango que se considera que es un valor bajo	Un número
High	Especifica el rango que se considera que es un valor alto	Un número
Optimum	Especifica qué valor es el valor óptimo para el indicador o medición.	Un número

Tabla 52. Atributos elemento meter.

Compatible en los navegadores:



Ejemplo:

```
<p>The grapefruit pie had a radius of 12cm and a height of 2cm.</p>
<dl>
  <dt>Radius: <dd> <meter min=0 max=20 value=12>12cm</meter>
  <dt>Height: <dd> <meter min=0 max=10 value=2>2cm</meter>
</dl>
```

Imagen 120. Ejemplo elemento meter.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-meter-element>]

1.5.11 Elementos interactivos

1.5.11.1 Details

Definición:

El elemento **details** representa detalles, que el usuario puede ver u ocultar.

Puede ser utilizado en conjunto con el elemento **summary** para proporcionar un encabezamiento que se puede hacer clic para expandir/contraer los detalles según se requiera.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Open	Especifica que los datos deben ser visibles (abierto) al usuario	open

Tabla 53. Atributos elemento details.

Compatible en los navegadores:



Ejemplo:

```
<details>
  <summary><label for=fn>Name & Extension:</label></summary>
  <p><input type=text id=fn name=fn value="Pillar Magazine.pdf">
  <p><label><input type=checkbox name=ext checked> Hide extension</label>
</details>
```

Imagen 121. Ejemplo elemento details.

Fuente: Consorcio W3C [Disponible en: <http://dev.w3.org/html5/spec/single-page.html#the-details-element>]

1.5.11.2 Summary

Definición:

El elemento **summary** define un título visible para el elemento **details**. En el título se puede hacer clic para ver/ocultar los detalles.

Atributos:

- ✓ Atributos globales

Compatible en los navegadores:



Ejemplo:

El ejemplo se puede observar en la imagen anterior (imagen 121).

1.5.11.3 Command

Definición:

El elemento **command** representa un comando que el usuario puede invocar.

El elemento **command** define un comando (un **radiobutton**, **checkbox** o un botón de comando) que el usuario puede invocar.

Un comando puede ser parte de un menú contextual o la barra de herramientas, utilizando el elemento **menu**, o se puede poner en cualquier otro lugar de la página, para definir un atajo de teclado.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Type	Especifica el tipo de comando	Checkbox, Command, Radio.
Label	Obligatorio. Especifica el nombre de el command , como se muestra al usuario.	Texto
Icon	Especifica una imagen que representa el comando	URL
Disabled	Especifica que el command debe estar desactivado o deshabilitado.	disabled
Checked	Especifica que command debe estar marcado (<i>checked</i>) cuando se carga la página. Sólo para type = "radio" o type = "checkbox"	checked
Radiogroup	Especifica el nombre del grupo de comandos que ser activadas cuando el propio comando se activa. Sólo para type = "radio"	Nombre del grupo (<i>group_name</i>)

Tabla 54. Atributos elemento command.

Compatible en los navegadores:



Ejemplo:

```
<menu>  
<command type="command" label="Save" onclick="save()">Save</command>  
</menu>
```

Imagen 122. Ejemplo elemento command.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_command.asp]

1.5.11.4 Menu

Definición:

El elemento **menu** define una lista/menú de comandos.

Se utiliza para los menús contextuales, barras de herramientas y los controles de formulario de lista y los comandos.

Atributos:

- ✓ Atributos globales

Nombre	Descripción	Valores
Type	Especifica el tipo de menú para mostrar. El valor por defecto es "list"	Context, toolbar, list.
Label	Especifica una etiqueta o título visible para el menú	Texto

Tabla 55. Atributos elemento menu.

Compatible en los navegadores:



Ejemplo:

```
<menu type="toolbar">
<li>
<menu label="File">
<button type="button" onclick="file_new()">New...</button>
<button type="button" onclick="file_open()">Open...</button>
<button type="button" onclick="file_save()">Save</button>
</menu>
</li>
<li>
<menu label="Edit">
<button type="button" onclick="edit_cut()">Cut</button>
<button type="button" onclick="edit_copy()">Copy</button>
<button type="button" onclick="edit_paste()">Paste</button>
</menu>
</li>
</menu>
```

Imagen 123. Ejemplo elemento menu.

Fuente: W3 Schools [Disponible en: http://www.w3schools.com/tags/tag_menu.asp]

CAPÍTULO 2

MONGODB

2.1 Introducción

En este capítulo lo que se trata de desarrollar es un manual para poder manejar la base de datos NoSQL denominada MongoDB. El objetivo no es aprender MongoDB desde su propia consola de comandos, sino algo mucho más práctico como lo es incluir MongoDB en un lenguaje de programación, en este caso se aprenderá a utilizar esta base de datos en el lenguaje PHP.

Al igual que el capítulo anterior, todo lo que se vaya explicando se detallará con ejemplos claros. Ejemplos con imágenes de cómo codificar consultas a la base de datos desde el lenguaje de programación antes mencionado, PHP.

2.2 ¿Qué es NoSQL?

Como se mencionó anteriormente MongoDB es una base de datos NoSQL, antes de comenzar a definir detalladamente de que se trata MongoDB, se esclarecerá un poco más el término NoSQL.

Cuando se habla de NoSQL, no es solo referirse a un tipo de base de datos, sino a varios tipos o métodos para almacenar datos, dichos métodos no cumplen con el esquema típico entidad-relación, y no imponen una estructura de datos fija en forma de tablas o relaciones. Estos tipos de base de datos permiten almacenar los datos en diferentes formatos como: llave-valor, documentos, xml, grafos, columnas, entre otros.

La principal característica de NoSQL, es que su sistema de almacenamiento de datos está pensado para el manejo de grandes cantidades de información de una manera muy ágil, escalando de manera horizontal, aumentando el rendimiento cuando se necesite crecer, y no como las base de datos relacionales que escalan o crecen de

manera vertical, es decir, añadiendo más memoria o un disco duro con mayor velocidad cuando se necesite crecer en almacenamiento de datos.

NoSQL no garantiza las características ACID (Aislamiento, Consistencia/Integridad, Atomicidad, Durabilidad) ya que ningún tipo de sus base de datos implementa transacciones (conjunto de operaciones que se ejecutan en forma indivisible, sus operaciones no finalizan en un estado intermedio, manteniendo la integridad de los datos), por lo tanto los cambios no se pueden volver a atrás. La consistencia de los datos es eventual (por ejemplo: a veces 2 clientes no ven los mismos datos), esto se conoce como BASE (Consistencia Eventual Flexible Básicamente Disponible), donde NoSQL le pone más atención a la disponibilidad de sus datos.

Ninguna base de datos NoSQL implementa aislamiento de datos, pero son tolerantes a fallos y redundancia en sus datos. Mejoran el rendimiento al no realizar una traducción SQL de cada sentencia para ser ejecutadas.

2.3 ¿Qué es MONGODB?

MongoDB es un sistema de base de datos NoSQL (No solo SQL, no relacional), éste término NoSQL como se mencionó anteriormente, indica diferentes tipos de base de datos, que se diferencian del típico modelo relacional de las bases de datos SQL, ya que la mayoría de estas bases de datos, por no decir todas, usan otros métodos para reemplazar a las sentencias SQL, y además los datos almacenados ya no necesitan estructuras fijas (tablas), ahora su estructura puede variar en cualquier momento, como es el caso de MongoDB, que utiliza colecciones (equivalente a tablas) y sus documentos (equivale a registros) pueden variar dinámicamente.

MongoDB es un sistema de base de datos orientada a documentos, esto significa que cada registro puede tener un esquema de datos diferente uno de otro. Está escrito en el lenguaje de C++, de código abierto y funciona tranquilamente en sistemas operativos como Windows, Linux, OS X.

MongoDB fue creado por Dwight Merriman y Eliot Horowitz, quienes se basaron en sus experiencias de construcción a gran escala y alta disponibilidad. MongoDB mantiene muchas de las características de una base de datos relacional.

MongoDB fue creado por el ex fundador y director de tecnología de DoubleClick Dwight Merriman y el ex DoubleClick ingeniero y fundador y CTO ShopWiki Eliot Horowitz. Se basó en sus experiencias construcción a gran escala, alta disponibilidad, sistemas robustos para crear un nuevo tipo de base de datos. MongoDB mantiene muchas de las grandes características de una base de datos relacional - como índices y consultas dinámicas. Pero al cambiar el modelo de datos relacional a partir documento orientado a obtener muchas ventajas, como una mayor agilidad a través de esquemas flexibles y fáciles de escalabilidad horizontal.

“Características importantes en MongoDB son su velocidad, el sencillo manejo de consultas (manejabilidad), el libre esquema de datos, escalamiento horizontal sin afectar el rendimiento, sin añadir recursos. MongoDB puede correr como una aplicación de 32bits ó 64 bits, se recomienda correr como una aplicación de 64 ya que Mongo cuando corre en 32 bits se limita a una base de datos alrededor de 2GB (Giga Bytes).”³

“El objetivo de MongoDB es tener la mayor funcionalidad posible pero sin perder la escalabilidad.”⁴

Para tener una idea clara sobre cómo interpretar MongoDB, se puede decir que un equivalente a un registro de base de datos SQL en MongoDB se denomina documento, la agrupación de documentos se conoce con el nombre de colección, que en base de datos SQL se las conoce como tabla.

Ahora teniendo una idea de lo que es un documento y una colección se puede decir que:

- Un sistema Mongo contiene un conjunto de base de datos.
- Una base de datos contiene un conjunto de colecciones.
- Una colección contiene un conjunto de documentos.

³ mongoDB. Hannan, Tony. 15 de Septiembre de 2011. 15 de Noviembre de 2012
<<http://es.wiki.mongodb.org/display/DOCS/Introduction>>.

⁴ La compañía MongoDB 10gen. Stam, Robert. 13 de Septiembre de 2012. 15 de Noviembre de 2012
<<http://www.10gen.com/events/webinar/Introduccion-al-NoSQL-y-MongoDB>>.

- Un documento es un conjunto de campos.
- Un campo es un par clave/valor.
- Una clave es un nombre (cadena).
- Un valor es:
 - Un tipo cadena, entero, binario, etc.
 - Un documento o un vector de valores.

Los documentos mencionados anteriormente se almacenan en la base de datos como un formato BSON (Binary JSON), que es una versión modificada de JSON (Notación de Objetos JavaScript para intercambio de datos). Aunque los datos se guardarán en formato BSON, en la práctica no se trabajará como un formato BSON, simplemente se mostrará como JSON.

Un ejemplo de un documento en MongoDB, sería el siguiente:

```
{
  "_id" : "4da2c0e2e999fb56bf000002",
  "producto" : "manzana",
  "categoría" : "fruta",
  "stock" : "100"
}
```

Imagen 124. Documento ejemplo de un producto en MongoDB.

Como se observa en la imagen anterior, es un típico documento en formato JSON, también se puede ver de la imagen que existe un id (_id), que puede tener un formato como se desee, pero en caso de no escribir un “_id” se generará automáticamente y de forma aleatoria un “_id” con un formato especial que posee MongoDB.

Ahora para mostrar una comparación entre el modelo relacional común y un documento MongoDB, a continuación una imagen:

```
Modelo Relacional

-----
                          Tabla Categoria
-----

idcategoria:      integer PRIMARY KEY
nombrecategoria: char(30)

-----

                          Tabla Producto
-----

idproducto:       integer PRIMARY KEY
nombre producto:  char(30)
idcategoria:      integer FOREIGN KEY

-----

MongoDB

{
  "_id" : "4da2c0e2e999fb56bf000002",
  "producto" : "pescado enlatado",
  "categoría" : ["alimentos","conservas"],
}
```

Imagen 125. Comparación de un documento MongoDB frente a un modelo relacional.

Como se observa en la imagen anterior, existe un modelo relacional típico con dos tablas, una llamada “Categoria” y otra llamada “Producto” que obviamente están relacionadas con una llave foránea (idcategoria). Por otro lado existe también en la imagen un equivalente a estas tablas, pero en un sistema de MongoDB que se nota la sencillez de su uso en formato JSON, en vez de crear otro documento para relacionarlo, simplemente existe en el documento un “campo” llamado “categoría”, el cual contendrá un arreglo con todas las categorías que pertenezcan al producto.

Como se puede apreciar existe una notable diferencia y mejora en cuanto a espacio físico en MongoDB, ya que sus campos no necesitan reservar un espacio fijo para sus valores, si no que tendrán de longitud el espacio que ocupen y éste podrá ir variando según las necesidades. MongoDB no se hace problema en cuanto a relaciones, ya que en un documento podría existir un campo que contenga otro documento y dentro de éste puede desarrollarse una larga secuencia de documentos o arreglos (arrays). A continuación un ejemplo:

```
{
  "_id" : "4da2c0e2e999fb56bf000002",
  "NombreBlog" : "Programación",
  "comentarios" : [
    {"usuario" : "juan", "comentario" : "Excelente Blog"},
    {"usuario" : "pedro", "comentario" : "Blog Malo"},
    {"usuario" : "maria", "comentario" : "Regular blog", "puntuacion": "6"}
  ]
}
```

Imagen 126. Documento de MongoDB con documentos o arrays dentro.

Se puede observar en la imagen anterior que se ha creado un documento, el cual interpreta un blog donde usuarios pueden hacer comentarios sobre éste. Existe un campo “comentarios”, el cual tiene asignado un array que posee 3 documentos, donde se guardan el nombre del usuario que ha comentado y de igual manera el comentario correspondiente. Resulta importante notar que en la imagen anterior, los 2 usuarios “juan” y “pedro” han comentado pero no han puntuado al blog, más si el usuario “maría” que ha dado al blog una puntuación de “6”. Con esto se puede decir que los documentos tienen un esquema libre y entre cada uno de ellos pueden variar, en este caso varían el número de campos que poseen.

2.4 Instalación de MongoDB

2.4.1 Instalación en Windows

1) **Descargar la versión más reciente de MongoDB desde la página de descargas (<http://www.mongodb.org/downloads>).**

Hay tres tipos de MongoDB para Windows:

- MongoDB para Windows Server 2008 R2, que se ejecuta en Windows Server 2008 R2, Windows 7 64-bit, y en las nuevas versiones de Windows. Esta construcción se aprovecha de las mejoras recientes de la plataforma Windows y no puede funcionar en versiones anteriores de Windows.
- MongoDB para Windows de 64 bits, que se ejecuta en cualquier versión de 64 bits de Windows más reciente de Windows XP, Windows Server 2008 R2 y Windows 7 64-bit.
- MongoDB para Windows de 32 bits, que se ejecuta en cualquier versión de 32 bits de Windows XP y versiones de Windows más recientes. Versiones de MongoDB de 32-bit son solamente para los sistemas más antiguos y para uso de pruebas.

Se recomienda utilizar una versión más reciente de Windows, para poder utilizar versiones más recientes de MongoDB.

Hay que tener en cuenta que versiones de MongoDB de 32-bits son adecuados sólo para fines de prueba y evaluación, y sólo admiten bases de datos de menos de 2GB.

Para conocer la arquitectura de su versión de Windows se puede utilizar el comando siguiente en el símbolo del sistema (sin las comillas por supuesto):

“wmic os get osarchitecture”.

En el Explorador de Windows, buscar el archivo donde se descargó MongoDB. Extraer los archivos en “C:\” haciendo clic derecho sobre el archivo comprimido y seleccionando “Extraer todo” y navegar a “C:\”.

La carpeta extraída se verá de la siguiente manera:

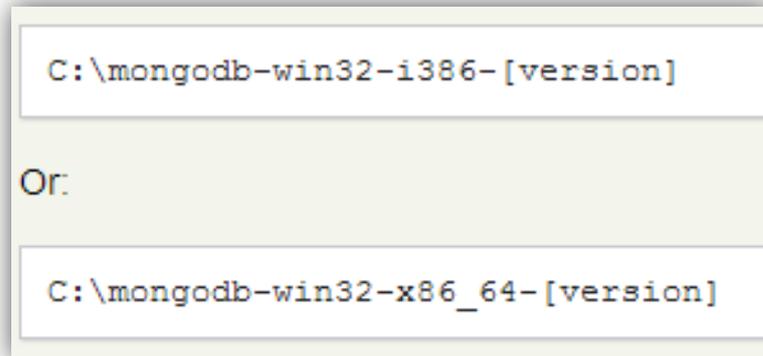


Imagen 127. Ubicación de carpeta extraída de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Donde la palabra “version” de la imagen anterior se reemplazará por la versión de la cual se ha descargado.

2) Configurar el entorno

Iniciar el símbolo del sistema, (Menú Inicio > Todos los programas > Accesorios, luego hacer clic derecho en Símbolo del sistema y seleccionar “Ejecutar como administrador”). En el símbolo del sistema, ejecutar los siguientes comandos:

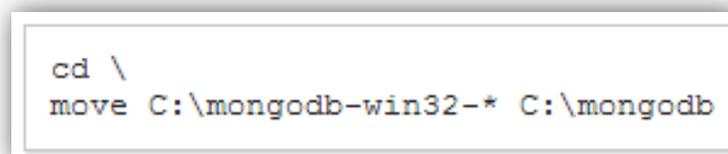


Imagen 128. Moviendo la carpeta extraída de MongoDB a otra ubicación en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Lo que se muestra en la imagen anterior simplemente es una sugerencia de donde ubicar la carpeta de MongoDB (en este caso sería en “C:\mongodb”), pero se puede colocar en cualquier lugar de las carpetas de Windows.

MongoDB requiere una carpeta de datos para almacenar sus archivos. La ubicación predeterminada para el directorio de datos MongoDB es “C:\data\db”. Por lo tanto hay que crear esta carpeta ya sea manualmente o mediante el símbolo del sistema, como se muestra a continuación la secuencia de comandos:

```
md data
md data\db
```

Imagen 129. Creando la carpeta para almacenar los archivos de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Hay como especificar una ruta alternativa a la ubicación predeterminada “C:\data\db”, con el ajuste “dbpath” de mongod.exe (es el demonio o daemon que corre el servicio de MongoDB), como a continuación se indica:

```
C:\mongodb\bin\mongod.exe --dbpath d:\test\mongodb\data
```

Imagen 130. Cambiar a otra ruta alternativa la ubicación para almacenar los archivos de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

3) Iniciar MongoDB

Para iniciar MongoDB, desde el símbolo del sistema ejecute lo siguiente:

```
C:\mongodb\bin\mongod.exe
```

Imagen 131. Iniciar MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Eso iniciará el proceso de MongoDB de la base de datos principal.

Ahora para conectar con MongoDB para poder comenzar ya a utilizar los comandos de Mongo, se necesita abrir otra ventana de símbolo del sistema y ejecutar el siguiente comando:



```
C:\mongodb\bin\mongo.exe
```

Imagen 132. Conectar con MongoDB en Windows.

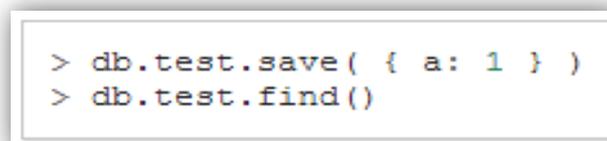
Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

El comando de la imagen anterior (mongo.exe), iniciará automáticamente la consola de mongo.exe en la ventana del símbolo del sistema.

La consola se conectará a “mongod.exe”, corriendo en Localhost en el puerto 27017 por default. Ahora ya se puede manipular las bases de datos, colecciones y documentos de MongoDB con sus respectivos comandos. Existe la posibilidad de autenticarse con un nombre de usuario (administrador) y contraseña para seguridad, para crear un usuario y autenticarse se hará lo siguiente:

```
db.addUser(“admin”,”password”);  
db.auth(“admin”,”password”);
```

En MongoDB por defecto viene una base de datos llamada “test”, a continuación un ejemplo de cómo guardar (.save()) un documento ({a : 1}) en una colección “test”, en esa base de datos por defecto (test), y después de ello se consultará (.find()) los datos :



```
> db.test.save( { a: 1 } )  
> db.test.find()
```

Imagen 133. Guardar y consultar un documento en la base de datos por defecto “test”.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

4) MongoDB como un servicio de windows

MongoDB se instala como un servicio de Windows, por lo que la base de datos se iniciará automáticamente en cada reinicio.

Configurar el sistema

Se debe especificar dos opciones al ejecutar MongoDB como un servicio de Windows: Una ruta para salida del registro (es decir, un logpath) y un archivo de configuración.

- Crear un directorio específico para los archivos de registro de MongoDB desde el símbolo del sistema:

```
md C:\mongodb\log
```

Imagen 134. Creación de un directorio para archivos de registro de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

- Crear un archivo de configuración para el “logpath” desde el símbolo del sistema:

```
echo logpath=C:\mongodb\log\mongo.log > C:\mongodb\mongod.cfg
```

Imagen 135. Creación de un archivo de configuración para el logpath en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Si bien estos pasos son opcionales, la creación de un lugar específico para los archivos de registro y el uso del archivo de configuración son recomendables.

Instalar y ejecutar el servicio de MongoDB

Hay que ejecutar los siguientes comandos en el símbolo del sistema con “privilegios de administrador”.

- Para instalar el servicio de MongoDB:

```
C: \ MongoDB \ bin \ mongod.exe - config C: \ MongoDB \ mongod.cfg - instalar
```

Imagen 136. Instalar el servicio de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

- Para ejecutar el servicio de MongoDB:

```
net MongoDB inicio
```

Imagen 137. Ejecutar el servicio de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

Detener y quitar el servicio de MongoDB

- Para detener el servicio de MongoDB:

```
net stop MongoDB
```

Imagen 138. Detener el servicio de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

- Para quitar el servicio de MongoDB:

```
C:\mongodb\bin\mongod.exe --remove
```

Imagen 139. Quitar el servicio de MongoDB en Windows.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-windows/>]

2.4.2 Instalación en Linux

1. Descargar MongoDB

Se debe colocar los archivos descargados de MongoDB en una ubicación central en el sistema de archivos que sea fácil de acceder y controlar. Considere la posibilidad en: “/opt” ó “usr/local/bin”.

En una terminal, comience la descarga de la última versión, en la mayoría de los casos se deberá descargar la versión de 64-bits de MongoDB.

```
curl http://downloads.mongodb.org/linux/mongodb-linux-x86_64-2.2.1.tgz > mongo.tgz
```

Imagen 140. Descarga de MongoDB para Linux 64-bits.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

En caso que se desee descargar la versión para 32-bits de MongoDB, se deberá utilizar el siguiente comando:

```
curl http://downloads.mongodb.org/linux/mongodb-linux-x86_64-2.2.1.tgz > mongo.tgz
```

Imagen 141. Descarga de MongoDB para Linux 32-bits.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

Una vez descargado, hay que extraer los archivos con el siguiente comando:

```
tar -zxvf mongo.tgz
```

Imagen 142. Extracción de los archivos descargados de MongoDB en Linux.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

El binario “mongod” y los demás servicios asociados a MongoDB se encuentran en el directorio “/bin”, dentro del directorio extraído.

2. Usando MongoDB

Para comenzar a utilizar “mongod” por primera vez, hay que crear el directorio para guardar los datos de MongoDB. De manera predeterminada los datos se guardan en el directorio “/data/db”. Para crear ese directorio, realizar lo siguiente:

```
mkdir -p /data/db
```

Imagen 143. Creación de directorio para guardar los datos de MongoDB en Linux.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

Hay que asegurarse de que la cuenta del sistema en que correrá el proceso “mongod” tenga permisos de lectura y escritura para el directorio. Si por ejemplo “mongod” se ejecuta bajo la cuenta de usuario “Usuario”, hay que ejecutar el siguiente comando para cambiar el propietario de la carpeta.

```
chown mongo /data/db
```

Imagen 144. Cambiar de propietario del de directorio donde se guardarán los datos de MongoDB en Linux.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

Se puede especificar y crear una ruta alternativa mediante la opción “-dbpath” de “mongod” y el comando anterior.

Para iniciar un proceso “mongod”, se lo puede hacer en la terminal, de la siguiente manera:

```
mongod --config /etc/mongod.conf
```

Imagen 145. Iniciar “mongod” de MongoDB en Linux.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

El comando que se muestra en la imagen anterior supone que el binario “mongod” ha creado un archivo de configuración por defecto, localizado en “/etc/mongod.conf”.

Entre las herramientas incluidas en esta distribución MongoDB, está la consola “mongo”. Se utiliza esa consola para conectarse a la instancia de MongoDB, de la siguiente manera:



Imagen 146. Conectarse a la instancia de MongoDB en Linux.

Fuente: Documentación del proyecto MongoDB. [Disponible en: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-linux/>]

El comando anterior “./bin/mongo” asume que el binario “mongo” se encuentra en el subdirectorio “bin/” del directorio actual. Este es el directorio en el que se extrajo el archivo .tgz.

Esto conectará a la base de datos corriendo en la interfaz localhost por defecto.

De igual manera se puede autenticar o con un nombre de usuario y contraseña, a continuación por ejemplo se muestra cómo se ha agregado un usuario (admin) con su contraseña (password) para autenticarse:

```
db.addUser(“admin”,”password”);  
db.auth(“admin”,”password”);
```

En la consola “mongo” se puede emitir los comandos de la imagen 133, para insertar o guardar un documento ({a : 1}) en la colección “test” de la base de datos predeterminada (test), para después consultar lo agregado.

2.5 PHP con MongoDB

Lo que se tratará en este apartado del capítulo, será cómo usar MongoDB específicamente sobre el manejo del controlador para PHP.

2.5.1 Instalación del controlador

2.5.1.1 Instalación en Windows

Instalar el controlador o driver de MongoDB para Php es muy sencillo, lo primero que hay que hacer es descargar un archivo o extensión .dll, dependiendo de la versión de PHP que se tenga instalado, a continuación una tabla con la lista de los archivos para descargar, con la versión de PHP respectiva:

Versión de PHP	URL para descargar
PHP 5.2 VC6 Non Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.2vc6.zip
PHP 5.2 VC6 Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.2vc6ts.zip
PHP 5.3 VC6 Non Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.3vc6.zip
PHP 5.3 VC6 Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.3vc6ts.zip
PHP 5.3 VC8 Non Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.3vc8.zip
PHP 5.3 VC8 Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.3vc8ts.zip
PHP 5.3 VC9 Non Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.3vc9.zip
PHP 5.3 VC9 Thread Safe	http://downloads.mongodb.org/mongo-latest-php5.3vc9ts.zip

Tabla 56. Lista de archivos o extensiones .dll para instalar el controlador de PHP para MongoDB.

Fuente: Web PHP. [Disponible en:
<http://www.php.net/manual/es/mongo.installation.php#mongo.installation.windows>]

Se descargarán 2 archivos, un “.txt” que contendrá información de la versión del controlador y otro archivo que será el “.dll” llamado, “php_mongo.dll”.

El archivo “.dll” hay que copiarlo en la carpeta de extensiones “ext” donde se tiene instalado PHP, y después añadir una línea en el archivo “php.ini” de ésta forma:

```
extension=php_mongo.dll
```

Con esto ya se puede comenzar a programar en php con MongoDB.

2.5.1.2 Instalación en Linux

Para instalar en Linux es igual de sencillo, simplemente descargamos la extensión de MongoDB para PHP. En Linux se lo puede descargar mediante repositorios PECL, con los siguientes comandos:

```
sudo pecl install mongo
```

Aquello hará que se descargue la extensión o módulo para Linux, llamada “mongo.so”. Lo que resta es agregar al archivo “php.ini” la siguiente línea:

```
extension=mongo.so
```

Restaurando los servicios, ya se puede ocupar el controlador, para poder usar MongoDB en PHP.

A continuación se comenzará a explicar lo que más interesa en éste capítulo, que es el código de php en sí, para manipular la base de datos MongoDB.

2.5.2 Conexión

Conectarse a MongoDB es muy fácil. La manera más sencilla es cuando se desea conectar a localhost en el puerto predeterminado “27017”, de la siguiente manera:

```
$conexion = new MongoClient();
```

Imagen 147. Conexión MongoDB simple en PHP.

Ahora si se quiere conectar a un host remoto, con el puerto por defecto “27017”, se hará de la siguiente forma:

```
$conexion = new Mongo("ejemplo.com");
```

Imagen 148. Conexión MongoDB a un host remoto en PHP.

Otro caso es cuando se desea conectar a un host remoto, pero con un puerto determinado, entonces se deberá escribir lo siguiente:

```
$conexion = new Mongo("ejemplo.com:54321");
```

Imagen 149. Conexión MongoDB a un host remoto con puerto determinado en PHP

A veces desde MongoDB se autentifica con un nombre de usuario y contraseña, pues entonces si se ha producido esa autenticación, en PHP se deberá conectar con los datos de usuario y contraseña. A continuación un ejemplo de cómo conectarse a un host remoto con puerto 54321 con autenticación:

```
$conexion = new Mongo("mongodb://admin:password@ejemplo.com:54321");
```

Imagen 150. Conexión MongoDB a un host remoto con puerto determinado y autenticación en PHP.

En el caso del ejemplo de la imagen anterior, se ha conectado a un host “ejemplo.com” con puerto “54321”, autenticándose con el usuario “admin” y contraseña “password”. Nótese que al principio se pondrá “mongodb://”.

Con la variable “\$conexion” de los ejemplos anteriores, quedará establecida la conexión y con aquella variable se podrá acceder a las base de datos, colecciones y documentos.

Algo que cabe mencionar es que no es necesario desconectar explícitamente de la base de datos, es decir con código, porque cuando “\$conexion” queda fuera de ámbito, la conexión se cierra automáticamente.

2.5.3 Seleccionar una base de datos y una colección.

Para seleccionar una base de datos es tan simple como esto:

```
$conexion = new Mongo();  
$db = $conexion -> BaseEjemplo;
```

Imagen 151. Seleccionando una base de datos en MongoDB desde PHP.

Como se observa en la imagen anterior, con la conexión ya creada (\$conexion), se selecciona una base de datos llamada “BaseEjemplo”, la cual se podrá acceder a sus colecciones mediante la variable “\$db” a la cual se le asigna la conexión (\$conexion) y se apunta a “BaseEjemplo” con este signo “->”. No es necesario que la base de datos exista, con seleccionar de esa manera se creará automáticamente.

Y ahora para seleccionar una colección de una base de datos se hará de la siguiente manera:

```
$conexion = new Mongo();  
$db = $conexion -> BaseEjemplo;  
$coleccion = $db -> ColeccionEjemplo;
```

Imagen 152. Seleccionando una colección de una base de datos en MongoDB desde PHP.

Como se puede observar ahora para seleccionar una colección “ColeccionEjemplo”, se ha creado una variable “\$coleccion” a la cual se le asigna la variable “\$db” que está apuntando a la base de datos “BaseEjemplo” y ésta apuntará y seleccionará la colección denominada “ColeccionEjemplo”. Con la variable “\$coleccion” se podrá acceder a los documentos, para realizar operaciones de escritura y consultas.

2.5.4 Operaciones de escritura

Para los siguientes ejemplos se tomará como base el ejemplo de la imagen 152, para realizar las operaciones sobre la base de datos “BaseEjemplo” y la colección “ColeccionEjemplo”.

2.5.4.1 Insertar documentos

Para insertar documentos en una colección, es tan simple como esto:

```
$documento = array("nombre" => "juan", "apellido" => "perez");  
$coleccion -> insert($documento);
```

Imagen 153. Insertar un documento en una base de datos en MongoDB desde PHP.

En el ejemplo de la imagen anterior se puede observar de qué manera se ha creado un documento, que es igual a crear un array y se le asigna a una variable (\$documento), para después mediante la propiedad “insert()”, agregar a la colección de la base de datos.

En la base de datos MongoDB se guardará el documento de la siguiente forma:

```
{  
  "_id"      : ObjectId("50a85a5200565bb814000000"),  
  "nombre"   : "juan",  
  "apellido" : "perez"  
}
```

Imagen 154. Documento insertado en MongoDB.

El documento que se muestra en la imagen anterior, es el que se ha insertado y de esa manera se observará en la base de datos de MongoDB, se puede apreciar que se ha generado un ID automáticamente, que es un Objeto Id de MongoDB, que siempre se generará automáticamente a no ser que se especifique un “_id” específicamente.

A continuación se puede apreciar más inserciones de diferentes documentos, que como se mencionó anteriormente se los puede comparar como un array de PHP.

Ahora se insertará un documento un poco más complejo, en cuanto a la estructura, con una ID específica:

```

$documento = array("_id" => "123", "nombre" => "pedro",
                  "apellido" => "lopez",
                  "hijos" => array("luis", "maria", "mateo"));

$coleccion -> insert($documento);

```

Imagen 155. Insertar un documento con ID específica, en una base de datos en MongoDB desde PHP.

Como se puede observar en la imagen anterior, ahora se ha especificado una “_id” (123), por lo tanto MongoDB ya no generará un Objeto ID automáticamente. También se puede notar de la imagen ejemplo, que existe un campo o llave denominada “hijos”, la cual contiene un arreglo, obviamente con el nombre de los hijos de esa persona (pedro). En la siguiente imagen se muestra, cómo se visualizará el documento en la base de datos:

```

{
  "_id"      : "123",
  "nombre"   : "pedro",
  "apellido" : "lopez",
  "hijos"    : ["luis", "maria", "mateo"]
}

```

Imagen 156. Documento insertado en MongoDB con una ID específico.

También se puede ir anidando dentro de cada array más y más arrays para insertar en la base de datos, por ejemplo:

```

$documento = array("nombre" => "manuel",
                  "apellido" => "lopez",
                  "edad" => 62,
                  "hijos" =>
                    array(array("nombre" => "maría", "edad" => 15),
                          array("nombre" => "gerardo"),
                          array("nombre" => "rocío", "edad" => 21)));

$coleccion -> insert($documento);

```

Imagen 157. Insertar un documento en una base de datos en MongoDB con documentos o arrays anidados desde PHP.

En la base de datos se mostrará el registro agregado de la siguiente manera o estructura:

```
{
  "_id"      : ObjectId("50a86a3a00565b8c10000000"),
  "nombre"   : "manuel",
  "apellido" : "lopez",
  "edad"     : 62,
  "hijos"    : [
    {
      "nombre" : "mar tija",
      "edad"   : 15
    },
    {
      "nombre" : "gerardo"
    },
    {
      "nombre" : "roc tjo",
      "edad"   : 21
    }
  ]
}
```

Imagen 158. Documento insertado en MongoDB con documentos anidados.

Como se observa en la imagen anterior, ahora se ha insertado un documento en el cual se tiene un campo o llave denominado “hijos”, éste campo contiene un arreglo (array) con más documentos dentro. Si se desearía poner más arrays dentro pues no existiría ningún inconveniente. Es importante notar que si el valor de un campo (llave) tiene el carácter de tilde, entonces MongoDB guardará esa tilde como un carácter o caracteres especiales no entendibles, pero que al momento de consultar esos datos se mostrarán con la tilde normalmente.

Algo que se puede notar en todos los ejemplos anteriormente explicados de inserción de documentos, es que se están agregando a una colección (ColeccionEjemplo), y tienen cada uno de ellos una estructura diferente.

2.5.4.2 Actualizar documentos

Las actualizaciones en MongoDB combinan una consulta (que se verá detalladamente más adelante) y una acción (modificación), modificando documentos que cumplan con un criterio establecido.

Existen dos tipos de actualizaciones: De modificación y de reemplazo.

2.5.4.2.1 Actualizaciones de modificación:

Este tipo de actualizaciones contienen un operador “\$”, y cambian los campos de los documentos, también pueden incrementar contadores, agregar y eliminar elementos a un array o simplemente modificar el dato (valor) de un campo (llave). Lo nombrado anteriormente se puede llevar a cabo gracias a los operadores de modificación siguientes: \$inc, \$set, \$unset, \$push, \$pull.

\$inc

Sirve para aumentar o disminuir un valor numérico.

Por ejemplo, en base al documento de la imagen 158, en el cual existe un campo “edad” con el valor de 62, si se desea aumentar la edad a 63 se hará lo siguiente:

```
$coleccion -> update(array("nombre" => "manuel",  
                           "apellido" => "lopez"),  
                    array('$inc' => array("edad" => 1)));
```

Imagen 159. Actualizar un documento con el operador \$inc en MongoDB.

En el ejemplo de la imagen anterior se está utilizando “update” para actualizar el documento, en la primera parte agregamos un array y le indicamos la condición para actualizar el documento, en el ejemplo anterior la condición sería: Actualizar un documento que tenga el campo “nombre” = “manuel” y el campo “apellido” = “lopez”. En la segunda parte igualmente agregamos un array ahora con el operador \$inc (tener en cuenta que el operador va entre comillas simples) y después el array con el campo (edad) que incrementará, y el valor con que incrementará (en ese caso el 1), eso hará que cambie el campo edad a 63. Si lo que se desea es disminuir el valor se pondrá un valor negativo.

También se puede modificar el mismo documento de la imagen 158, consultando por el Objeto ID de MongoDB. Por ejemplo:

```
$coleccion -> update(array("_id" => new ObjectId("50a86a3a00565b8c10000000")),  
                    array('$inc' => array("edad" => 1)));
```

Imagen 160. Actualizar un documento con el operador \$inc, consultando por el Objeto ID de MongoDB.

Ahora la condición será: Actualizar el documento que tenga un “_id” = Un Objeto ID de MongoDB (50a86a3a00565b8c10000000).

Pero ahora si lo que se quiere es actualizar la edad de algún campo que esté contenido en un array (documentos anidados). Por ejemplo y en base nuevamente en la imagen 158, si se desea disminuir 2 años la edad de un hijo de “manuel lopez”, por ejemplo de “rocío”, se hará de la siguiente manera:

```
$coleccion -> update(array("nombre" => "manuel",  
                          "apellido" => "lopez"),  
                    array('$inc' => array("hijos.2.edad" => -2)));
```

Imagen 161. Actualizar un documento con el operador \$inc, en documentos anidados de MongoDB.

Con la expresión “hijos.2.edad” se señala al hijo “rocío” de “manuel lopez”. Se debe tener presente que para llegar al campo de un documento anidado, se debe hacer de una manera jerárquica.

A continuación se muestra una imagen que muestra el por qué la expresión “hijos.2.edad” señala a “rocío”:

```
{
  "_id"      : ObjectId("50a86a3a00565b8c10000000"),
  "nombre"   : "manuel",
  "apellido" : "lopez",
  "edad"     : 62,
  "hijos"    : [
    {
      "nombre" : "maría",
      "edad"   : 15
    },
    {
      "nombre" : "gerardo"
    },
    {
      "nombre" : "rocío",
      "edad"   : 21
    }
  ]
}
```

Imagen 162. Explicación de un documento anidado de MongoDB.

Como se puede observar en la imagen anterior el campo “hijos” contiene todo lo que está entre dentro del recuadro de color amarillo (array), y a su vez ese recuadro contiene 3 recuadros verdes (documentos) donde cada uno de ellos tienen el index de posición respectivo. Entonces si se quiere señalar la edad de “rocío” para modificarla, se debe acceder al campo “hijos”, en su documento con index “2” y en su campo “edad” que resultaría la expresión: “hijos.2.edad”.

Cabe mencionar que dentro de un campo pueden existir muchos documentos y será difícil conocer su index de posición, por lo tanto se puede reemplazar ese número de index por un signo “\$”. A continuación se muestra cómo modificar la edad ahora del hijo “maría” de “manuel lopez”, sin conocer su posición de index:

```
$coleccion -> update(array("nombre" => "manuel",
                           "apellido" => "lopez",
                           "hijos.nombre" => "rocío"),
                    array('$inc' => array("hijos.$.edad" => 1)));
```

Imagen 163. Actualizar un documento anidado sin conocer la posición index de MongoDB.

Como se observa en la imagen anterior, ahora a más de las condiciones de “nombre” = “manuel” y “apellido” = “lopez”, aparece otra condición que es “hijos.nombre” =

“rocío”, en la que indicamos que señale dentro del campo “hijos”, el campo “nombre” que sea igual a rocío, para después con “hijos.\$edad” modificar el documento.

\$set

Sirve para cambiar el valor de un campo (llave), si el campo no existiera pues entonces se creará.

Por ejemplo, si se tuviera este documento:

```
{
  "_id"      : ObjectId("50a987fb00565b4010000000"),
  "nombres"  : "juan perez",
  "direccion": "Av. Americas",
  "telefono" : "09888810"
}
```

Imagen 164. Documento ejemplo para actualizarlo con \$set en MongoDB.

Y ahora se quisiera modificar o actualizar el campo “telefono” pues es tan simple como esto:

```
$coleccion -> update(array("nombres" => "juan perez"),
                  array('$set' => array("telefono" => "09999910")));
```

Imagen 165. Actualizar documento con el operador \$set en MongoDB.

Eso actualizará el campo “telefono” de “09888810” a “09999910”. También se puede actualizar documentos anidados.

Como se manifestó antes, si es que se quiere modificar un campo que no existiera, el operador de modificación “\$set” agregará ese campo. Por ejemplo si se tuviera un documento de esta manera:

```

{
  "_id" : ObjectId("50a9978b00565b4010000001"),
  "nombres" : "henry campoverde",
  "telefono" : "0998765432",
  "materias" : [
    { "nombre" : "matematicas" },
    { "nombre" : "geografia" },
    { "nombre" : "dibujo" }
  ]
}

```

Imagen 166. Documento ejemplo para agregar un campo con \$set en MongoDB.

Como se observa en el documento de la imagen anterior, se ha logrado hacer como un listado de materias de un estudiante, en ese caso de un estudiante llamado “henry campoverde”, con el teléfono “0998765432” y con las materias de “matematicas”, “geografia” y “dibujo”. Ahora si por ejemplo se quiere añadir una calificación de “9” a la materia “geografia”, se hará de la siguiente manera:

```

$coleccion -> update(array("nombres" => "henry campoverde",
                           "materias.nombre" => "geografia"),
                   array('$set' => array("materias.$.calificacion" => 9)));

```

Imagen 167. Actualizar un documento agregando un nuevo campo con \$set en MongoDB.

Ahora el documento quedará de la siguiente forma:

```

{
  "_id" : ObjectId("50a9978b00565b4010000001"),
  "nombres" : "henry campoverde",
  "telefono" : "0998765432",
  "materias" : [
    { "nombre" : "matematicas" },
    { "nombre" : "geografia", "calificacion" : 9 },
    { "nombre" : "dibujo" }
  ]
}

```

Imagen 168. Documento después de agregarse un campo nuevo con \$set en MongoDB.

\$unset

Sirve para eliminar un campo (llave) de un documento, por ejemplo si se tuviera el siguiente documento:

```
{
  "_id"      : ObjectId("50a9abd100565ba406000003"),
  "producto" : "camiseta",
  "descripcion1" : "camiseta small roja",
  "descripcion2" : "exclusiva para hombres"
}
```

Imagen 169. Documento para borrar un campo con \$unset en MongoDB.

Y ahora si se quisiera eliminar el campo “descripcion2”, se haría de la siguiente manera:

```
$coleccion -> update(array("producto" => "camiseta"),
                    array('$unset' => array("descripcion2" => 1)));
```

Imagen 170. Ejemplo para Borrar un campo con \$unset en MongoDB.

Con esa sintaxis (descripcion2 => 1) se borrará el campo “descripcion2”.

Ahora el documento quedará de la siguiente manera:

```
{
  "_id"      : ObjectId("50a9abd100565ba406000003"),
  "producto" : "camiseta",
  "descripcion1" : "camiseta small roja"
}
```

Imagen 171. Documento después de haber borrado un campo con \$unset en MongoDB.

\$push

Sirve para añadir un elemento al final de un array. Por ejemplo en base al siguiente documento:

```
{
  "_id"      : "usuario12345",
  "nombres"  : "mauricio rodas",
  "telefonos" : [ "2876524", "2871717" ]
}
```

Imagen 172. Documento ejemplo para añadir un elemento a un array con \$push en MongoDB.

Como se puede observar en el documento ejemplo de la imagen anterior, existe un campo “telefonos”, el cual contiene un array con 2 números de teléfono, si se quisiera añadir otro número de teléfono en ese array, por ejemplo el número “2888990”, se hará de la siguiente manera:

```
$coleccion -> update(array("_id" => "usuario12345"),
                    array('$push' => array("telefonos" => "2888990")));
```

Imagen 173. Actualizar un documento con \$push en MongoDB.

El documento ahora quedará de la siguiente forma:

```
{
  "_id"      : "usuario12345",
  "nombres"  : "mauricio rodas",
  "telefonos" : [ "2876524", "2871717", "2888990" ]
}
```

Imagen 174. Resultado de un documento actualizado con \$push en MongoDB.

Se puede observar que se ha agregado el nuevo elemento al array “288990”. En caso de que el elemento ya existiera en el array, simplemente el elemento se repetirá, y si lo que se desea es no repetir un elemento que ya exista en el array, se deberá utilizar en vez del operador “\$push” el operador “\$ne”, el cual agregará un elemento al array siempre y cuando no exista ya dentro del array.

\$pull

Es el opuesto del operador “\$push”, ya que “\$pull” sirve para eliminar un elemento de un array. Por ejemplo, y en base al documento de la imagen 174, si ahora se quisiera eliminar el teléfono “2871717”, entonces se codificaría de la siguiente manera:

```
$coleccion -> update(array("_id" => "usuario12345"),
                    array('$pull' => array("telefonos" => "2871717")));
```

Imagen 175. Actualizar un documento con \$pull en MongoDB.

Con eso se ha eliminado un elemento del array del campo “telefonos”, el resultado de esa actualización quedará de la siguiente manera:

```
{
  "_id"      : "usuario12345",
  "nombres"  : "mauricio rodas",
  "telefonos" : [ "2876524", "2888990" ]
}
```

Imagen 176. Resultado de actualizar un documento con \$pull en MongoDB.

Cabe mencionar que todos los operadores de modificación se puede utilizar para modificar o actualizar documentos anidados, solo es cuestión de aplicar los conceptos de jerarquía que se explicó anteriormente. También se pueden realizar condiciones o consultas más complejas, como por ejemplo: -Actualizar un documento donde el campo sea mayor o igual a un número determinado-. Consultas que más adelante se explicarán con ejemplos simples y claros.

2.5.4.2.2 Actualizaciones de reemplazo:

Estas actualizaciones no utilizan los operadores de modificación y sirven para cambiar la estructura completamente de un documento señalado. Por ejemplo, teniendo el siguiente documento:

```
{
  "_id"      : ObjectId("50aa96c800565b4010000002"),
  "nombres"  : "marco jaramillo",
  "direccion" : "Cdla. Totoracocha",
  "telefono" : "098765432"
}
```

Imagen 177. Documento para actualizar mediante actualizaciones de reemplazo en MongoDB.

Si se quisiera cambiar la estructura del documento de la imagen anterior, se podría hacer de la siguiente manera:

```
$coleccion -> update(array("nombres" => "marco jaramillo"),
                    array("nombre" => "marco",
                          "apellido" => "jaramillo",
                          "telefonos" => array("2876754", "097864235")));
```

Imagen 178. Actualizar un documento mediante actualizaciones de reemplazo en MongoDB.

Con la actualización que se puede observar en la imagen anterior ya no se utiliza operadores de modificación (\$inc, \$set, \$unset, \$push, \$pull), el documento cambia su estructura, y quedaría de la siguiente forma:

```
{
  "_id"      : ObjectId("50aa96c800565b4010000002"),
  "nombre"   : "marco",
  "apellido" : "jaramillo",
  "telefonos" : [ "2876754", "097864235" ]
}
```

Imagen 179. Documento actualizado mediante actualizaciones de reemplazo en MongoDB.

2.5.4.3 Eliminar documentos

El eliminar documentos es muy sencillo, ahora se utiliza el método “remove()”, acompañada de una condición o consulta para obtener el documento que se quiera borrar, por ejemplo si se quisiera borrar el documento que está en la imagen anterior, se haría de la siguiente manera:

```
$coleccion -> remove(array("nombre" => "marco",
                          "apellido" => "jaramillo"));
```

Imagen 180. Eliminar un documento en MongoDB.

Con la sintaxis de la imagen anterior se eliminará el documento que tenga por nombre “marco” y apellido “jaramillo”.

2.5.4.4 Eliminar colección

La eliminación de una colección es tan simple como esto:

```
$eliminar = $coleccion -> drop();
```

Imagen 181. Eliminar una colección en MongoDB.

Lo que se hace para eliminar una colección crear una variable, en este caso se ha creado una variable llamada “\$eliminar”, a la cual se le asigna la colección que se quiere eliminar y ésta apuntará a un método llamado drop().

2.5.4.5 Eliminar base de datos

Eliminar una base de datos es igual que eliminar una colección, con la diferencia que obviamente se tendrá que asignar a una variable, el nombre de la base de datos a eliminar, como se muestra a continuación:

```
$eliminar = $db -> drop();
```

Imagen 182. Eliminar una base de datos en MongoDB.

En el caso de la imagen 181 y 182 las variables “\$coleccion” y \$db respectivamente deberían antes haberse asignado el nombre y la base a la cual pertenece, como se puede observar en el ejemplo de la imagen 152 anteriormente explicada.

2.5.5 Consultas

Las consultas se hacen para obtener documentos y así poder visualizar sus datos. En las actualizaciones explicadas anteriormente se ponía una condición para encontrar los documentos que se querían modificar, pues esas condiciones se puede incluir en las consultas, ya que poniendo esa condición se obtendrán los documentos que se requieren. Se utilizará como base los documentos de la imagen que a continuación se muestra para realizar consultas y mostrarlas en pantalla:

```

{
  "_id"      : ObjectId("50aac7f100565b4010000003"),
  "nombres"  : "marco jaramillo",
  "direccion": "Cdla. Totoracocha",
  "telefonos": [ "098765432", "099999123" ],
  "edad"     : 56,
  "hijos"    : [{ "nombres" : "luis jaramillo", "edad" : 20 },
                { "nombres" : "maria jaramillo", "edad" : 17 }],
  "email"    : "marcojaramillo@gmail.com"
}
{
  "_id"      : ObjectId("50aac95b00565ba406000004"),
  "nombres"  : "juan perez",
  "direccion": "Luis Cordero 5-45",
  "telefonos": [ "091897666", "2867346" ],
  "edad"     : 30,
  "hijos"    : [{ "nombres" : "luisa perez", "edad" : 10 }],
  "email"    : "juanperez@hotmail.com",
  "estado civil" : "casado"
}
{
  "_id"      : ObjectId("50aac9f200565b4010000004"),
  "nombres"  : "diana perez",
  "direccion": "Cdla. Orquideas",
  "telefonos": [ "2850365" ],
  "edad"     : 19,
  "email"    : "dianaperez@yahoo.com",
  "estado civil" : "soltero"
}
{
  "_id"      : ObjectId("50aaca9900565ba406000005"),
  "nombres"  : "rebeca montero",
  "direccion": "Miraflores",
  "telefonos": [ "091899999", "2886060" ],
  "edad"     : 71,
  "hijos"    : [{ "nombres" : "marco montero", "edad" : 32 },
                { "nombres" : "luis montero", "edad" : 43 },
                { "nombres" : "sofia montero", "edad" : 40 }],
  "email"    : "rebecamontero@gmail.com"
}

```

Imagen 183. Documentos para realizar ejemplos de consultas en MongoDB.

Todos los documentos que se observan en la imagen anterior, pertenecen a la colección llamada “ColeccionEjemplo” y a la base de datos “BaseEjemplo”.

Por ejemplo si se desea consultar los datos de la persona “diana perez”, se lo hará de la siguiente manera:

```

$consulta = $coleccion -> findOne(array("nombres" => "diana perez"));

```

Imagen 184. Consultar un solo documento en MongoDB.

Como se puede observar en la imagen anterior, se utiliza para realizar la consulta el método `findOne()`, el cual sirve para consultar 1 solo documento en una colección, por tanto lo que contendrá la variable “\$consulta” (variable que recibe los resultados de consulta en forma de array), será solo los datos de “diana perez”. Ahora ya con los datos de consulta obtenidos se los puede mostrar en pantalla de la siguiente manera:

```
$nombres = $consulta["nombres"];
$direccion = $consulta["direccion"];
$telefonos = $consulta["telefonos"];
$edad = $consulta["edad"];
$email = $consulta["email"];
$estadocivil = $consulta["estado civil"];

echo $nombres."<br>";
echo $direccion."<br>";
echo $telefonos[0]."<br>";
echo $edad."<br>";
echo $email."<br>";
echo $estadocivil."<br>";
```

Imagen 185. Mostrar datos resultantes de una consulta con `findOne()` en MongoDB.

Como se puede observar en la imagen anterior, de la variable que contendrá el resultado de la consulta (`$consulta`), se obtiene los campos del documento consultado, simplemente poniendo entre corchetes el nombre del campo y asignándole a una variable para poder mostrar en pantalla, en ese caso se muestra los datos con un “echo”. Cabe recalcar que cuando se obtiene el campo “telefonos” que contiene un array, se muestra entre corchetes el número 0, que es la posición del teléfono en el array.

Ahora si lo que se desea es consultar por ejemplo los nombres de todos los usuarios almacenados en los documentos de la colección, bastará con hacer esto:

```
$consulta = $coleccion -> find();
```

Imagen 186. Consultar datos de todos los documentos de una colección en MongoDB.

Se puede observar que ahora se utiliza el método “`find()`”, el cual recibe 2 parámetros, el primero es una condición (array) para consultar los documentos y el segundo parámetro es los campos que se desean sacar (si no se pone nada se sacaran

todos los campos). El método find() devolverá todos los documentos de la colección, entonces para poder mostrar el nombre de los usuarios en pantalla se hará de la siguiente forma:

```
foreach ($consulta as $valores) {  
    $nombres = $valores["nombres"];  
    echo $nombres."<br>";  
}
```

Imagen 187. Mostrar datos resultantes de una consulta con find() en MongoDB.

Ahora como ya no se está consultando con findOne(), se deberá obtener los datos con un foreach() ya que existen varios documentos como resultado de la consulta. Dentro de la sintaxis del foreach() se pondrá la variable a la que se asignó la consulta (\$consulta) y en cada iteración del bucle se obtendrá cada documento que se asignará a la variable \$valores, dentro del foreach() se podrá obtener cada campo de cada uno de los documentos, poniendo entre corchetes el nombre del campo a obtener, en el caso de la imagen 187 se está obteniendo el campo “nombres” de los documentos resultantes de la consulta y se muestra en pantalla con “echo”.

Como se observó en los ejemplos anteriores de las actualizaciones, se incluía condiciones para encontrar el documento a modificar, pues en las consultas también se puede poner condiciones para encontrar los documentos y así poder listarlos o mostrarlos en pantalla.

Por ejemplo, y basándonos nuevamente en los documentos de la imagen 183, si se desea por ejemplo obtener o consultar los datos de documentos que coincidan con el campo “nombres” igual a “rebeca montero”, se hará de la siguiente manera:

```

$consulta = $coleccion -> find(array("nombres" => "rebeca montero"));

foreach ($consulta as $valores) {
    $nombres = $valores["nombres"];
    echo $nombres."<br>";
    $direccion = $valores["direccion"];
    echo $direccion."<br>";
    $telefonos = $valores["telefonos"];
    for ($i=0; $i <= count($telefonos) ; $i++) {
        echo $telefonos[$i]."<br>";
    }
    $edad = $valores["edad"]."<br>";
    $hijos = $valores["hijos"];
    foreach ($hijos as $hijo) {
        $nombrehijo = $hijo["nombres"];
        echo $nombrehijo."<br>";
        $edadhijo = $hijo["edad"];
        echo $edadhijo."<br>";
    }
    $email = $valores["email"];
    echo $email."<br>";
}

```

Imagen 188. Mostrar datos resultantes de una consulta con find() con una condición en MongoDB.

Como se observa ahora se tiene dentro del método find(), la condición de que busque documentos que tengan por campo (nombres) el valor de “rebeca montero”. Dentro del foreach() se obtendrán los datos y campos de los documentos que cumplan con esa condición.

Una observación que se puede hacer sobre la imagen anterior, es que cuando existen campos que contienen documentos (documentos anidados), se puede realizar los foreach() necesarios, uno dentro de otro, como es el caso de la imagen anterior (188), donde se ocupan 2 foreach() uno para sacar los documentos que cumplan la condición y otro foreach() para sacar los documentos están contenidos en el campo “hijos”. Otra cosa que hay que notar de la imagen, es que para obtener el campo “telefonos”, el cual contiene elementos en un array, se lo hace con un bucle “for”. Se debe tener presente que hay que conocer el lenguaje PHP y sus funciones, para poder mezclarlos con las funciones o métodos de MongoDB.

Como se puede notar en las consultas realizadas hasta ahora, se han sacado todos los datos de los documentos cuando se hace una consulta, obviamente se puede mostrar en pantalla solo los que se necesite, pero se tiene guardado en memoria todos los datos de los documentos consultados y sería más factible sacar solo los campos que se necesiten. Por tal motivo se puede sacar solo campos determinados, por ejemplo si se desea sacar solo el campo “edad” del documento que tenga la llave o campo “nombres” igual a “marco jaramillo”, se haría de la siguiente manera:

```
$consulta = $coleccion -> find(array("nombres" => "marco jaramillo"),
                                array("edad" => 1));

foreach ($consulta as $valores) {
    $edad = $valores["edad"];
    echo $edad."<br>";
}
```

Imagen 189. Mostrar datos resultantes de una consulta con find(), sacando un campo determinado en MongoDB.

Se puede observar que ahora el método find(), tiene dos parámetros, el primero es la condición de consulta que ya se ha explicado (array(“nombres”) => “marco jaramillo”), y el segundo parámetro (array(“edad”) => 1) es el campo que se desea sacar. Se puede sacar los campos que se desee, simplemente hay que separarlos por una coma. Ejemplo:

```
$consulta = $coleccion -> find(array("nombres" => "marco jaramillo"),
                                array("edad" => 1, "nombres" => 1));
```

Imagen 190. Consulta de datos con find(), sacando varios campos determinados en MongoDB.

Existen varias condiciones que se pueden incluir en las consultas de MongoDB, a continuación se muestra una tabla de correlación entre las consultas de MongoDB y las consultas del lenguaje de sentencias SQL:

Correlación de consultas entre MongoDB y SQL en PHP

Sentencias con SQL	Sentencias con MongoDB
SELECT a,b FROM Users	\$ColeccionUsers -> find(array(), array("a" => 1, "b" => 1));
SELECT * FROM Users WHERE age=33	\$ColeccionUsers -> find(array("age" => 33));
SELECT a,b FROM Users WHERE age=33	\$ColeccionUsers -> find(array("age" => 33), array("a" => 1, "b" => 1));
SELECT a,b FROM Users WHERE age=33 ORDER BY name	\$ColeccionUsers -> find(array("age" => 33), array("a" => 1, "b" => 1)) -> sort(array("name" => 1));
SELECT * FROM Users WHERE age>33	\$ColeccionUsers -> find(array("age" => array('\$gt' => 33)));
SELECT * FROM Users WHERE age<33	\$ColeccionUsers -> find(array("age" => array('\$lt' => 33)));
SELECT * FROM Users WHERE name LIKE "%Joe%"	\$ColeccionUsers -> find(array("name" => new MongoRegex("/Joe/")));
SELECT * FROM Users WHERE name LIKE "Joe%"	\$ColeccionUsers -> find(array("name" => new MongoRegex("/^Joe/")));
SELECT * FROM Users WHERE age>33 AND age<=40	\$ColeccionUsers -> find(array("age" => array('\$gt' => 33, '\$lte' => 40)));
SELECT * FROM Users ORDER BY name DESC	\$ColeccionUsers -> find() -> sort(array("name" => -1));
SELECT * FROM Users WHERE a=1 or b=2	\$ColeccionUsers -> find(array('\$or' => array(array("a" => 1), array("b" => 2))));
SELECT * FROM Users LIMIT 10 SKIP 20	\$ColeccionUsers -> find() -> limit(10) -> skip(20);
SELECT DISTINCT last_name FROM Users	\$db -> command(array("distinct" => "Users", "key" => "last_name"));
SELECT COUNT(*y) FROM Users	\$ColeccionUsers -> count();
SELECT COUNT(*y) FROM Users where AGE > 30	ColeccionUsers -> find(array("age" => array('\$gt' => 30))) -> count();
SELECT COUNT(AGE) from Users	ColeccionUsers -> find(array("age" => array('\$exists' => true))) -> count();

Tabla 57. Correlación entre sentencias de consulta SQL y MongoDB.

Fuente: Web PHP. [Disponible en: <http://www.php.net/manual/es/mongo.sqltomongo.php>]

Todas estas sentencias de consulta se pueden realizar en PHP, solo es cuestión de ir jugando con cada una de ellas y claro el lenguaje PHP y lograr obtener los datos o documentos que se deseen.

Como se pudo observar en la tabla 58, en las consultas también existen operadores, como también lo hay en las actualizaciones, a continuación se muestran los operadores para describir criterios de búsqueda:

\$gt = Mayor que (>).

\$lt = Menor que (<).

\$gte = Menor o igual que (\leq).

\$lte = Mayor o igual que (\geq).

\$ne = Diferente de (\neq).

\$or = Operador lógico “O” (or).

2.5.6 Manejo de Archivos

MongoDB utiliza una especificación para el almacenamiento y extracción de archivos desde la base de datos. Esta especificación define 2 colecciones: “files”, para los metadatos del archivo, y “chunks” para el contenido del fichero.

Cada documento de la colección de archivos contiene el nombre del archivo, fecha de subida, y un código. También contiene un campo único “_id”, que se puede utilizar para consultar el contenido del archivo en la colección de bloques. Cada documento de la colección de bloques contiene un bloque de datos binarios, un campo “files_id” que corresponde con el “_id” del archivo.

Para almacenar cualquier archivo en MongoDB, teniendo el nombre del archivo, se haría de la siguiente manera:

```
$nombreArchivo = $_FILES['archivo']['name'];  
  
$conexion = new Mongo();  
$db = $conexion -> BaseEjemplo;  
  
$grid = $db -> getGridFS();  
$id = $grid->storeUpload('archivo',$nombreArchivo);
```

Imagen 191. Almacenando un archivo en MongoDB.

Como se observa en la imagen anterior, se obtiene el nombre del archivo que se está queriendo almacenar en la base de datos (en este caso, por ejemplo, se supone que se obtiene el nombre de un archivo de un input tipo “file” llamado “archivo”) y se almacena en una variable (\$nombreArchivo), ya con la conexión a la base de datos realizada, se obtiene las colecciones para almacenar el archivo mediante el método “getGridFS()” que se almacenaran en la variable “\$grid”. Ahora solo queda aplicar un método llamado “storeUpload()”, el cual recibe 2 parámetros: el primero es el nombre del input tipo file, y el segundo es el nombre del archivo, esto hará que se almacene el archivo y se asigne en una variable (\$id) un ObjectID generado automáticamente por MongoDB, sobre la cual se podrá consultar sus datos. Por ejemplo para consultar el archivo mediante el ID generado se hará de la siguiente manera:

```
$id = $grid->storeUpload('archivo',$nombreArchivo);  
$imagen = $grid->findOne(array("_id" => new MongoClient($id)));
```

Imagen 192. Consultando un archivo por el ID en MongoDB.

Como se observa, ya con el “\$id” obtenido anteriormente, se realiza una simple consulta en la colección “\$grid”, mediante el ID objeto de MongoDB (new MongoClient(\$id)).

Para realizar un ejemplo práctico, a continuación se almacenará en la base de datos “BaseEjemplo” una foto, para después consultarla mediante el ID, y mostrar esa foto en pantalla.

Primeramente se creará un documento HTML5 con un input tipo “file” para escoger una foto, y también se creará un botón tipo “submit” para mandar el nombre del input (file) a un archivo de PHP, que recogerá el nombre de la foto y la almacenará en la base de datos, para después mostrarla en pantalla.

A continuación se muestra el código para crear el documento HTML5:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Almacenar y mostrar foto en MONGODB</title>
  </head>
  <body>
    <header>MongoDB Fotos</header>
    <section>
      <article>
        <form action="subirFoto.php" method="POST" enctype="multipart/form-data">
          Seleccione Foto:<br>
          <input type="file" name="imagen"><br>
          <input type="submit" value="Subir y mostrar">
        </form>
      </article>
    </section>
  </body>
</html>
```

Imagen 193. Documento HTML5 para escoger una foto y almacenarla en MongoDB.

Como se observa en la imagen anterior al momento de escoger una foto y dar click en el botón tipo “submit” (Subir y mostrar), se pasará al archivo “subirFoto.php” el input tipo “file” que por nombre lleva “imagen”. Por ejemplo si se escoge una foto llamada “imagen.jpg”:

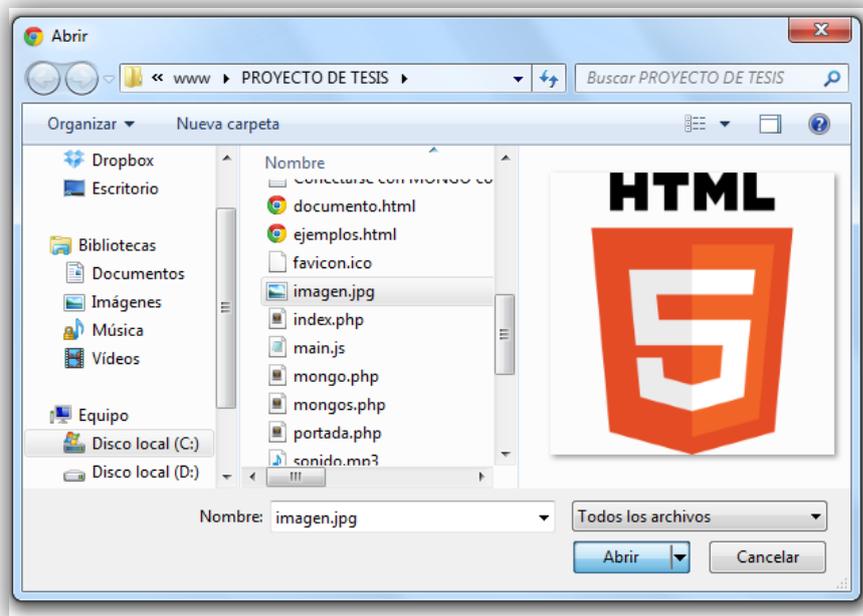


Imagen 194. Seleccionar una foto para almacenarla en MongoDB.

Como se mencionó anteriormente, al dar click sobre el botón “submit” se pasará el nombre del input “file”, ahora solo queda obtener el input “file” en el archivo PHP

(subirFoto.php), almacenar en la base de datos “BaseEjemplo” y mostrar en pantalla, como se muestra a continuación:

```
<?PHP

$nombreFoto = $_FILES['imagen']['name'];

$conexion = new MongoClient();
$db = $conexion -> BaseEjemplo;
$grid = $db -> getGridFS();
$id = $grid->storeUpload('imagen',$nombreFoto);
$imagen = $grid->findOne(array("_id" => new MongoClient($id)));
header('Content-type: image/jpeg');
echo $imagen->getBytes();

?>
```

Imagen 195. Almacenar y mostrar en pantalla una foto en MongoDB.

En la pantalla del navegador, se mostrará la imagen almacenada:



Imagen 196. Foto almacenada en la base de datos y mostrada en pantalla en MongoDB.

2.5.7 Seguridad

Para protegerse de ataques maliciosos a la base de datos, es necesario tomar en cuenta ataques de inyección de código que se pueden ejecutar:

2.5.7.1 Ataques de inyección de código

Cuando se pasa parámetros con “\$_GET” para consultar en la base de datos, hay que asegurarse de que esos parámetros se han convertido en “strings”, ya que por ejemplo un usuario (malicioso) puede insertar un array en una petición GET, provocando así consultas no deseadas.

Por ejemplo, si se buscara información o datos de un usuario con una petición “http://www.ejemplo.com?nombreusuario=juan”. En el archivo PHP se llevará a cabo la consulta:

```
$coleccion -> find(array("nombreusuario" => $_GET['nombreusuario']));
```

Entonces alguien podría alterar aquello, realizando una petición: “http://www.ejemplo.com?nombreusuario[\$ne]=foo”, con lo que PHP convertirá automáticamente los parámetros a un array, creando una consulta así:

```
$coleccion -> find(array("nombreusuario" => array('$ne' => "foo")));
```

Aquella consulta devolverá todos los usuarios con nombre distinto de “foo”, que probablemente serán todos. Es sencillo defenderse de un ataque como el antes explicado, pues simplemente hay que asegurarse de que los parámetros “\$_GET”, sean del tipo esperado antes de enviarlos a la base de datos, en este caso hay que asegurarse de convertirlos a “string”. Para convertir a “string” simplemente se puede colocar “(string)” antes de la variable que se desee realizar la conversión.

Cabe mencionar que este tipo de ataque se puede realizar en actualizaciones, consultas con modificación y eliminaciones en la base de datos.

2.5.7.2 Operaciones seguras

El controlador de PHP para MongoDB no espera a que la base de datos responda para realizar escrituras (inserciones, actualizaciones, eliminaciones). Esto quiere decir que las escrituras se llevan a cabo de una manera extremadamente rápida, pero no se sabe si realmente han tenido éxito o no.

Hay muchas razones para que una escritura pueda fallar, entre ellas:

Problemas de Red

Caída del servidor

Escritura inválida (Ej: Escribir en una colección inválida).

Pero existe una opción para obtener una respuesta de la base de datos, se denomina “safe”, que está disponible en todos los tipos de escritura. Esta opción asegura de que la base de datos realiza la escritura antes de notificar su éxito. Si la escritura fallara, entonces emitirá una excepción de Mongo, explicando el fallo. A continuación un ejemplo de uso:

```
$conexion = new Mongo();
$db = $conexion -> BaseEjemplo;
$coleccion = $db -> ColeccionEjemplo;

try
{
    $coleccion -> update(array("nombres" => "marco jaramillo"),
                        array('$set' => array("direccion" => "Totora")),
                        array("safe" => true));
    echo "Actualización realizada con éxito";
}
catch (MongoCursorException $e)
{
    echo "Error en la actualización: ". $e;
}
```

Imagen 197. Actualizar un documento de manera segura con “safe” en MongoDB.

Como se observa en el ejemplo de la imagen anterior, aparte de realizar la condición de búsqueda para actualizar, y de poner el operador “\$set” para actualizar un documento, también ahora se pone al final un array (“safe” => true), para asegurar que la operación de escritura, en este caso actualización, se realice con éxito.

CAPÍTULO 3

HERRAMIENTAS UTILIZADAS EN LA APLICACIÓN

3.1 Introducción

En este capítulo se mostrarán las herramientas utilizadas en la aplicación web que se ha desarrollado, herramientas que se complementarán con el lenguaje HTML5 y la base de datos MongoDB explicados ya en los capítulos anteriores.

Todas las herramientas que se explicarán a continuación están basadas en el lenguaje de programación JavaScript, tanto librerías, API's y NodeJs.

3.2 Librerías JavaScript

Existen varias librerías o también llamadas bibliotecas, que facilitan la programación de un lenguaje y dan una gran cantidad de funcionalidades, en este caso las librerías que se ocuparon para el desarrollo de la aplicación web son enfocadas al lenguaje de programación JavaScript, que a continuación se explicarán.

3.2.1 JQuery

Jquery es una librería o biblioteca que cambia la forma en que se codifica JavaScript.

“JQuery es una biblioteca de JavaScript que simplifica el recorrido de un documento HTML, el manejo de eventos, animación, y las interacciones Ajax para el desarrollo web rápido”⁵.

JQuery es una librería de software libre y de código abierto.

Se puede descargar la versión más actual desde la página oficial: <http://jquery.com/>

⁵ *jQuery*. Septiembre de 2009. 2 de Diciembre de 2012. <<http://jquery.com/>>

Ejemplos básicos:

Para iniciar jquery se hará de la siguiente manera:

```
$(document).ready(function() {  
    //Aquí van todas las acciones del documento.  
});
```

Imagen 198. Ejemplo de inicio de jquery.
Fuente: La Enciclopedia Libre Wikipedia. [Disponible en:
<http://es.wikipedia.org/wiki/JQuery>]

La manera de interactuar con los documentos HTML es mediante una función “\$()” que recibe parámetros como: una expresión CSS o un nombre (Id, Class) de un elemento HTML, y devuelve todos los elementos que concuerden con la expresión solicitada, por ejemplo:

```
$("#tablaAlumnos"); // Devolverá el elemento con id="tablaAlumnos"  
$(".activo");      // Devolverá una matriz de elementos con class="activo"
```

Imagen 199. Ejemplo de interacción con JQuery.
Fuente: La Enciclopedia Libre Wikipedia. [Disponible en:
<http://es.wikipedia.org/wiki/JQuery>]

Se puede realizar animaciones simples a los elementos del documento HTML, como por ejemplo una animación con el método “slideToggle”:

```
// Anima todos los componentes con class="activo"  
$(".activo").slideToggle("slow");
```

Imagen 200. Ejemplo de animación simple JQuery.
Fuente: La Enciclopedia Libre Wikipedia. [Disponible en:
<http://es.wikipedia.org/wiki/JQuery>]

Existen varias funciones y formas de interactuar con JQuery, para minimizar el uso de código extenso y lograr resultandos excelentes en JavaScript.

3.2.2 JQuery UI

Jquery UI es una librería o biblioteca de componentes creada a partir de la librería JQuery, permitiendo interactuar con la interfaz de usuario, crear efectos, crear widgets, entre otras cosas muy interactivas mejorando el aspecto visual y haciendo muy sencillo la codificación.

Jquery UI se puede descargar desde su página oficial: <http://jqueryui.com/>

Ejemplo básico:

Si por ejemplo se desea realizar un efecto que al hacer click en un botón se muestre un elemento HTML **div**, se podría hacer de la siguiente manera:

```
$( "button" ).click(function() {  
    $( "div" ).show( "fold", 1000 );  
});
```

Imagen 201. Ejemplo de efecto con JQuery UI.
Fuente: Documentación JQuery UI. [Disponible en: <http://api.jqueryui.com/>]

3.2.3 TableSorter

Es una librería de JavaScript con el objetivo de ordenar tablas a partir de una tabla estándar creada desde HTML.

Resulta sencillo el manejo de TableSorter ya que lo único que se necesita es asignarle a la tabla que se desea ordenar, la clase “tablesorter” y desde JavaScript iniciar el ordenamiento mediante el método “.tablesorter()” como se muestra a continuación:

```
$(document).ready(function()  
{  
    $("#myTable").tablesorter();  
})  
);
```

Imagen 202. Ejemplo ordenamiento de una tabla con TableSorter.
Fuente: Documentación TableSorter. [Disponible en: <http://tablesorter.com/docs/>]

La librería se puede descargar desde su página oficial:
<http://tablesorter.com/docs/#Download>

Es importante mencionar el uso obligatorio de la librería JQuery para el funcionamiento de TableSorter.

3.2.4 PrintArea

Es una librería de JavaScript creada a partir de JQuery y su objetivo primordial es el de imprimir un elemento de HTML determinado, con solo indicar el ID del elemento a imprimir y desde JavaScript activando la opción de imprimir mediante el método “printArea()”. A continuación un ejemplo de modo de uso para imprimir un elemento **div**:

```
-html-  
<div id="print_button">Print</div>  
  
<div id="myPrintArea">  
    .....html to print.....  
</div>  
  
-javascript-  
$("#div#print_button").click(function(){  
    $("#div#myPrintArea").printArea([options]);  
});
```

Imagen 203. Ejemplo de uso para imprimir un DIV con PrintArea.

Fuente: JQuery Plugins. [Disponible en: <http://archive.plugins.jquery.com/project/PrintArea>]

Se puede descargar ésta librería desde la página de JQuery:
<http://archive.plugins.jquery.com/node/3874/release>

3.2.5 Keyboard

Esta librería de JavaScript es de código abierto y su objetivo es la adición de una interfaz de teclado gráfico para campos de texto, campos de contraseña y áreas de texto para que puedan ser llenados con el ratón solamente.

Cuenta con 92 teclados de acuerdo a los idiomas.

Para comenzar a utilizar se debe descargar un archivo javascript (<http://www.greywyvern.com/code/javascript/keyboard.js>), así como también su hoja de estilos CSS (<http://www.greywyvern.com/code/javascript/keyboard.css>) y también el ícono que se mostrará junto al campo que se desea poner esta interfaz gráfica (<http://www.greywyvern.com/code/javascript/keyboard.png>).

Descargado lo anteriormente mencionado, lo único que resta por hacer es asignarle a un campo de texto cualquiera la clase “keyboardInput” y con esto se podrá observar en el navegador lo siguiente:



Imagen 204. Resultado en el navegador de la librería Keyboard.

3.3 BootStrap

BootStrap es una colección de herramientas para la creación de aplicaciones web, tiene la posibilidad de diseñar plantillas (templates) con HTML y CSS (con clases predefinidas), formularios, botones, gráficos, e inclusive la posibilidad de manejar JavaScript para manipular componentes como: carrusel de imágenes, menús interactivos, transiciones, entre otros.

BootStrap es de código abierto, y se puede descargar última versión desde la página oficial: <http://twitter.github.com/bootstrap/>

Ejemplos básicos:

Para crear por ejemplo un botón grande con un estilo predeterminado, se lo puede realizar de la siguiente manera:

```
<button class="btn btn-large" type="button">Large button</button>
```

Imagen 205. Ejemplo crear un botón grande en Bootstrap.

Fuente: Documentación Bootstrap. [Disponible en: <http://twitter.github.com/bootstrap/base-css.html#buttons>]

Como se observa en la imagen anterior, simplemente se crea un elemento **button** y se le asigna las clases: **btn** (para aplicar al elemento un estilo predefinido de un botón) y **btn-large** (para aplicar al elemento un estilo predefinido de un botón grande).

Se puede crear botones de diferentes tipos, con tan solo indicar el nombre o nombres de las clases correspondientes, como se muestra en la siguiente tabla:

Botón	Class =	Descripción
Default	btn	Botón gris estándar con gradiente
Primario	btn btn-primary	Proporciona peso visual adicional e identifica la acción principal en un conjunto de botones
Informativo	btn btn-info	Usado como una alternativa a los estilos predeterminados.
Éxito	btn btn-success	Indica una acción positiva o exitosa
Advertencia	btn btn-warning	Indica que se debe tener cuidado con esta acción
Peligro	btn btn-danger	Indica una acción peligrosa o potencialmente negativo
Inverso	btn btn-inverse	Alterna el botón gris oscuro, no atado a una acción semántica o uso.
Link	btn btn-link	Restarle importancia a un botón, haciendo que parezca un enlace, manteniendo el comportamiento del botón

Tabla 58. Diferentes tipos de botones en Bootstrap.

Fuente: Documentación Bootstrap. [Disponible en: <http://twitter.github.com/bootstrap/base-css.html#buttons>]

Se puede dar estilo a varios elementos de HTML, como también se pueden crear controles para trabajar junto con JavaScript, por ejemplo para crear un carrusel en el que pueden ir elementos, imágenes, y lo que se quiera introducir en un carrusel. Un ejemplo podría hacer el siguiente:

```

<div id="myCarousel" class="carousel slide">
  <!-- Carousel items -->
  <div class="carousel-inner">
    <div class="active item">...</div>
    <div class="item">...</div>
    <div class="item">...</div>
  </div>
  <!-- Carousel nav -->
  <a class="carousel-control left" href="#myCarousel" data-slide="prev">&lsaquo;</a>
  <a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a>
</div>

```

Imagen 206. Ejemplo crear carrusel con Bootstrap.
 Fuente: Documentación Bootstrap. [Disponible en:
<http://twitter.github.com/bootstrap/javascript.html#carousel>]

Se puede controlar el carrusel desde JavaScript y con la ayuda de JQuery, como por ejemplo, indicándole al carrusel que gire en un intervalo de tiempo de 2 segundos:

```

$('.carousel').carousel({
  interval: 2000
})

```

Imagen 207. Ejemplo de manejo de carrusel de Bootstrap desde JavaScript.
 Fuente: Documentación Bootstrap. [Disponible en:
<http://twitter.github.com/bootstrap/javascript.html#carousel>]

Existen muchas más opciones, estilos, controles que se puede añadir, por ejemplo: alertas de aviso, ventanas modales, varios tipos de tabulaciones (tabs), contenedor de imágenes, tablas con estilos, formularios (forms) y varios controles más.

3.4 API's JavaScript

Una API es un conjunto de funciones y procedimientos o métodos para poder ser utilizado por un software, en este caso por JavaScript, a continuación se listarán las API's utilizadas en la aplicación de este proyecto de tesis, se dará un concepto simple de cada una y ejemplos de uso.

3.4.1 Audio y Video

Las API's de Audio y Video describen un alto nivel de API de JavaScript para el procesamiento y síntesis de audio en aplicaciones web.

Se conectan una serie de objetos juntos para definir la representación de audio o video.

Para manipular el audio o video que se ha colocado en una web mediante los elementos de HTML5 **audio** y **video**, se lo puede hacer mediante varios métodos que surgen de esta API ya sea de video o audio.

Ejemplo:

Por ejemplo mediante los métodos de play(), pause() o volumen, se puede manipular ya sea el audio o el video, como se muestra en el ejemplo siguiente:

```
<script>
  function Play()
  {
    document.getElementById("audio").play();
  }
  function Pause()
  {
    document.getElementById("audio").pause();
  }
  function SubirVolumen()
  {
    document.getElementById("audio").volume += 0.1;
  }
  function BajarVolumen()
  {
    document.getElementById("audio").volume -= 0.1;
  }
</script>
</head>
<body>
  <audio id="audio" src="sonido.mp3" controls preload="auto" loop>
    Tu navegador no soporta AUDIO
  </audio>
  <button onclick="Play()">Play</button>
  <button onclick="Pause()">Pause</button>
  <button onclick="SubirVolumen()">Subir Volumen</button>
  <button onclick="BajarVolumen()">Bajar Volumen</button>
</body>
</html>
```

Imagen 208. Ejemplo API Audio y Video.

Los métodos utilizados en el ejemplo de la imagen anterior, podrán ser utilizados tanto para audio como para video.

3.4.2 Canvas

Como se explicó en el capítulo 1, existe un elemento HTML5 llamado **canvas**, el cual se utiliza para dibujar sobre él: gráficos, gráficos de juego, arte, u otras imágenes visuales, es ahí donde el API Canvas de JavaScript se hace presente para con los métodos que agrega, poder dibujar sobre ese elemento HTML5 **canvas**.

Ejemplos:

A continuación se mostrarán ejemplos de cómo dibujar con del API Canvas algunos gráficos y también cómo realizar una animación simple:

Se necesitaría un elemento **canvas** para poder dibujar sobre él, como por ejemplo:

```
<canvas id="canvas" style="border: solid;" width="300px" height="300px">
  Este navegador no soporta CANVAS
</canvas>
```

Imagen 209. Ejemplo elemento canvas para dibujar con el API Canvas.

Para comenzar ya a codificar el API Canvas de JavaScript se hará lo siguiente:

```
<script>
  function iniciar()
  {
    var canvas = document.getElementById("canvas");
    if (canvas.getContext)
    {
      //Comenzar a codificar CANVAS
    }
    else
    {
      //Navegador no compatible con CANVAS
    }
  }
</script>
</head>
<body onload="iniciar()">
  <canvas id="canvas" style="border: solid 2px;" width="300px" height="300px">
    Este navegador no soporta CANVAS
  </canvas>
</body>
```

Imagen 210. Codificación inicial con el API Canvas.

Como se observa en el codificado de la imagen anterior, lo primero que se hace es declarar una variable y obtener el elemento CANVAS mediante el ID y después mediante una condición IF aseguramos que si “canvas.getContext” (obtiene el

contexto en el que CANVAS va a dibujar) es verdadero, se pueda continuar a programar el CANVAS, caso contrario se indicará al navegador que no es compatible la etiqueta ya que “canvas.getContext” devolverá falso en la condición.

Como ya se mencionó “canvas.getContext” obtiene el contexto en el que CANVAS va a dibujar, habiendo la posibilidad de contexto en 2 dimensiones (2D) y con ayuda de otras tecnologías o librerías hasta 3 dimensiones (3D) (como lo es por ejemplo el proyecto Three.js).

Para comenzar a dibujar en 2D simplemente se declara una variable y se le asigna el elemento CANVAS con la función o método “getContext” y entre paréntesis se escribe: “2d”, de la siguiente manera:

```
if (canvas.getContext)
{
    //Comenzar a codificar CANVAS
    var contexto = canvas.getContext("2d");
}
```

Imagen 211. Obtener contexto para dibujar con API Canvas.

Mediante la variable que se ha definido como lo es “contexto”, podemos partir de ella para empezar con el primer dibujo que a continuación se comenzará a explicar.

Algo muy sencillo de dibujar es un rectángulo o a su vez un cuadrado, como se ve en el siguiente ejemplo:

```
var contexto = canvas.getContext("2d");
contexto.fillRect(0,0,100,200);
contexto.strokeRect(150,50,50,50);
```

Imagen 212. Dibujar un rectángulo o cuadrado con API Canvas.

Como se observa en la imagen anterior, se utiliza el método “fillRect” para dibujar un rectángulo con un relleno y “strokeRect” para hacerlo sin relleno y solo con borde. Estos métodos reciben 4 parámetros, el primero es el origen en X donde comenzará a trazarse el rectángulo (medido en pixeles), el segundo es el origen en Y

(medido en píxeles), el tercer y cuarto parámetro es el ancho y alto respectivamente que tendrá el rectángulo (medido en píxeles). El origen en X y Y se asimila como un plano cartesiano con sus ejes.

En el CANVAS se empieza a medir el origen X desde la esquina superior izquierda del lienzo con dirección hacia la derecha y obviamente el origen Y desde la parte superior izquierda con dirección hacia abajo.

A continuación una imagen la cual nos muestra el resultado de los rectángulos antes codificados y con las acotaciones de las medidas colocadas en los métodos “fillRect” y “strokeRect”:

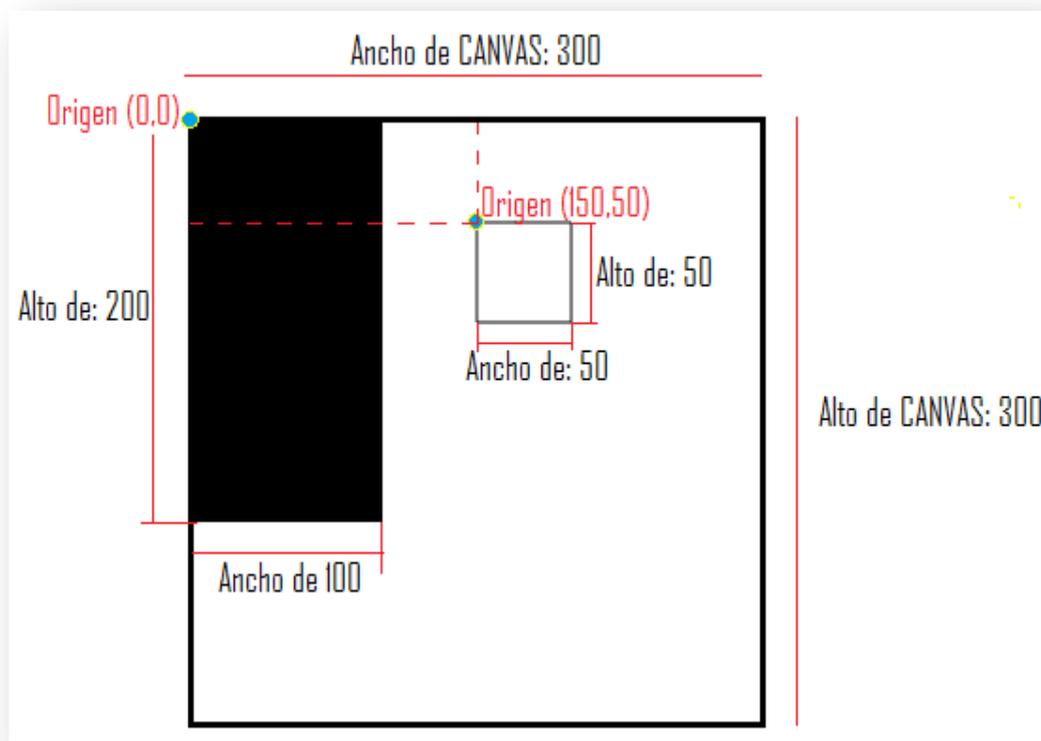


Imagen 213. Resultado de la creación de 2 rectángulos con sus medidas mediante la API Canvas.

Como vemos en la imagen anterior se muestran las figuras dibujadas (rectángulo con relleno negro y cuadrado sin relleno con borde color plomo), que obtienen colores por defecto, aunque hay la posibilidad de cambiar esos colores con otros 2 métodos, expuestos a continuación:

```
var contexto = canvas.getContext("2d");
contexto.fillStyle = 'red';
contexto.fillRect(0,0,100,200);
contexto.strokeStyle = 'blue';
contexto.strokeRect(150,50,50,50);
```

Imagen 214. Dibujar rectángulos con color de relleno en la Api Canvas.

Los métodos utilizados para dar color a un rectángulo creado con el método “fillRect” y “strokeRect”, son “fillStyle” y “strokeStyle” respectivamente, asignándoles una cadena con el color; ya sea con el nombre del color, con el valor en hexadecimal o con la función RGB.

En el navegador el resultado de lo programado anteriormente, se visualizaría de la siguiente forma:

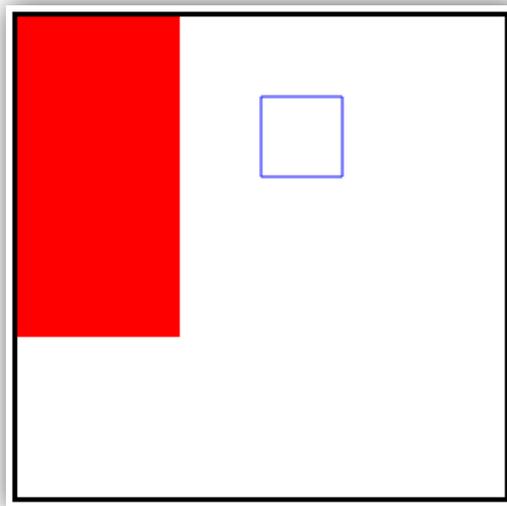


Imagen 215. Resultado en el navegador de la creación de 2 rectángulos con relleno en el API Canvas.

En la programación de CANVAS existe un método o en realidad varios métodos para poder dibujar a placer, trazando lo que se denomina “caminos”. Esto consiste en situarse primeramente en un punto del lienzo e ir colocando los puntos por donde se trazará el dibujo o el camino, para después ir dando forma al trazado con relleno o simplemente con borde.

Para lograr el objetivo de dibujar caminos se necesita de algunas funciones o métodos que se explicará a continuación:

BeginPath(): Este método se utiliza para indicarle al contexto del canvas que se empezará a dibujar un camino, no se le asigna ningún parámetro.

Moveto(): Este método indica un punto imaginario de donde se partirá para dibujar el camino. Es un punto imaginario ya que no se dibuja, ni se muestra nada en el CANVAS, simplemente se indica un punto inicial con los parámetros de posición (x,y).

Lineto(): Esta función o método de CANVAS dibuja una línea recta, desde la posición actual hasta un punto (x,y) que se le indica como parámetro en la función. La posición actual de camino es la que se indicó en la función anterior (moveto()), ó también puede ser la posición donde haya terminado el dibujo de una línea anterior.

Closepath(): Esta función cierra el trazado de un camino, volviendo a su punto inicial de dibujo. Este método o función no recibe ningún parámetro.

Fill(): Se utiliza esta función para rellenar de color un área resultante del trazado de puntos que han dibujado un camino.

Stroke(): Es similar al método fill(), con la diferencia que ahora no rellena de color un área resultante dibujada por el camino, si no que ahora pinta la silueta, borde o para entender mejor, pinta el camino mismo.

Para entender mejor el uso de las funciones descritas anteriormente, a continuación se muestra un ejemplo de cómo dibujar 2 rombos, con y sin relleno:

```
var contexto = canvas.getContext("2d");
//ROMBO CON RELLENO
contexto.beginPath();
contexto.moveTo(100,90);
contexto.lineTo(150,150);
contexto.lineTo(100,210);
contexto.lineTo(50,150);
contexto.closePath();
contexto.fillStyle = 'blue';
contexto.fill();
//ROMBO SIN RELLENO
contexto.beginPath();
contexto.moveTo(200,90);
contexto.lineTo(250,150);
contexto.lineTo(200,210);
contexto.lineTo(150,150);
contexto.closePath();
contexto.strokeStyle = 'red';
contexto.stroke();
```

Imagen 216. Dibujar 2 rombos con caminos en el API Canvas.

Como se observa en la imagen anterior se comienza indicándole al contexto de CANVAS que se empezará a dibujar un camino, con la función BEGINPATH() se le hace saber. Después con MOVETO() se señala el punto de inicio con las coordenadas (x,y), y para ir trazando las líneas e ir dándole forma al rombo, se utiliza LINETO() que une todos los puntos que se irán pasando como parámetros en la función. Para indicar que se cierre el camino, como se observa en la imagen 29, se utiliza CLOSEPATH(). Aquí también se puede ir utilizando las funciones antes vistas para darle un color personal tanto al relleno como al borde, para eso están las funciones FILLSTYLE() y STROKESTYLE() antes explicadas. Por último con las funciones FILL() y STROKE() le indicamos al contexto del CANVAS si es que el camino tendrá o no relleno.

El resultado en el navegador se visualizará de la siguiente manera:

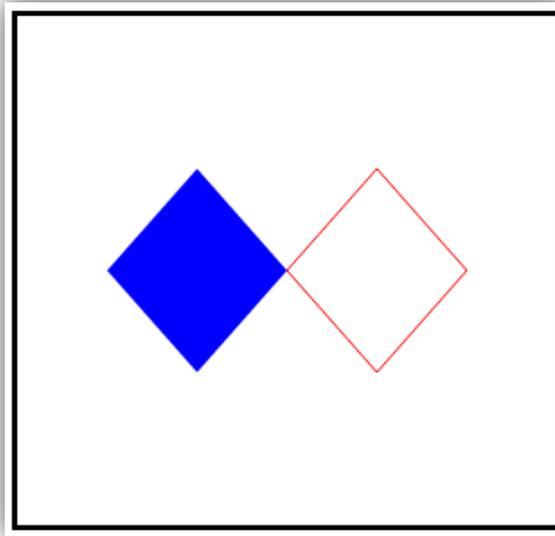


Imagen 217. Resultado de dibujar 2 rombos en el navegador con el API Canvas.

Con los caminos de CANVAS se puede dibujar de una forma libre, con líneas rectas y se puede manipular y pintar con colores que se desee.

No hay que olvidar que se está programando en JavaScript y por lo tanto se puede ir codificando con todas las funciones natas de este lenguaje, como por ejemplo: `setTimeout`, `setInterval`, `math`, y demás, que podrán ir mezclando con los métodos del API Canvas y generando un sinfín de resultados grandiosos.

Otras de las cosas interesantes en CANVAS son las animaciones de las cuales podemos valernos para hacer juegos, entre otras cosas. A continuación se explicará una animación básica para a partir de ello, tener una visión clara para poder lograr animaciones mucho más complejas.

Lo primero que se debe tener claro para una animación, es conocer el concepto de frames, que no es más que secuencias de imágenes, en este caso en diferentes posiciones, que darán a la vista la sensación de movimiento. En JavaScript existe una función que deriva del objeto “window”, ésta cumple el rol de los frames, denominada `Request Animation Frame`, y con esta ya se puede comenzar a codificar la animación básica, expuesta a continuación:

```

var contextoGeneral = canvas.getContext("2d");
var ElementoRectangulo = function(){
  this.ancho = 70;
  this.alto = 70;
  this.x = (canvas.width / 2) - (this.ancho / 2);
  this.y = 0;
  this.dibujar = function(contexto){
    contexto.fillRect(this.x,this.y,this.ancho,this.alto);
  }
};

```

Imagen 218. Creación de una animación de movimiento de un rectángulo con el API Canvas. Parte 1.

Como se muestra en la imagen anterior, lo primero que se hace es declarar una variable “contextoGeneral” que obtendrá el contexto en 2D del CANVAS. Luego se declara una variable tipo clase denominada “ElementoRectangulo”, la cual tendrá sus atributos (ancho, alto, x, y), para poder ir dibujando un rectángulo en diferentes posiciones y dando la impresión de movimiento. De igual manera la variable tipo clase tendrá un método llamado “dibujar”, el cuál como su nombre lo indica dibujará, en este caso un rectángulo con los atributos que se hayan indicado en la clase “ElementoRectangulo”.

El método “dibujar” tendrá una variable que se recogerá como parámetro, denominada “contexto”, la cual se obtiene como un contexto nuevo cada vez que se dibuje un nuevo rectángulo, dibujando éste con la función fillRect(), a la cual le pasamos los cuatro parámetros (x, y, ancho, alto).

```

var Rectangulo = new ElementoRectangulo();
var direccion = "bajar";
function BucleFrames(){
    window.webkitRequestAnimationFrame(BucleFrames);
    contextoGeneral.clearRect(0,0,canvas.width,canvas.height);
    if (direccion == "bajar")
        Rectangulo.y += 1;
    else
        Rectangulo.y -= 1;

    if (Rectangulo.y == (canvas.height - Rectangulo.alto))
        direccion = "subir";
    if (Rectangulo.y == 0)
        direccion = "bajar";

    Rectangulo.dibujar(contextoGeneral);
}
BucleFrames();

```

Imagen 219. Creación de una animación de movimiento de un rectángulo con el API Canvas. Parte 2.

Ahora se declara una variable “Rectangulo” para instanciar o construir desde la clase “ElementoRectangulo” un objeto, de igual manera se declara una variable (direccion) que será la encargada de decidir si el rectángulo se moverá para abajo o para arriba.

Para proceder a simular el movimiento del rectángulo agregamos una función “BucleFrames” con la ayuda de Request Animation Frame. En este ejemplo como se utiliza un motor de renderizado de Chrome llamado Webkit, el cual es el que toma el contenido HTML y lo interpreta de manera visual.

Con WebKit la función para lograr el efecto de frames es “webkitRequestAnimationFrame()”, que recibe por parámetro una función la cual será “BucleFrames” que se repetirá en un intervalo de tiempo ya predeterminado por la función antes mencionada de Webkit.

Al momento de ir dibujando un nuevo rectángulo, se necesita borrar o limpiar el CANVAS para que surja efecto el movimiento, para esto se utiliza la función “clearRect()” la cual recibe cuatro parámetros, es igual a la antes explicada función “fillRect()”, con la diferencia de que ahora ya no dibuja si no limpia un área definida por los parámetros que recibe (x, y, un ancho, y un alto).

De acuerdo con la variable “direccion”, la cual tendrá 2 valores: “bajar” y “subir”, el atributo “y” del objeto “Rectangulo” aumentará en 1 si es que la variable “direccion” tiene el valor de “bajar”, caso contrario el atributo “y” disminuirá en 1, dando así la sensación de movimiento para arriba o para abajo.

A continuación para que la imagen rectángulo no salga de los límites del CANVAS, se pone dos condicionantes, los cuales permiten establecer que si la posición “y” del rectángulo es igual al alto del CANVAS (canvas.height) menos el alto del propio rectángulo dibujado (Rectangulo.alto), entonces la dirección de movimiento del rectángulo será para arriba, y si el rectángulo en la posición “y” es igual a 0 la dirección será para abajo.

Por último dentro de la función “BucleFrames”, se invocará al método de dibujar del objeto “Rectangulo” y se le pasará el parámetro del contexto del CANVAS donde dibujar (Rectangulo.dibujar(contextoGeneral)). Para que todo lo mencionado y explicado antes funcione, obviamente se deberá llamar a la función BucleFrames().

En el navegador se mostrará un CANVAS con el rectángulo en movimiento de arriba para abajo y viceversa.

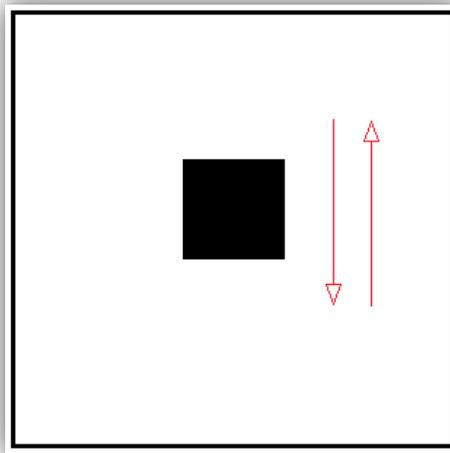


Imagen 220. Resultado en el navegador de la animación de movimiento de un rectángulo con el API Canvas.

En la imagen anterior se puede observar cómo se mostrará en el navegador el movimiento continuo del rectángulo (las flechas mostradas en la imagen son para indicar que el rectángulo está en movimiento).

3.4.3 Geolocation

Esta API de JavaScript permite que el navegador acceda a los datos de la posición (coordenadas de latitud y longitud) del dispositivo desde el que se está accediendo a la web, en este caso puede ser un navegador web.

Ejemplo:

Con el siguiente ejemplo se obtendrá los valores de Latitud y Longitud de ubicación de un dispositivo, mediante los cuales se puede por ejemplo, enviar a la API de Google Maps y mostrar la ubicación en un mapa:

```
<script>
function ubicacion(){
    if(navigator.geolocation){
        navigator.geolocation.getCurrentPosition(function(objetoPosicion){
            var Longitud=objetoPosicion.coords.longitude;
            var Latitud=objetoPosicion.coords.latitude;
        });
    }
}
</script>
</head>
<body onload="ubicacion()">
</body>
```

Imagen 221. Sintaxis para obtener la ubicación (Latitud y Longitud) de navegación mediante la API Geolocation.

En la imagen anterior se puede observar que mediante el objeto de la API Geolocation de JavaScript “navigator”, se puede tener acceso a sus propiedades y obtener la posición.

Lo primero que se puede observar es que se le pregunta al navegador si soporta la obtención de localización, mediante un “If (navigator.geolocation)”, si es que el navegador devuelve “true” entonces significa que si soporta o si hay como trabajar con la API Geolocation.

Ahora mediante la propiedad “getCurrentPosition” se obtendrá un objeto que contiene las coordenadas de Longitud y Latitud.

Algo muy importante que mencionar de la API Geolocation es que por seguridad el navegador pregunta al usuario si es que desea mostrar su ubicación a la página web, el navegador mostrará lo siguiente:

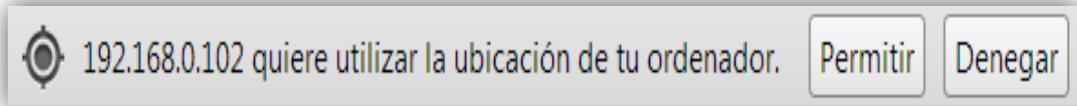


Imagen 222. Seguridad del navegador para conceder el permiso de mostrar la ubicación, mediante Geolocalización.

Así es que si el usuario no permite mostrar la ubicación, no se podrá acceder a las coordenadas.

Ahora para aprovechar la obtención de las coordenadas (Latitud y Longitud) se las puede incluir en algún generador de mapas, como Google Maps u OpenStreet Maps.

A continuación se muestra un código para que de las coordenadas que se obtengan con la API Geolocation de JavaScript, se genere un mapa con la API de Google Maps:

```
<script>
function ubicacion(){
  if(navigator.geolocation){
    navigator.geolocation.getCurrentPosition(function(objetoPosicion){
      var Longitud=objetoPosicion.coords.longitude;
      var Latitud=objetoPosicion.coords.latitude;

      var divMapa=document.getElementById('mapa');
      divMapa.src = "http://maps.google.com/maps/api/staticmap?center=" +
      Latitud + "," + Longitud + "&zoom=17&size=400x400&markers=color:green|label:A|" +
      Latitud + "," + Longitud + "&sensor=false";
    });
  }
}
</script>
</head>
<body onload="ubicacion()" width="100%" height="100%">
  <img id="mapa" src="" />
</body>
```

Imagen 223. Sintaxis para generar un mapa con las coordenadas obtenidas de la API Geolocation, mediante Google Maps.

El código de la imagen anterior lo que hace es obtener la Latitud y Longitud mediante la API Geolocation y enviarle a un vínculo de Google Maps para que muestre el mapa sobre un elemento IMG. A continuación se muestra el resultado:

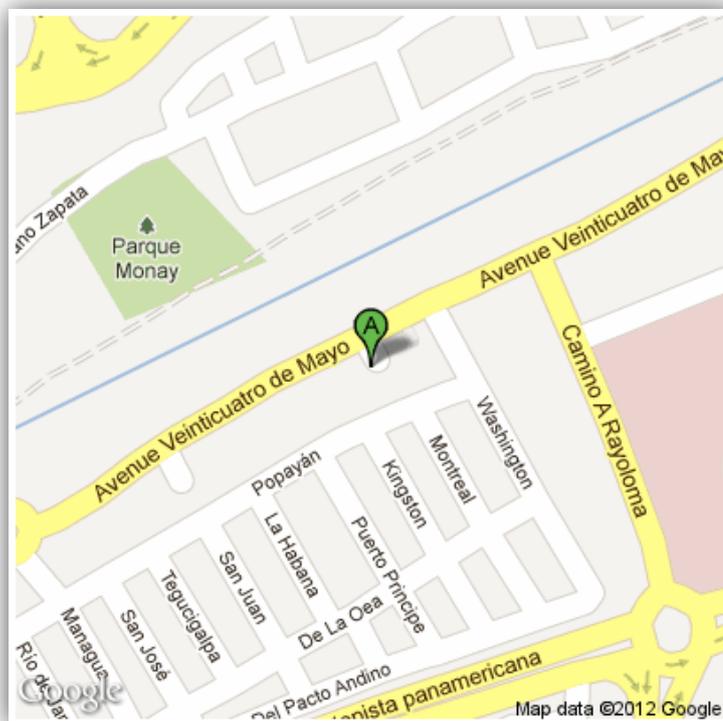


Imagen 224. Resultado en el navegador de la generación de un mapa con Google Maps.

3.4.4 Web Storage

Web Storage es una API de JavaScript que introduce dos mecanismos relacionados, de forma similar a la cookies, para el almacenamiento de datos del lado del cliente.

Normalmente los datos se guardan en pares de datos Llave/Valor (ej. Llave: nombre, Valor: Juan), para manipularlos en el navegador.

Los dos mecanismos de almacenamiento que propone esta API, son: Session Storage y Local Storage. La única diferencia entre estos dos mecanismos es su manera de permanecer almacenado, ya que Local Storage se mantiene almacenado hasta que el usuario elimine el contenido de almacenamiento, y Session Storage se mantiene almacenado hasta que se termine la sesión de un usuario, es decir, cuando un usuario cierre de ventana del navegador donde se encuentra almacenados los datos.

Para utilizar los mecanismos de almacenaje, mencionados anteriormente, existen dos objetos de la API Web Storage, el uno denominado “LocalStorage” para almacenamiento local, y “SessionStorage” para almacenamiento de sesión.

Ejemplos:

A continuación se explicará cómo guardar, obtener y eliminar los datos para utilizarlos en una aplicación web. Se mostrarán ejemplos solo con “LocalStorage”, ya que la manera de codificar es la misma, con la única diferencia que se deberá cambiar “LocalStorage” por “SessionStorage”.

Almacenar datos

Para almacenar los datos se utiliza el método “setItem”, como ya se mencionó antes la manera de guardar los datos es Llave/Valor, por lo tanto para guardar un dato se especificará la llave, y si ya existiera almacenada esa llave pues el dato a guardar se reemplazará. A continuación un ejemplo de cómo guardar:

```
localStorage.setItem('nombre', 'Juan');  
localStorage.setItem('contrasena', '12345');
```

Imagen 225. Sintaxis para almacenar datos con LocalStorage.

Se observa en la imagen anterior que se está almacenando la llave “nombre” con su valor “Juan”, de igual manera la llave “contrasena” con su valor “12345”.

Obtener los datos almacenados

Para obtener los datos que ya estén almacenados, se utiliza el método “getItem”. Por ejemplo para recuperar los datos guardados anteriormente se hará lo siguiente:

```
var nombre = localStorage.getItem('nombre');  
var contrasena = localStorage.getItem('contrasena');
```

Imagen 226. Sintaxis para obtener los datos almacenados con LocalStorage.

Como se observa en la imagen anterior, se está obteniendo los datos de la llave “nombre” y de la llave “contrasena”. Ahora la variable declarada “nombre” tomará el valor “Juan” y la variable “contrasena” tomará el valor “12345”.

Eliminar una Llave/Valor

Para poder eliminar una Llave/Valor determinada, se utiliza el método “removeItem”. Por ejemplo si se quisiera eliminar la Llave “contrasena”, se lo puede hacer de la siguiente forma:

```
localStorage.removeItem('contrasena');
```

Imagen 227. Sintaxis para eliminar una Llave/Valor, almacenada con LocalStorage.

Ahora si se quiere eliminar todo lo que esté almacenado con “LocalStorage” o de igual manera todo lo que esté almacenado con “SessionStorage” se deberá utilizar el método “clear()”.

3.4.5 Web SQL Database

Esta API de JavaScript permite almacenar datos en el PC de cada usuario, de una manera estructural relacional SQL. Teniendo un nombre de la base de datos, tablas, campos y relaciones.

Ejemplo:

Para ver cómo funciona esta API, se irá detallando a continuación cómo crear una base de datos, una tabla, de igual manera las operaciones de agregación, modificación y consultas.

Para abrir o crear una nueva base de datos se hará lo siguiente:

```
var db = openDatabase('BaseDeDatosEjemplo', '1.0', 'Descripción', 2 * 1024 * 1024);
```

Imagen 228. Sintaxis para abrir o crear una base de datos local con el API Web SQL Database.

Como se observa en la imagen anterior, se asigna a una variable (db) una función “openDatabase()”, la cual abrirá una base de datos si estuviera ya creada, caso contrario primero creará la base de datos y abrirá la misma para poder realizar operaciones sobre ella.

Se puede apreciar también en la imagen anterior, que a la función “openDatabase()” se pasan cuatro parámetros, esos parámetros (en su orden) son: nombre

(BaseDeDatosEjemplo), versión (1.0), descripción (Descripción) y tamaño (2 * 1024 * 1024) de la base de datos. El tamaño de la base de datos viene dado en bytes.

Hay que tener en cuenta que la base de datos web es transaccional, así que todas las operaciones o sentencias SQL se harán dentro de una transacción, por lo tanto mientras se esté ejecutando alguna transacción la base de datos se bloqueará hasta finalizar dicha transacción.

Ahora hay que crear la función de transacción donde irán las operaciones de agregación, modificación y consulta.

```
db.transaction(function(tx) {  
    //Aquí se codificará las sentencias SQL  
});
```

Imagen 229. Sintaxis de la función de transacción de una base de datos local con la API Web SQL Database.

Como se muestra en la imagen anterior (se supone que la base de datos ya está creada) con la variable “db” (que contiene la creación de la base de datos) se accederá al método “transaction” para ejecutar una función, la cual creará un objeto (tx), el cual se encargará de realizar las operaciones SQL. Ejemplo a continuación, donde se creará una tabla en la base de datos:

```
db.transaction(function(tx) {  
    tx.executeSql("  
        CREATE TABLE IF NOT EXISTS  
        usuario(userid integer primary key autoincrement,nombre text)");  
});
```

Imagen 230. Sintaxis para crear una tabla de una base de datos local con la API Web SQL Database.

Ahora con el objeto creado “tx”, se puede ejecutar una sentencia SQL, gracias al método “executeSql()”, el cual recibe una cadena de texto en la que se escribe la sentencia. En el caso de la imagen anterior, se está creando una tabla llamada “usuario”, con los campos “userid”, que es la llave primaria de tipo entero y es de autoincremento, y el campo nombre de tipo texto.

Por motivos de espacio, en la imagen anterior se muestra la sentencia de creación de la tabla “usuario” con saltos de línea, pero en realidad deberá ir en una sola línea junto al método “executeSql()” para su correcto funcionamiento.

Ya creada la tabla “usuario”, se puede ir agregando o insertando registros, de la siguiente manera:

```
var nombre = "Juan";
db.transaction(function(tx) {
    tx.executeSql("INSERT INTO usuario(nombre) VALUES(?)", [nombre]);
});
```

Imagen 231. Sintaxis para insertar un registro en una tabla de una base de datos local con la API Web SQL Database.

Como se puede ver en la imagen anterior, se está insertando un registro a la tabla “usuario”, el registro que se está agregando es el nombre “Juan” en el campo “nombre”.

Para consultar los registros que se hayan insertado, se lo puede hacer como en el siguiente ejemplo que se muestra a continuación:

```
db.transaction(function(tx) {
    tx.executeSql("SELECT * FROM usuario", [],
        function(tx, result){
            for(var i=0; i < result.rows.length; i++) {
                alert(result.rows.item(i)['nombre']);
            }
        });
});
```

Imagen 232. Sintaxis para consultar registros en una tabla de una base de datos local con la API Web SQL Database.

Como se muestra en la imagen anterior, ahora con el método “executeSql()” se realiza una consulta SQL, sacando todos los datos de la tabla “usuario”.

Cabe recalcar la diferencia entre ésta sentencia (SELECT) y la antes vista INSERT, donde se ocupaba 2 parámetros en el método “executeSql()”, el segundo parámetro era un array donde se ingresa los valores que se insertarán al nuevo registro. Ahora con SELECT ya no se va a insertar nada, por lo tanto ahora el segundo parámetro será un array vacío ([]) como se puede apreciar en la imagen anterior, pero ahora existirá un tercer parámetro que será una función, la cual sacará el resultado de la consulta SQL y mediante un bucle FOR se podrá ir mostrando los registros que existieran.

Ahora bien si se quiere modificar algún registro, simplemente se tendrá que cambiar la sentencia SQL. Por ejemplo en la base de datos (BaseDeDatosEjemplo) que se ha creado durante toda esta la explicación, se creó igualmente una tabla llamada “usuario” y se agregó un registro en el campo “nombre” y quedó de la siguiente manera:

usuarioid	nombre
1	Juan

Tabla 59. Resultado de un registro agregado en una tabla SQL desde la API Web SQL Database.

Si lo que se desea es, por ejemplo, modificar el nombre “Juan” por “Pedro”, la sentencia SQL quedaría de la siguiente manera:

```
var antiguoNombre = "Juan";
var nuevoNombre = "Pedro";
db.transaction(function(tx) {
    tx.executeSql("UPDATE usuario SET nombre=? WHERE
                 nombre = '"+ antiguoNombre + "'", [nuevoNombre]);
});
```

Imagen 233. Sintaxis para modificar registros en una tabla con la API Web SQL Database.

No hay que olvidar que por cuestiones de espacio, se puede observar de la imagen anterior, que la sentencia de modificado (UPDATE) está con un salto de línea, para su correcto funcionamiento se deberá poner en una sola línea.

Por último, si se desea eliminar un registro, por ejemplo en este caso, el registro de la tabla “usuario” que tenga el nombre de “Pedro”, entonces quedaría la sentencia de la siguiente manera:

```
var nombre = "Pedro";
db.transaction(function(tx) {
    tx.executeSql("DELETE FROM usuario WHERE nombre=?", [nombre]);
});
```

Imagen 234. Sintaxis para eliminar registros en una tabla de una base de datos local con la API Web SQL Database.

3.4.6 Drag and Drop

Esta API de JavaScript, permite arrastrar y soltar elementos HTML5 de un origen a un destino. Para lograr el arrastre se necesita de los eventos nuevos de HTML5, para poder mover elementos de una posición a otra con el ratón (mouse).

JavaScript incluye esta API para con los eventos de HTML5 producir un proceso de arrastre y no valerse de plugins para cumplir con esa función.

Ejemplo:

Primeramente se deberá elegir el elemento al cual se le permitirá arrastrarse de un lugar a otro, para ello se le agrega, a dicho elemento, un atributo llamado “draggable” que tendrá el valor de “true”. Con esto se le indica al navegador que sin ningún problema este elemento podrá moverse a cualquier lugar. En la siguiente imagen se puede ver un ejemplo:

```
<p id="parrafo" draggable="true">
    
    <span>Logo de HTML5</span>
</p>
```

Imagen 235. Sintaxis para que un elemento HTML sea draggable.

Como se puede observar en la imagen anterior se ha creado un párrafo (<p></p>) con ID “parrafo”, el cual tiene un atributo llamado “draggable” con su valor en “true”, dando así el permiso de movilidad del elemento.

Con el permiso para mover del elemento párrafo (p), hay que poner en marcha la secuencia “Drag and Drop”, para ello existe un atributo (evento) que logrará aquello y se denomina “ondragstart”, al cual se le asigna una función con dos parámetros para poder establecer el contenido del objeto que se arrastrará. Como se muestra en la siguiente imagen:

```
<p id="parrafo" draggable="true" ondragstart="drag(this, event)">
  
  <span>Logo de HTML5</span>
</p>
```

Imagen 236. Sintaxis para comenzar a arrastrar un elemento HTML con Drag and Drop.

En la imagen anterior se observa que al atributo o evento “ondragstart” se le asigna una función llamada “drag()”, la cual tiene dos parámetros, el uno (this) para pasar a la función el elemento mismo, y el otro (event) para indicarle al navegador el evento que está ocurriendo. La función en JavaScript se muestra a continuación:

```
function drag(parrafo,evento)
{
  evento.dataTransfer.setData('Text', parrafo.id);
}
```

Imagen 237. Función en JavaScript para comenzar a arrastrar un elemento HTML con Drag and Drop.

Se observa en la imagen anterior la función “drag()”, que recoge en las variables “parrafo” y “evento”, los parámetros antes enviados.

Con la variable “evento” se obtiene el evento mismo que está ocurriendo ese preciso momento, el cual con la propiedad “dataTransfer.setData” del API Drag and Drop, se establece el contenido del elemento para transmitir la carga de datos, y después obtener el contenido en otra posición.

Para indicarle al navegador el destino del elemento que se está arrastrando, se deberá crear un espacio en el cual pueda soltarse el elemento.

En la siguiente imagen se crea una etiqueta DIV donde el objeto o elemento podrá moverse:

```
<div id="contenedor">
  Arrastra hasta aquí
</div>
```

Imagen 238. Elemento destino donde un elemento se arrastrará con Drag and Drop.

Por último se agregaran unos atributos (eventos) al contenedor destino donde se soltarán los elementos que serán arrastrados. En la siguiente imagen se puede observar los atributos:

```
<div id="contenedor" ondrop="drop(this, event)"
  ondragenter="return false" ondragover="return false">
  Arrastra hasta aquí
</div>
```

Imagen 239. Elemento destino con sus atributos, para poder soltar los elementos arrastrados con Drag and Drop.

Como se observa en la imagen anterior existe un atributo o evento (ondrop), que permitirá llamar a la función “drop” para obtener el contenido del elemento que se está arrastrando una vez que el elemento haya sido soltado sobre el contenedor, se pasarán 2 parámetros, uno el elemento mismo, y el otro es el evento que está ocurriendo en el instante de arrastre.

De igual manera existen 2 atributos (eventos) más, uno llamado “ondragenter” que se activa cuando el usuario arrastra el puntero del Mouse sobre el área válida de destino (contenedor), y otro denominado “ondragover”, el cual se activa continuamente cuando el cursor se encuentra sobre el área de destino (contenedor). Esos atributos o eventos mencionados retornarán un valor de “false” para impedir la acción del navegador por defecto.

A continuación se muestra la función “drop() la cual obtendrá el contenido del elemento que se arrastrará”:

```
function drop(contenedor, evento) {  
    var id = evento.dataTransfer.getData('Text');  
    contenedor.appendChild(document.getElementById(id));  
}
```

Imagen 240. Función en JavaScript para obtener el elemento arrastrado con Drag and Drop.

Como se observa en la imagen anterior se recogen los dos parámetros, enviados anteriormente, y con la propiedad “dataTransfer.getData” del evento, se obtendrá el ID del elemento que se está arrastrando y se agregará al contenedor con la función “appendChild()”.

Con la API Drag and Drop y la ayuda de otra API de JavaScript para manejo de archivos (API File) existe la posibilidad de arrastrar archivos desde una ubicación de la PC, hasta el navegador. Para lograr esto se debe realizar lo mismo que se explicó anteriormente, con la diferencia de que ahora en el evento “ondrop” en lugar de utilizar “dataTransfer.getData()” se utilizará “dataTransfer.files”, para con esto acceder a los archivos.

A continuación un ejemplo con el código completo para subir una foto desde el PC al navegador, y poder mostrarla en un elemento DIV:

```

<script>
function procesarArchivos(archivos) {
    var archivo = archivos[0];
    var reader = new FileReader();
    reader.readAsDataURL(archivo);
    reader.onload = function (e) {
        var output = document.getElementById("destino");
        output.style.backgroundImage = "url('" + e.target.result + "')";
    };
}
function drop(evento) {
    evento.preventDefault();
    var archivos = evento.dataTransfer.files;
    procesarArchivos(archivos);
}
</script>
</head>
<body>
<div id="destino" ondrop="drop(event)"
    ondragenter="return false" ondragover="return false">
</div>
</body>

```

Imagen 241. Ejemplo de cómo arrastrar una foto desde la PC al navegador mediante Drag and Drop.

Como se observa en la imagen anterior, existe un DIV con ID “destino”, en el cual se mostrará la foto que será arrastrada desde la PC, este DIV tiene el evento ondrop el cual se activará cuando se suelte un elemento sobre él, en este caso se soltará una foto. Igualmente existen los eventos “ondragover” y “ondragenter”, explicados ya anteriormente.

Cuando se produce el evento “ondrop” se realizará una función llamada “drop()”, donde se pasa el parámetro “event”. Dentro de la función “drop()” se obtiene mediante la variable “evento” el parámetro pasado anteriormente, para poder realizar un “evento.preventDefault()” el cual evita el comportamiento por defecto del navegador, en el caso de no poner el “preventDefault()” el navegador lo que hará es abrir la foto que estemos soltando sobre ella como un archivo normal y no sobre el contenedor (destino). Con la variable “evento” también se accederá a los archivos, en este caso a la foto, mediante la propiedad “dataTransfer.files” que estará asignada en

una variable “archivos”. Después se llamará a una función llamada “procesarArchivos()” pasando el parámetro del archivo al que se accederá y se soltará en el contenedor destino.

Dentro de la función “procesarArchivos()” se podrá leer el archivo (foto) y obtener la URL para mostrar como background en el DIV con ID “destino”. Igualmente se leerá el archivo mediante la instancia de un objeto llamado FileReader, cuando finaliza de leer el archivo, se activa el evento onload del lector y se puede utilizar su atributo result para acceder a los datos del archivo, en este caso se accederá para obtener la URL mediante una propiedad de FileReader llamada “readAsDataURL” y con esta darle un background al DIV con la URL de la foto.

3.4.7 Web Workers

Esta API de JavaScript da la capacidad de ejecutar tareas en segundo plano, con el fin de no embrutecer, es decir, no volver lento las interacciones del usuario con la página web. De esta manera operaciones que necesiten gran cantidad de tiempo para procesamiento, pueden ejecutarse en segundo plano, o más bien dicho pueden ejecutarse en otro archivo “.js”.

En un primer archivo “JavaScript” se instanciará el WebWorker con su ubicación y en otro “JavaScript” estarán las operaciones que se realizarán en segundo plano, es decir en el WebWorker mismo.

Existirá una comunicación bidireccional entre el WebWorker y el archivo principal desde donde fue creado el trabajador.

Ejemplo:

Para crear el WebWorker se hará lo siguiente:

```
var worker = new Worker('worker.js');
```

Imagen 242. Sintaxis para crear un Web Worker.

Como se observa en la imagen anterior se crea un objeto worker dándole un nombre del archivo que va a ejecutar el trabajador, en ese caso se llama “worker.js”, con esto ya se tiene creado un worker listo para que ejecute un script.

Para enviar un mensaje al worker:

```
worker.postMessage("Mensaje 1");
```

Imagen 243. Sintaxis para enviar un mensaje a un Web Worker.

Para enviar un mensaje desde el worker:

```
self.postMessage("Mensaje 2");
```

Imagen 244. Sintaxis para enviar un mensaje desde un Web Worker.

Para recibir un mensaje en el archivo principal:

```
worker.addEventListener('message', function(evento) {  
    var recibido = evento.data;  
});
```

Imagen 245. Sintaxis para recibir un mensaje de un Web Worker.

Para recibir un mensaje en el worker:

```
self.addEventListener('message', function(evento) {  
    var recibido = evento.data;  
});
```

Imagen 246. Sintaxis para recibir un mensaje en el Web Worker.

3.4.8 WebNotifications

Esta API de JavaScript sirve para mostrar notificaciones o avisos a los usuarios sobre eventos que ocurran en la web, pero fuera del contexto de la página web en que el usuario esté navegando.

Se mostrará como una pequeña ventana, algo parecido como ocurre con el programa “Windows Live Messenger”, que al recibir un mensaje aparece un cuadro pequeño de aviso en la parte inferior derecha.

Para que las notificaciones se muestren, de igual manera como en la API Geolocation vista anteriormente, se necesita permiso del usuario caso contrario no se mostrarán absolutamente nada.

Cabe recalcar que ésta API como la mayoría de nuevas herramientas de JavaScript no son compatibles en todos los navegadores y en los ejemplos que se mostrarán de notificaciones se basarán en el navegador Google Chrome.

Ejemplo:

A continuación se muestra cómo preguntar al usuario si concede el permiso de mostrar notificaciones web:

```
<script>
function notificacion()
{
    if (window.webkitNotifications.checkPermission() == 0)
    {
        //AQUI SE CREARÁ LA NOTIFICACIÓN
    }
    else
    {
        window.webkitNotifications.requestPermission();
    }
}
</script>
</head>
<body>
    <input type="button" value="Mostrar Notificacion" onclick="notificacion()">
</body>
```

Imagen 247. Sintaxis para pedir permiso al usuario de mostrar las Notificaciones con la API Web Notifications.

En la imagen anterior se puede observar que se ha creado un INPUT de tipo botón para que cuando se haga click sobre este se dispare una función llamada “notificación()”, en esa función se pregunta al usuario lo siguiente:

```
if (window.webkitNotifications.ckeckPermission == 0)
```

Esto significa que si el objeto window con la propiedad de webkitNotifications tiene el permiso para mostrar las notificaciones (chechPermission == 0), pues entonces se

procederá a crear la notificación (el número 0 significa que se tiene permiso), caso contrario si no se tiene el permiso entonces se agregará la siguiente línea de código para preguntarle al usuario si concede el permiso, esto se hace con la siguiente línea de código:

```
window.webkitNotifications.requestPermission();
```

En el navegador se mostrará lo siguiente:

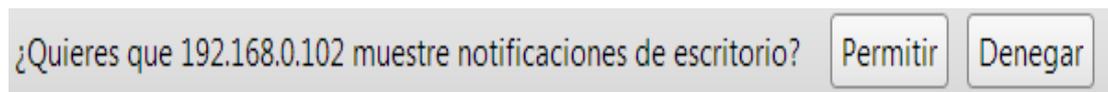


Imagen 248. Seguridad del navegador para conceder el permiso de mostrar Notificaciones con la API Web Notifications.

Así el navegador dará la opción al usuario si quiere mostrar notificaciones web en su navegador.

A continuación se muestra cómo crear y mostrar la notificación, al igual que algunos eventos del objeto “window.webkitNotifications”:

```

<script>
function crearNotificacion() {
    return window.webkitNotifications.createNotification(
        'imagen.jpg', 'Titulo', 'Contenido de la notificación');
}
function notificacion()
{
    if (window.webkitNotifications.checkPermission() == 0)
    {
        var notificacion = crearNotificacion();
        notificacion.onclick = function() {
            notificacion.cancel();
        };
        notificacion.show();
    }
    else
    {
        window.webkitNotifications.requestPermission();
    }
}
</script>
</head>
<body>
    <input type="button"
    value="Mostrar Notificacion" onclick="notificacion()">
</body>

```

Imagen 249. Sintaxis para crear Notificaciones con la API Web Notifications.

Como se observa en la imagen anterior al momento de conocer que el navegador tiene el permiso necesario para mostrar las notificaciones se empieza a crear la notificación misma, para aquello se declara una variable “notificacion” a la cual retornará el resultado de una función llamada “crearNotificacion()”, ésta función es la encargada de crear el objeto en sí de la notificación con “window.webkitNotifications.createNotification()”, donde se pasarán 3 parámetros para la creación, el primer parámetro es el nombre de una foto, el segundo es el título y el tercero un contenido o descripción, los cuales se mostrarán en la notificación. Con esto ya se ha creado la notificación y no queda más que mostrarla, y para ello simplemente se hará con un método denominado “show()”, por lo tanto con la variable “notificacion” que contiene la instancia de la notificación se escribirá “notificacion.show()”, y esto hará que se muestre en pantalla la notificación web.

También se puede observar en la imagen anterior una función que se realizará cuando el usuario de click sobre la notificación (`notificación.onclick`), en esa función se muestra un `“notificación.cancel()”`, eso hará que se oculte la notificación al dar click en la notificación.

El resultado en el navegador cuando se muestre la notificación será como la imagen siguiente:

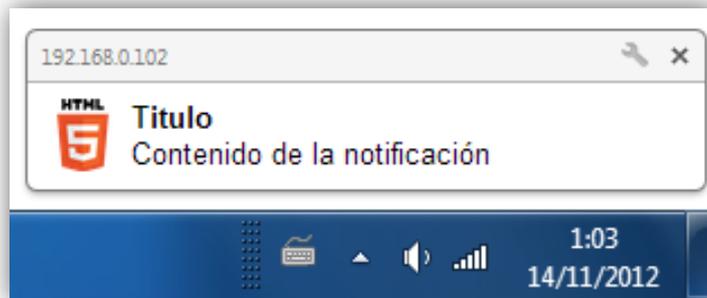


Imagen 250. Resultado en la PC de una Notificación con la API Web Notifications

3.5 Nodejs

Es una plataforma construida en JavaScript para la construcción de aplicaciones rápidas y escalables, ideal para aplicaciones en tiempo real.

NodeJs al contrario que la mayoría de código JavaScript, no se ejecuta en un navegador web, si no que ahora se ejecuta en el lado del servidor escuchando un determinado puerto.

Es como una librería, orientada a eventos, con la posibilidad de crear aplicaciones web complejas desarrolladas completamente en JavaScript.

Posee varios módulos que se pueden agregar, módulos para diferentes fines como: manejo de sockets web, manejo de base de datos, entre otras.

Ejemplo:

Para crear un servidor http en nodejs:

```
var http = require('http');

http.createServer(function (request, response) {
  response.writeHead(200, {'Content-Type': 'text/plain'});
  response.end('Hello World\n');
}).listen(8000);

console.log('Server running at http://127.0.0.1:8000/');
```

Imagen 251. Ejemplo de un servidor http con NodeJS.

Fuente: Enciclopedia Libre Wikipedia. [Disponible en: <http://es.wikipedia.org/wiki/Nodejs>]

El ejemplo anterior, imprimirá en pantalla un “Hello World” (Hola Mundo), aunque antes se debe haber instalado NodeJs e iniciado el servidor node.

Para instalar nodejs, se lo puede descargar desde su página oficial: <http://nodejs.org/>

CAPÍTULO 4

MANUAL DE USO DE LA APLICACIÓN WEB

4.1 Introducción

En este capítulo se proporcionará un manual de uso de la aplicación web desarrollada en este proyecto de tesis, un manual tanto para el usuario como para el administrador de la web.

Primeramente se dará una descripción de la finalidad de la aplicación web, para después ir detallando el uso cada sección de la misma.

4.2 Descripción de la aplicación

La aplicación web que se ha desarrollado en este proyecto de tesis, trata sobre una red social orientada a compartir y ofrecer servicios que puedan prestar los usuarios registrados en ésta.

Ésta aplicación se la ha denominado con el nombre de **PreServi**.

Al momento de registrarse en **PreServi**, se podrá buscar personas que ofrezcan servicios, de interés para cada usuario y de igual forma se podrá publicar los servicios que cada usuario ofrezca y así generar una comunicación con cada uno de los usuarios que formarán parte de los contactos.

Cada usuario registrado recibirá notificaciones, noticias online de las actividades más importantes que realicen sus contactos o usuarios de interés, además habrá la posibilidad de que los usuario registrados tengan un perfil personalizado, pudiendo así mejorar el aspecto de su perfil.

4.3 Manual para usuarios

4.3.1 Página de inicio

A continuación se puede observar la página de inicio de la aplicación, con sus secciones principales de contenido, marcadas en recuadros de color rojo:



Imagen 252. Página principal de la aplicación web.

En la imagen anterior se puede observar la página inicial cuando se ingrese a la aplicación web denominada **PreServi**. Se puede notar que existen 3 secciones principales, que se han marcado en recuadros de color rojo:

- Registrarse
- Información sobre la aplicación
- Iniciar sesión

Más adelante se explicará cómo utilizar cada una de esas secciones.

4.3.1.1 Registrarse

En la página de inicio de **PreServi** existe la sección de Registrarse, en la que se procederá a llenar los campos de datos para formar parte de esta aplicación, como se muestra a continuación:

Regístrate

Nombres: → **Campo nombres**

Apellidos: → **Campo apellidos**

Correo electrónico: → **Campo email**

Repite tu correo electrónico: → **Campo email de confirmación**

Contraseña: → **Campo contraseña**

Repite la contraseña: → **Campo contraseña de confirmación**

Teléfono: → **Campo teléfono**

Dirección: → **Campo dirección**

→ **Botón para completar el registro**

Imagen 253. Sección Registrarse en la aplicación web.

Como se puede observar en la imagen anterior, existen 8 campos de datos que se deben llenar para poder registrarse en **PreServi** y poder iniciar sesión y acceder a las herramientas que ofrece esta aplicación, al igual que existe un botón para presionar una vez completado el ingreso de todos los campos.

Cada campo de datos que se debe completar o llenar es obligatorio y tiene ciertas condiciones o validaciones de ingreso, a continuación se explicará las condiciones de cada campo:

Campo nombres

En este campo el usuario deberá ingresar sus dos nombres obligatoriamente, separados por un espacio.

Campo apellidos

En este campo el usuario deberá ingresar sus dos apellidos obligatoriamente, separados por un espacio.

Campo email

En este campo el usuario deberá ingresar su correo electrónico (email), debe ser un correo electrónico válido.

El email ingresado servirá para iniciar sesión en la aplicación web.

Campo email de confirmación

En este campo el usuario deberá ingresar nuevamente el correo electrónico que escribió antes, es una manera de seguridad para que el usuario confirme si el correo está escrito correctamente.

Campo contraseña

En este campo el usuario deberá ingresar una contraseña para dar una seguridad a su cuenta, ya que con esta contraseña podrá iniciar sesión en la aplicación web.

La contraseña que el usuario ingrese deberá cumplir con las siguientes condiciones:

- Se debe ingresar entre 8 y 10 caracteres
- Entre los caracteres debe existir por lo menos un dígito (número) y un alfanumérico (letra)
- No debe ingresar caracteres especiales (\$%{#!.)

Campo contraseña de confirmación

En este campo el usuario deberá ingresar nuevamente la contraseña que escribió antes, es una manera de seguridad para que el usuario confirme si la contraseña está escrita correctamente.

Campo teléfono

En este campo el usuario deberá ingresar su número de teléfono. En este campo no existe ninguna condición especial, a excepción de que debe ser llenado obligatoriamente.

Campo dirección

En este campo el usuario deberá ingresar su dirección de domicilio. En este campo no existe ninguna condición especial, a excepción de que debe ser llenado obligatoriamente.

Una vez llenado todos los campos y cumpliendo con las condiciones antes mencionadas, se podrá finalizar el registro de usuario haciendo click en el botón para completar el registro denominado “Registrarme” (color azul).

La aplicación web guardará sus datos y mostrará en pantalla el siguiente mensaje de finalización:



Imagen 254. Mensaje de finalización de registro en la aplicación web.

4.3.1.2 Información sobre la aplicación

En la página de inicio de **PreServi** existe la sección de Información sobre la aplicación, que consta de dos botones, uno para leer la descripción sobre la aplicación y el otro botón para ver un video de demostración de la aplicación web, a continuación se mostrará la sección:

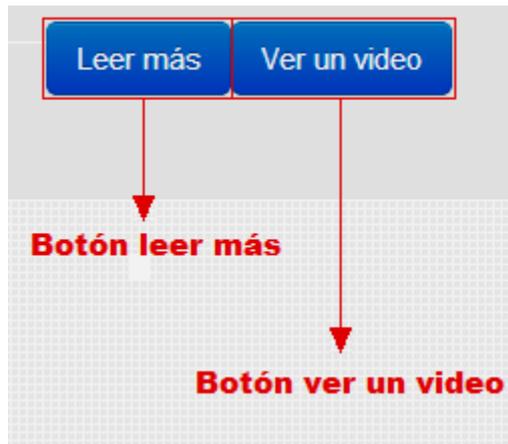


Imagen 255. Sección Información sobre la aplicación.

Como se mencionó anteriormente existen 2 botones: Botón leer más y Botón ver un video, los cuales darán acceso a información y una demostración de la aplicación.

Botón leer más

Al hacer click sobre este botón se mostrará una ventana con una descripción breve sobre las características de **PreServi**, como se muestra a continuación:



Imagen 256. Descripción breve de las características de la aplicación web.

En la imagen anterior se muestra la ventana de descripción de la aplicación, en la que se muestra las características de **PreServi** y de igual manera una foto del autor de la aplicación.

Botón Ver un video

Al hacer click sobre este botón se mostrará una ventana con un video de demostración de **PreServi**, como se muestra a continuación:

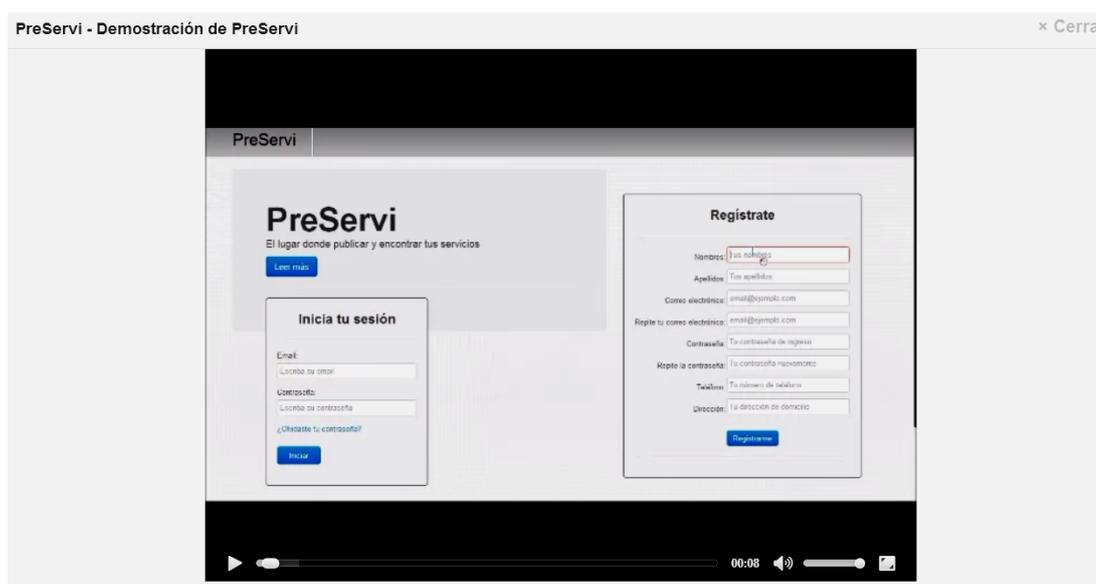


Imagen 257. Video de demostración de utilización de la aplicación web.

4.3.1.3 Iniciar sesión

En la página de inicio de **PreServi** existe la sección de Iniciar sesión, para así comenzar a utilizar la aplicación web y todas sus características, siempre y cuando el usuario se haya registrado anteriormente.

Se podrá iniciar sesión con el email y la contraseña que el usuario se registró.

A continuación se muestra una imagen de la sección de Iniciar sesión:



Imagen 258. Sección Iniciar sesión.

La sección de Iniciar sesión como se observa en la imagen anterior, tiene dos campos de datos, el uno para ingresar (escribir) el email (correo electrónico) y el otro para ingresar la contraseña, y un botón para iniciar la aplicación con la cuenta del usuario.

Campo email

Se debe ingresar el correo electrónico con el que el usuario se ha registrado en **PreServi**.

Campo contraseña

Se debe ingresar la contraseña con el que el usuario se ha registrado en **PreServi**.

Botón iniciar

Cuando se hayan ingresado los campos anteriores, y si los datos son los correctos, la aplicación iniciará con la cuenta del usuario y se mostrará la Página principal de usuarios.

4.3.2 Página principal de usuarios

Cuando el usuario ha iniciado sesión la página que se mostrará es la principal de usuarios.

A continuación se puede observar la página principal de usuarios, con sus secciones principales de contenido, marcadas en recuadros de color rojo:



Imagen 259. Página principal de usuarios de la aplicación web.

La página principal de usuarios de **PreServi** se divide en 5 secciones principales: Contenidos, Menú Superior, Menú Izquierdo, Barra de Información y Herramienta de chat. A continuación se explicará cada sección:

4.3.2.1 Contenidos

Esta sección de Contenidos es donde se mostrará el contenido de cada una de las funciones del Menú Izquierdo, por ejemplo en la imagen anterior se puede observar el contenido de la función Noticias, a continuación se muestra una imagen de ésta sección:

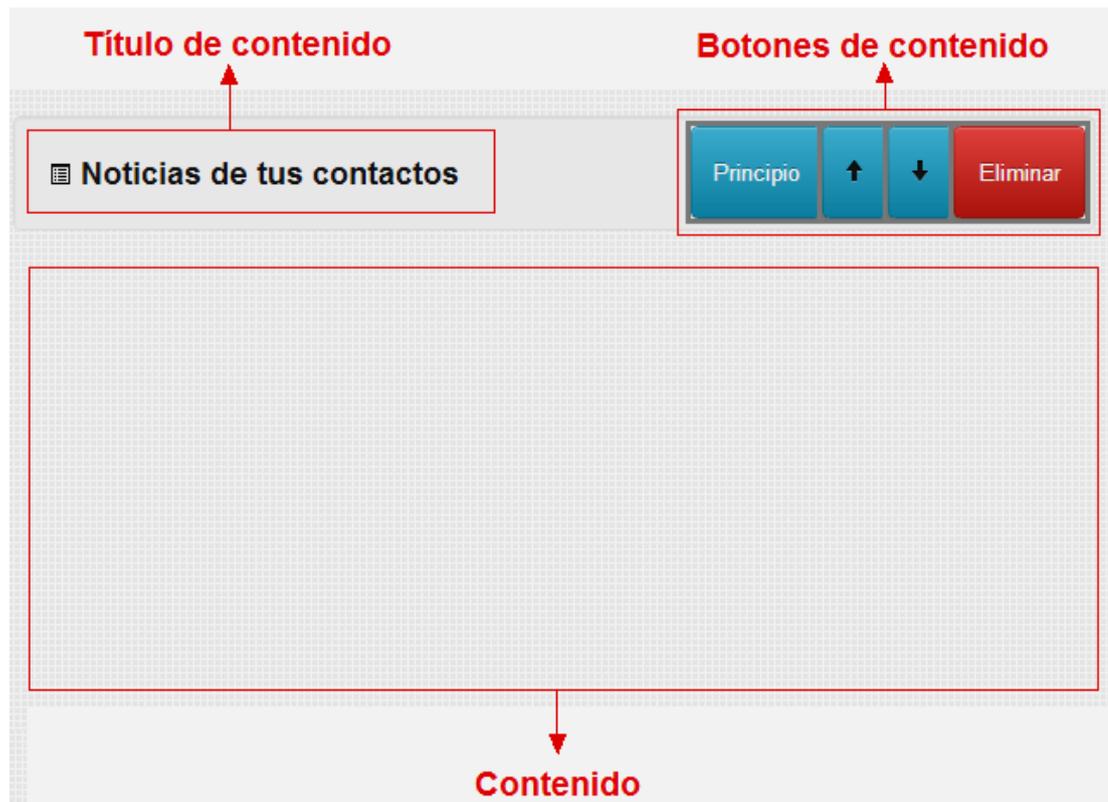


Imagen 260. Sección Contenidos de la aplicación web.

En la imagen anterior se puede observar la sección Contenidos, el cual posee 3 partes:

Un título de contenido: El título del contenido, es decir el título del contenido de la función que se visualizará, por ejemplo en la imagen anterior se muestra el título de la función Noticias (Noticias de tus contactos).

Botones de contenido: Son botones de manipulación del contenido.

Contenido: Es el contenido en sí de la función que se ha seleccionado.

Es importante mencionar que los Botones de contenido solo se encuentran disponibles para las funciones de: Noticias, Mensajes y Ubicación.

4.3.2.2 Menú superior

El menú superior es un menú que se compone de 4 botones: Botón Nombre de usuario, Botón Ver mi perfil, Botón Inicio, Botón Cerrar sesión, y un Buscador de usuarios o personas.

El menú superior, es un menú que siempre estará a la vista del usuario, por lo tanto se podrá acceder a él en cualquier momento.

A continuación se muestra el menú, remarcadas en cuadros de color rojo sus diferentes composiciones:



Imagen 261. Menú superior de la aplicación web.

4.3.2.2.1 Botón Nombre de usuario

Este botón contiene el primer nombre del usuario que inició sesión, y al hacer click sobre este botón inmediatamente la aplicación accederá a la Página principal de usuarios, explicada anteriormente.

4.3.2.2.2 Botón Ver mi perfil

Es botón como su nombre lo indica, muestra el perfil del usuario que ha iniciado su sesión. Éste perfil es la página que los demás usuarios mirarán, es el perfil público de cada usuario.

Por ejemplo si el usuario de nombre “Juan” da click en el Botón Ver mi perfil para revisar cómo los demás observarán su perfil, se mostrará lo siguiente:



Imagen 262. Ejemplo de usuario "Juan" al revisar su perfil en la aplicación web.

La imagen que se muestra es el perfil del usuario “Juan”, es el perfil que será visto por los demás usuarios, el contenido de este perfil se explicará más adelante.

4.3.2.2.3 Buscador de usuarios ó personas

Es un campo de datos, donde se puede ingresar el nombre, servicio o correo electrónico de la persona que se desea buscar, y enseguida la aplicación buscará al usuario/s que cumplan con lo solicitado y mostrará una lista de ellos. Por ejemplo si se desea buscar una persona por nombre “juan”, se mostrará algo así:



Imagen 263. Ejemplo de utilización del Buscador de usuarios en la aplicación web.

Como se observa en la imagen anterior al momento de escribir el nombre “juan”, aparecen los usuarios que coincidan con el nombre “juan”.

De igual manera si el usuario está buscando personas que ofrezcan o busquen un servicio cualquiera, como por ejemplo, un Ingeniero de Sistemas pues entonces deberá ingresar “Ingeniero de Sistemas” en el campo de datos y se listarán las coincidencias.

También se puede buscar a un usuario, por el correo electrónico.

Al momento de que se listan los usuarios que coincidan con lo que se está buscando, al pasar con el ratón (mouse) sobre cada usuario, se mostrará un cuadro con el nombre y servicios que busca y ofrece dicho usuario, como se muestra a continuación:

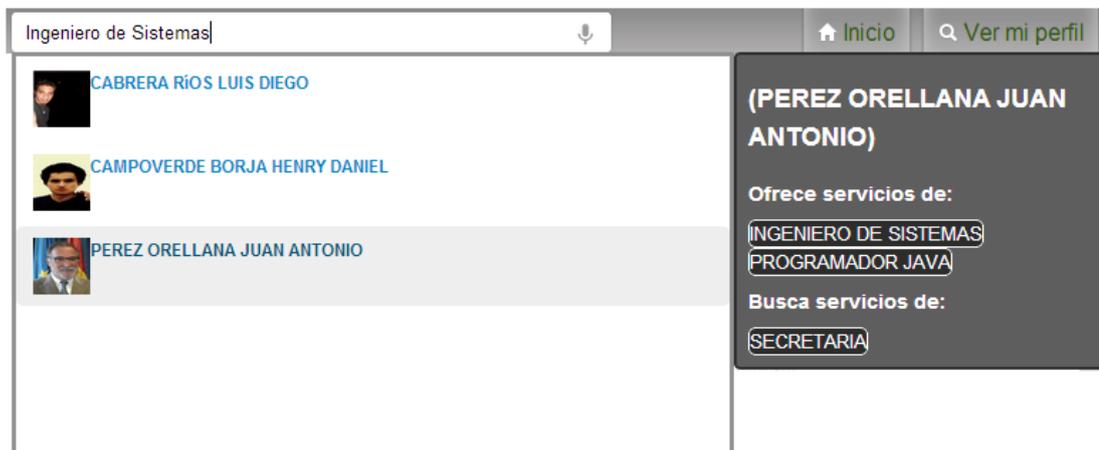


Imagen 264. Ejemplo de búsqueda de usuarios, mostrando el cuadro de descripción del usuario en la aplicación web.

Como se muestra en la imagen anterior, se muestra un cuadro de descripción de color plomo del usuario al cual se le señale con el mouse.

Ahora si se quiere revisar el perfil completo del usuario, simplemente se debe dar click sobre él.

Por ejemplo si se quiere revisar el perfil de “Henry”, se buscará en el Buscador de usuarios y se dará click sobre él, entonces aparecerá el perfil de ese usuario:



Imagen 265. Ejemplo para revisar el perfil de un usuario encontrado con el Buscador de usuarios en la aplicación web.

Como se observa en la imagen anterior, se puede notar el perfil del usuario “Henry”, donde se muestra sus datos personales, los servicios que ofrece y busca, entre otras cosas que se explicará en el tema: “Perfil visible a los demás usuarios”.

También se puede observar que existe un botón de color verde en la parte superior izquierda, haciendo click en ese botón se eliminará/agregará ese usuario (en este caso el usuario Henry) al contenido de la función Contactos y a la Herramienta de chat.

4.3.2.2.4 Botón inicio:

Al hacer click sobre este botón, al igual que el Botón Nombre de Usuario, la aplicación accederá a la Página principal de usuarios, explicada anteriormente.

4.3.2.2.5 Botón Cerrar sesión

Este botón sirve para cerrar sesión del usuario que esté conectado (que haya iniciado sesión), al hacer click sobre este botón se cerrará la sesión y la aplicación accederá o mostrará la Página de inicio, explicada anteriormente.

4.3.2.3 Menú izquierdo

El Menú izquierdo es un menú de navegación desde el cual se podrá acceder a las funciones para los usuarios de **PreServi** como: Noticias, Mi Cuenta, Personalización, Servicios, Fotos, Contactos, Mensajes, Ubicación, Currículum, Entretenimiento, y además existe la Foto de perfil actual de los usuarios.

Cuando se accede a cada una de las funciones del Menú Izquierdo, su contenido se visualizará en la sección Contenidos (ubicado en la mitad de la página).

Cabe recalcar que al iniciar sesión la primera función que se mostrará es la de Noticias, como se puede observar en la imagen anterior.

A continuación se puede observar la sección de Menú izquierdo con cada una de sus funciones y la Foto de perfil de los usuarios:



Imagen 266. Menú Izquierdo de la aplicación web.

Como se indicó anteriormente, la función Noticias siempre se mostrará cuando se inicie la aplicación, por ese motivo se puede observar en la imagen anterior que la

función Noticias está de un color diferente (blanco) que las demás, indicando con esto que la función está señalada y se visualizará en la sección Contenidos.

A continuación se explicará cada una de las funciones y la foto de perfil:

4.3.2.3.1 Foto de perfil



Imagen 267. Foto de perfil actual inicial de la aplicación web.

Se asignará a cada usuario la foto de perfil que se muestra en la imagen anterior, hasta que el usuario cambie a su foto preferida.

La foto de perfil será con la que otros usuarios se reconocerán al momento de buscar a otros usuarios.

4.3.2.3.2 Función Noticias



Imagen 268. Función Noticias de la aplicación web.

Esta función recibe Noticias en tiempo real (online), noticias de las actividades más importantes de los contactos que el usuario haya agregado, y que aparecerán en la Herramienta de chat y en la función Contactos (que se explicarán más adelante).

En esta función Noticias existen los Botones de contenido para manipular las noticias, como son:

Botón *Principio*: Se dirige al principio de las noticias.

Botón *flecha para abajo* (↓): Se desplaza la página en dirección hacia abajo, para revisar las noticias más antiguas.

Es importante mencionar que si se tiene muchas noticias, no se mostrarán todas a la vez, mientras se vaya desplazando hacia abajo aparecerán de a poco las noticias.

Botón flecha para arriba (↑): Se desplaza la página en dirección hacia arriba, para revisar las noticias más actuales.

Botón Eliminar: Elimina todas las noticias que tenga el usuario (Mensaje de confirmación al momento de intentar eliminar las noticias).

4.3.2.3.3 Mi cuenta

Esta función se utiliza para modificar la cuenta de cada usuario, se divide en 2 secciones:

Datos Personales: Para editar o cambiar los datos personales del usuario.

Configuración: Para configurar la cuenta del usuario, como: habilitar/deshabilitar notificaciones de escritorio, cambiar contraseña y eliminar cuenta del usuario.

4.3.2.3.3.1 Datos Personales



Foto de perfil actual

JUAN MANUEL

Datos Personales Configuración

Nombres: JUAN MANUEL **Teléfono:** 0998765432

Apellidos: Pérez CASTILLO **Dirección:** Cdla. Totoracoha

Ciudad: Cuenca **País:** Ecuador

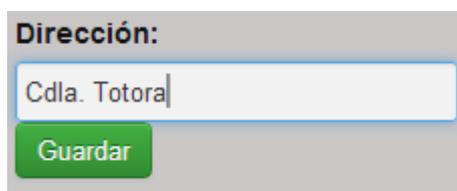
E-mail: perezcastillo@gmail.com

Click en cada uno de los datos personales para editarlos.

Imagen 269. Función Mi cuenta, sección Datos personales de la aplicación web.

En esta sección se puede modificar los datos personales del usuario, con tan solo dar click sobre el campo de dato, editar y después dar click sobre un botón que aparecerá

automáticamente para guardar lo modificado. Por ejemplo para editar la dirección del usuario:



A screenshot of a web form titled "Dirección:". It features a text input field containing "Cdla. Totorá" and a green "Guardar" button below it.

Imagen 270. Ejemplo de modificación de un campo en la función Mi cuenta, sección Datos personales de la aplicación web.

Es importante mencionar que el campo de dato email (correo electrónico), no se puede modificar.

4.3.2.3.2 Configuración



A screenshot of the "Configuración" section in the "Mi cuenta" area. On the left is a sidebar with a profile picture placeholder and a menu including "Noticias", "Mi cuenta", "Personalización", "Servicios", "Fotos", "Contactos", "Mensajes", "Ubicación", "Currículum", and "Entretenimiento". The main content area shows the user's name "JUAN MANUEL" and two tabs: "Datos Personales" and "Configuración". Under "Configuración", there are three sections: "Notificaciones de escritorio" with radio buttons for "Habilitadas" and "Deshabilitadas" (selected), and a "Más información" link; "Cambiar contraseña" with a "Cambiar" button; and "Eliminar mi cuenta de PreServi" with an "Eliminar" button.

Imagen 271. Función Mi cuenta, sección Configuración de la aplicación web.

En ésta sección el usuario puede configurar su cuenta, permitiendo habilitar o deshabilitar las notificaciones de escritorio, cambiar la contraseña con la cual se registró, y por último el usuario puede eliminar su cuenta en **PreServi**.

Notificaciones de escritorio

Las notificaciones de escritorio son avisos que aparecerán como una ventana pequeña cuando le lleguen mensajes a los usuarios, así no estén navegando en la página de **PreServi**.

Por defecto **PreServi** deshabilita las notificaciones de escritorio, aunque se puede habilitar con tan solo dar click en “Habilitadas” o deshabilitar dando click en “Deshabilitadas”, como se muestra en la imagen siguiente:



Imagen 272. Habilitar/Deshabilitar notificaciones de escritorio en la aplicación web.

Por razones de seguridad el navegador web (Google Chrome) pide una autorización al usuario para mostrar las notificaciones, así que el usuario deberá dar acceso al navegador a que muestre o no las notificaciones:

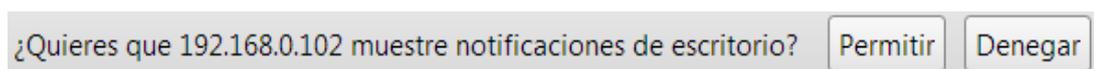


Imagen 273. Permiso para notificaciones de escritorio en la aplicación web.

Cambiar contraseña



Imagen 274. Acceso a cambiar contraseña de usuario de la aplicación web.

Para cambiar la contraseña del usuario simplemente haciendo click sobre el botón “Cambiar”, se accederá a una ventana como la siguiente:

PreServi - Cambiar contraseña

Ingrese contraseña actual:

Ingrese contraseña nueva:

Repita contraseña nueva:

Imagen 275. Ventana para cambiar la contraseña del usuario de la aplicación web.

En la imagen anterior se observa la ventana para cambiar la contraseña del usuario, la aplicación por seguridad pide al usuario ingresar 3 campos de datos: La contraseña anterior, la nueva contraseña, y la reiteración de la nueva contraseña para asegurarse de que esté bien escrita. Una vez ingresado los datos en los campos simplemente con un click en el botón “Cambiar” la contraseña será cambiada de inmediato.

Eliminar cuenta de la aplicación

Eliminar mi cuenta de PreServi

Imagen 276. Acceso a eliminar cuenta de usuario de la aplicación web.

Para eliminar la cuenta de **PreServi** simplemente haciendo click sobre el botón “Eliminar”, se accederá a una ventana como la siguiente:



Imagen 277. Ventana para eliminar cuenta del usuario de la aplicación web.

En la imagen anterior se observa la ventana para eliminar la cuenta del usuario, la aplicación por seguridad pide al usuario ingresar un campo de dato: La contraseña de la cuenta del usuario, haciendo click en el botón “Eliminar mi cuenta” se perderán todos los datos del usuario y no habrá forma de recuperarlos.

4.3.2.3.4 Función Personalización



Imagen 278. Función Personalización de la aplicación web.

Esta función sirve para cambiar el aspecto del perfil del usuario, como se observa en la imagen anterior existen 3 partes importantes como son: La Selección de tema, Vista previa del tema y un Botón guardar.

Selección de tema:

Aquí se podrá elegir el tema del perfil del usuario, existen 4 temas:

Predeterminado:

Este tema muestra un fondo negro de tablas con su formato predeterminado por la aplicación web, es el que se le asigna por defecto a todos los perfiles de los usuarios de **PreServi**.

Cuadrícula:

Este tema muestra un fondo de cuadrículas azules con su formato (tamaño de letra, color de letra, etc) predeterminado por la aplicación web.

Cielo:

Este tema muestra un fondo de cielo con su formato (tamaño de letra, color de letra, etc.) predeterminado por la aplicación web.

Personalizado:

Este tema es el más importante ya que permite dar una personalización del fondo de perfil del usuario, al igual que personaliza su formato (tamaño letra, color letra, etc.).

Al momento de elegir este tema se mostrará lo siguiente:

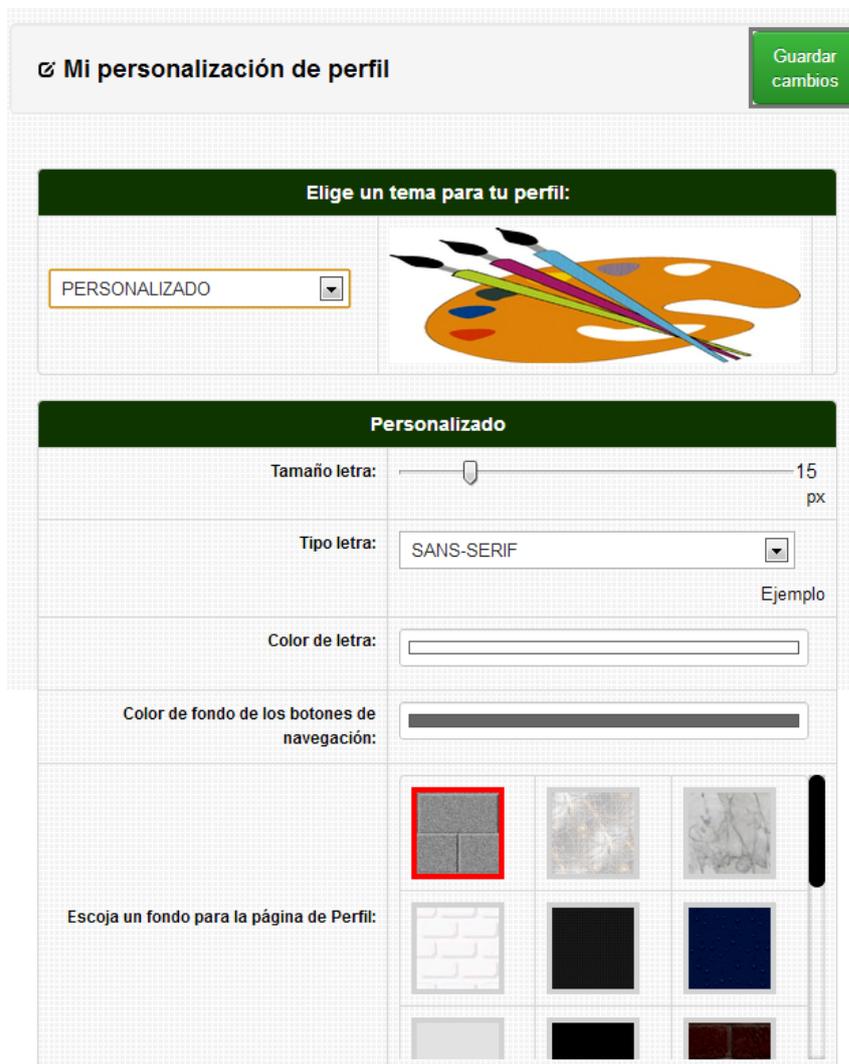


Imagen 279. Función Personalización, tema personalizado de la aplicación web.

Como se observa en la imagen anterior, al momento de escoger el tema Personalizado, se mostrarán opciones como: cambiar el tamaño de letra, tipo de letra, color de letra, y algunas opciones más personalizadas del perfil del usuario.

Para guardar o realizar los cambios que se han realizado solo hay que hacer click en el botón “Guardar cambios”.

Vista previa del tema:

Aquí se mostrará una foto de vista previa del tema que se escoja.

Botón guardar:

Haciendo click en este botón se guardará el tema que se haya escogido, y se modificará en el perfil del usuario.

4.3.2.3.5 Función Servicios



Imagen 280. Función Servicios de la aplicación web.

El objetivo de ésta función es agregar los servicios que el usuario busca y ofrece, para que esos servicios puedan ser publicados su perfil y pueda ser encontrado con esos servicios por otros usuarios.

Como se observa en la imagen anterior, el contenido de la Función Servicios, está compuesta por 4 partes:

4.3.2.3.5.1 Agregar un servicio para ofrecer:

En esta parte del contenido, existe un campo de datos en el cual se puede ingresar un servicio que el usuario ofrezca a los demás usuarios, por ejemplo: Carpintero, Ingeniero Civil, Zapatero, etc, y ya después de haber ingresado el servicio se puede dar click en el botón “+ Publicar” para que se muestre o publique en el perfil del usuario y así poder ser encontrado por otros usuarios.

Lista de servicios para ofrecer:

Aquí se mostrará una lista de los servicios para ofrecer que hayas agregado, y en cada servicio se mostrará un signo “X” para poder eliminar si es que se desea.

Por ejemplo si se ha agregado un servicio “CARPINTERO”, y por alguna razón se desea eliminar, pues solo se deberá hacer click sobre el signo “X”, como se muestra a continuación:



Imagen 281. Ejemplo de agregar un servicio con la opción de eliminarlo en la aplicación web.

Como se puede observar en la imagen anterior, se ha agregado un servicio “CARPINTERO” y si se desea eliminarlo, simplemente se deberá dar click en el signo “X”.

4.3.2.3.5.2 Agregar un servicio que se busca:

En esta parte del contenido, existe un campo de datos en el cual se puede ingresar un servicio que el usuario necesita o busca, por ejemplo: Carpintero, Ingeniero Civil, Zapatero, etc, y ya después de haber ingresado el servicio se puede dar click en el botón “+ Publicar” para que se muestre o publique en el perfil del usuario y así poder ser encontrado por otros usuarios.

Lista de servicios que se busca:

Aquí se mostrará una lista de los servicios que el usuario busca o necesita, y en cada servicio se mostrará un signo “X” para poder eliminar si es que se desea.

Por ejemplo si se ha agregado un servicio “CARPINTERO”, y por alguna razón se desea eliminar, pues solo se deberá hacer click sobre el signo “X”.

4.3.2.3.6 Función Fotos



Imagen 282. Función Fotos de la aplicación web.

Esta función lo que realiza es agregar fotos para el usuario, y tenerlas en una lista para poder elegir una y colocarla como foto de perfil.

De ésta función se destacan 3 principales cosas: Botón agregar, Lista de fotos y el Cambio de foto de perfil.

4.3.2.3.6.1 Botón agregar:

Este botón lo que hace es desplegar una ventana de diálogo para elegir la foto que el usuario desee agregar a su lista de fotos, como se muestra a continuación:

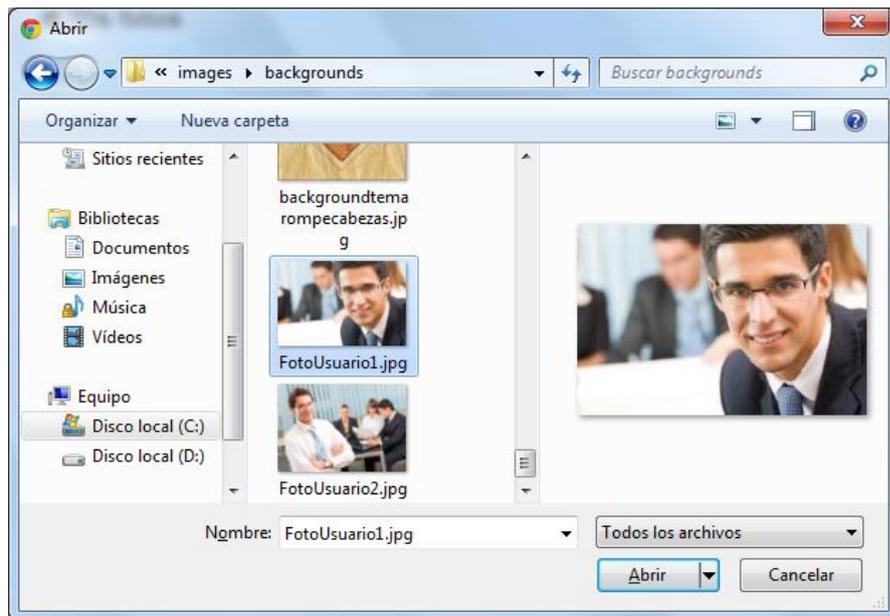


Imagen 283. Ejemplo de elegir una foto para agregar a la lista de fotos de un usuario en la aplicación web.

Al momento de que se elige la foto y se procede a hacer click en el botón “Abrir”, la foto comenzará a guardar o subir en la aplicación web, se mostrará una barra de progreso, dando a conocer o mostrando al usuario cuanto falta para que se acabe de completar la subida de la foto, como se muestra a continuación:

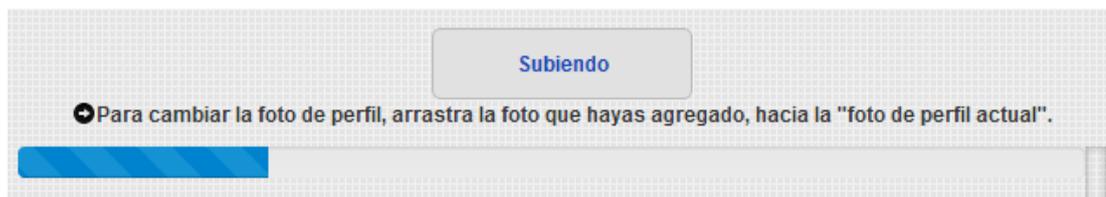


Imagen 284. Ejemplo del progreso de subida de una foto en la aplicación web.

Se podrán subir fotos, máximo de 2 Mega Bytes.

4.3.2.3.6.2 Lista de fotos:

Cuando el progreso de subida finalice, se mostrará la foto en la Lista de fotos del usuario, como se muestra a continuación:



Imagen 285. Ejemplo de foto ya agregada en la lista de fotos del usuario en la aplicación web.

Como se puede notar en la imagen anterior, la foto que se ha agregado tiene un signo "X", que haciendo click sobre el signo se eliminará la foto.

Se puede agregar más fotos para que se almacenen en la aplicación web y se muestren en la lista de fotos, como se muestra a continuación:



Imagen 286. Ejemplo de 2 fotos agregadas en la lista de fotos del usuario en la aplicación web.

4.3.2.3.6.3 Cambio de foto de perfil:

Para cambiar la foto de perfil del usuario lo único que se debe hacer es elegir una foto desde la Lista de Fotos del usuario y arrastrar hacia la Foto de perfil actual, como se muestra a continuación:

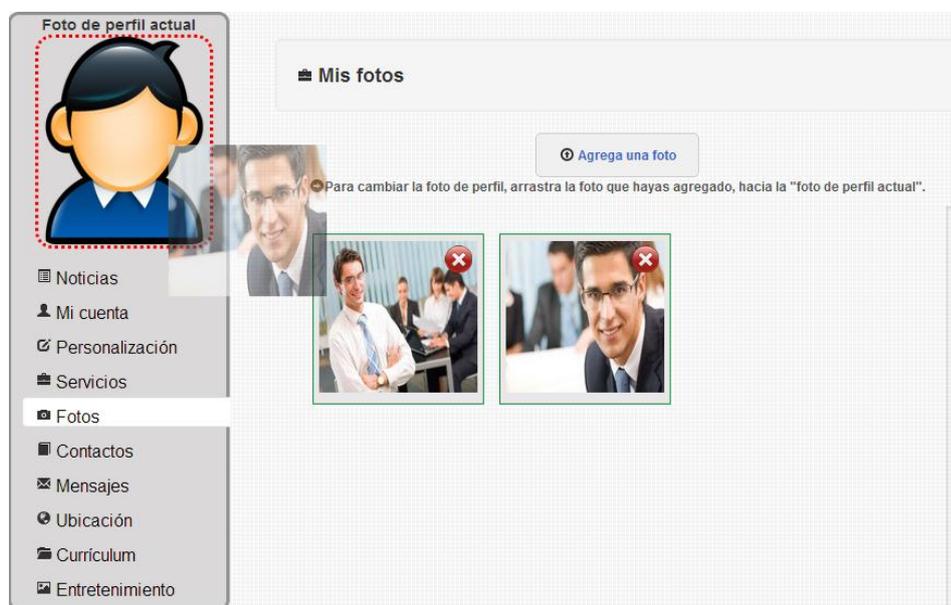


Imagen 287. Ejemplo de cómo cambiar la foto de perfil en la aplicación web.

Al momento de que se suelte la imagen sobre la Foto de perfil actual, cambiará la foto de perfil, como se muestra a continuación:

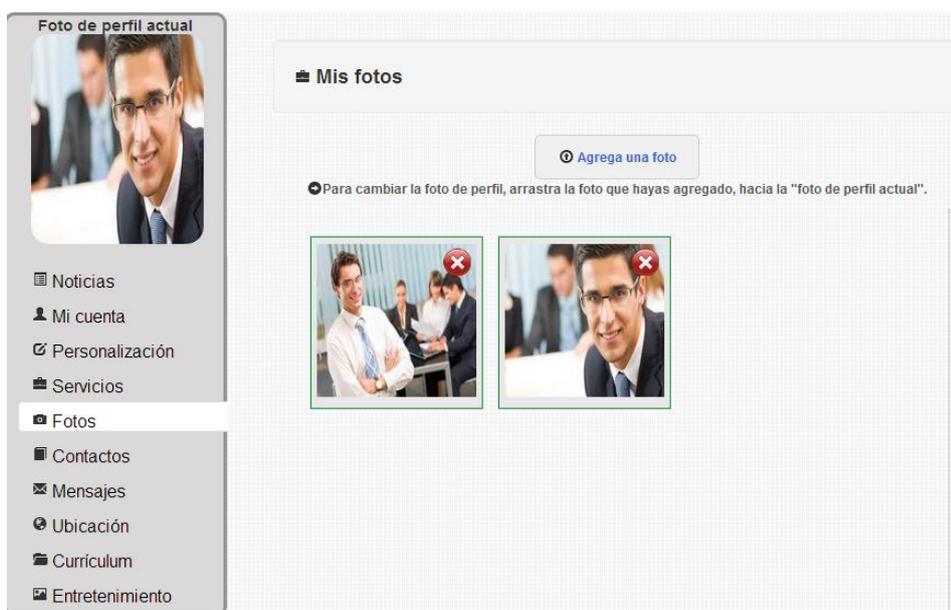


Imagen 288. Ejemplo de cómo se cambió una foto de perfil del usuario en la aplicación web.

4.3.2.3.7 Función Contactos

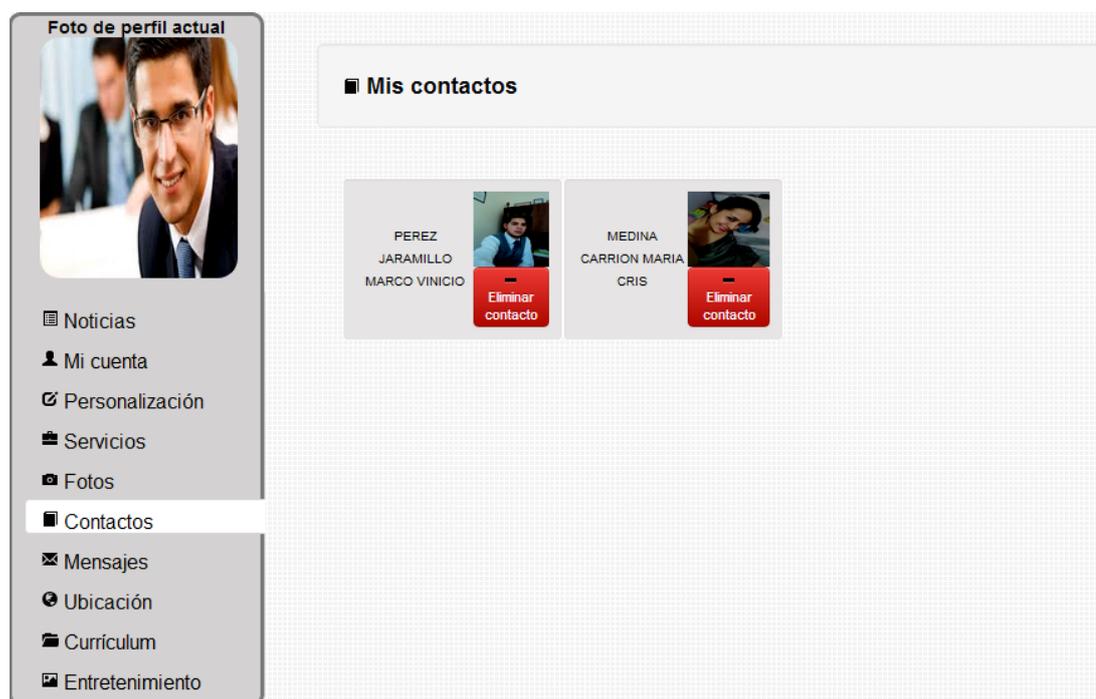


Imagen 289. Función Contactos de la aplicación web.

La función Contactos, contiene la lista de contactos del usuario, contactos que el usuario haya agregado para ponerse en comunicación con ellos.

Como se observa en la imagen anterior cada contacto posee un botón de color rojo, para poder eliminar al contacto si es que se desea.

Algo que cabe mencionar es que en cada contacto de la lista se puede revisar su perfil, simplemente con dar click en su foto.

4.3.2.3.8 Función Mensajes

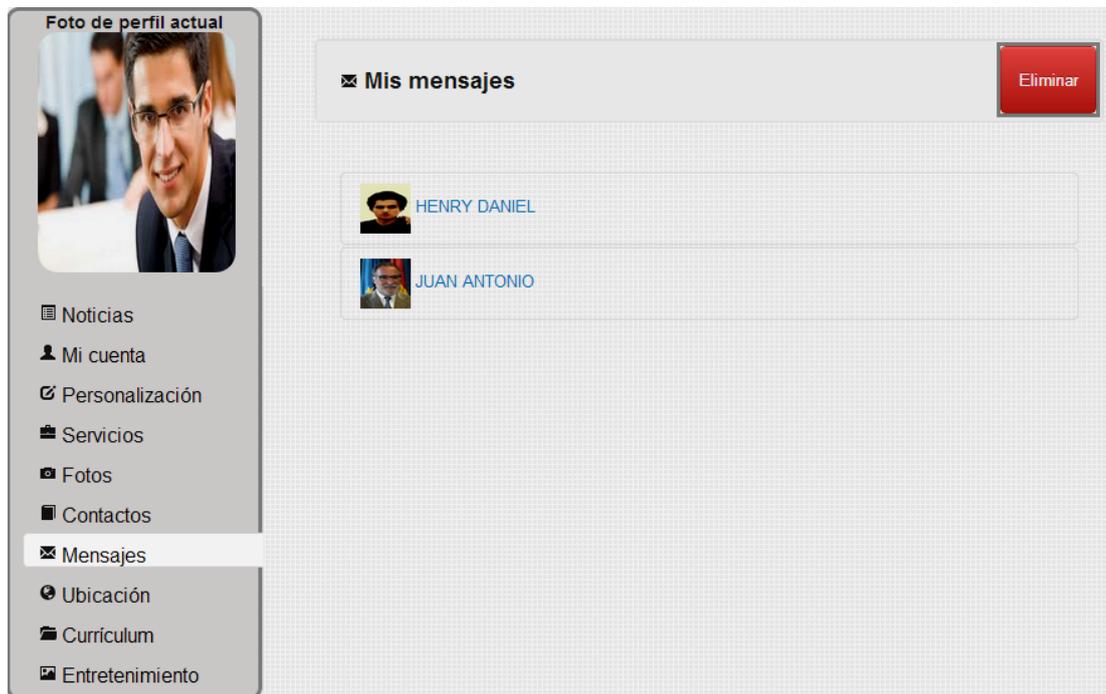


Imagen 290. Función Mensajes de la aplicación web.

La función de mensajes sirve para almacenar todos los mensajes que le lleguen al usuario. Se puede observar en la imagen que existen mensajes de dos usuarios (HENRY DANIEL y JUAN ANTONIO), para ver los mensajes se debe hacer click en cada usuario, por ejemplo para ver los mensajes de JUAN ANTONIO:

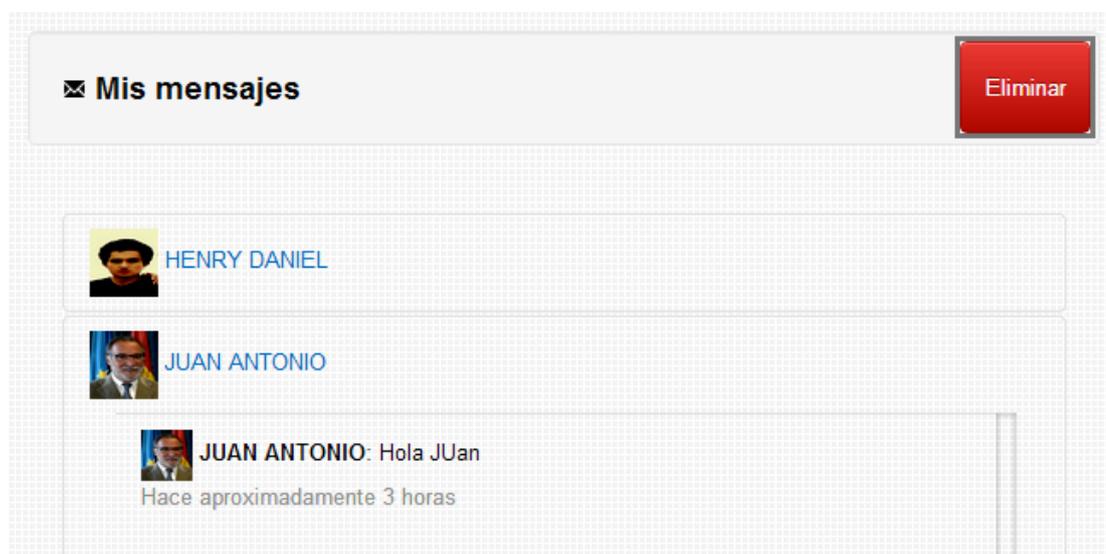


Imagen 291. Ejemplo de revisión de mensajes del usuario en la aplicación web.

Para eliminar toda la lista de mensajes se podrá llevar a cabo mediante el botón de color rojo “Eliminar”, que se encuentra en la parte superior derecha (Botones de contenido).

4.3.2.3.9 Función Ubicación

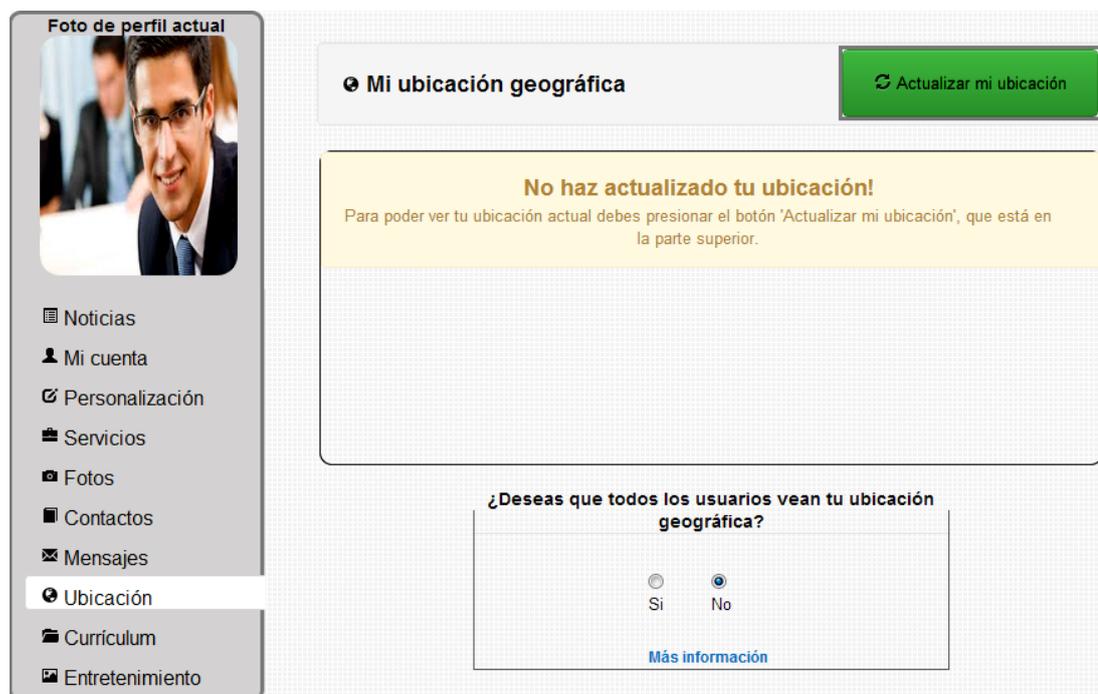


Imagen 292. Función Ubicación de la aplicación web.

Esta función permite mostrar la localización geográfica del usuario en un mapa, con tan solo dar click sobre el botón de color verde “Actualizar mi ubicación”, que se encuentra en la parte superior derecha (Botones de contenido).

El navegador web, por cuestiones de seguridad, preguntará al usuario si es que permite utilizar la ubicación geográfica, como se muestra a continuación:

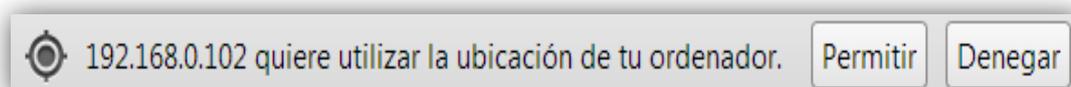


Imagen 293. Seguridad en el navegador al momento de Actualizar la ubicación del usuario en la aplicación web.

Si es que el usuario lo permite, pues entonces **PreServi**, localizará la ubicación del usuario en un mapa, como se muestra a continuación:



Imagen 294. Ejemplo al momento de actualizar la ubicación del usuario en la aplicación web.

Por defecto, la aplicación no muestra la ubicación geográfica a los demás usuarios, si es que el usuario desea que todos los usuarios vean su información geográfica en el perfil, simplemente deberá dar click en la opción "Si", como se muestra a continuación:

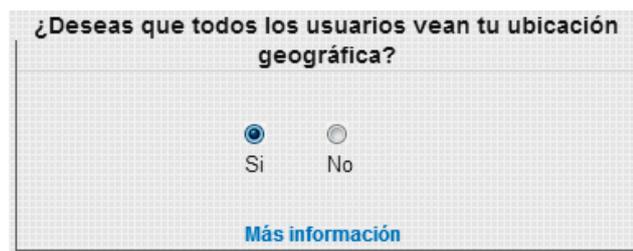


Imagen 295. Permiso para que todos los usuarios vean la ubicación geográfica del usuario en la aplicación web.

4.3.2.3.10 Función Currículum

Foto de perfil actual

Mi currículum

Datos personales Estudios Méritos Experiencia Adjuntos ➔ Secciones

Nombres: JUAN MANUEL Apellidos: Pérez CASTILLO

Edad: Ingresar edad País: Ecuador

Ciudad: Cuenca Dirección: Cdla. Totoracocha

Lugar de nacimiento: Ingresar lugar de nacimiento Documento de identificación: Ingresar número de identificación

Teléfono: 0998765432 E-mail: perezcastillo@gmail.com

➔ Contenido de secciones

Imagen 296. Función Currículum de la aplicación web.

Esta función como su nombre lo indica, es el currículum vitae del usuario y está dividido en 5 secciones principales: Datos Personales, Estudios, Méritos, Experiencia y Adjuntos.

Como se observa en la imagen anterior el contenido de cada sección se mostrará en la parte de Contenido de secciones.

4.3.2.3.10.1 Datos personales

Esta sección sirve para ingresar todos los datos personales del usuario para mostrar el currículum a todos los demás usuarios que puedan visitar el perfil de dicho usuario.

Existen 9 campos de datos para ingresar, incluidos los datos personales de la cuenta del usuario: Nombres, Apellidos, Edad, País, Ciudad, Dirección, Lugar de nacimiento, Documento de identificación y Teléfono.

El email también pertenece a los datos personales, pero no hay manera de modificarlo.

El ingreso de cada campo de datos, se lo puede realizar mediante un click en cada campo y enseguida se mostrará un botón para guardar lo que se ha ingresado, a continuación se muestra la sección Datos personales, ingresando los datos en sus campos:

Datos personales	Estudios	Méritos	Experiencia	Adjuntos	
Nombres:	<input type="text" value="JUAN MANUEL"/>		Apellidos:	<input type="text" value="Pérez CASTILLO"/>	
Edad:	<input type="text" value="28"/>	<input type="button" value="▲"/>	<input type="button" value="▼"/>	País:	<input type="text" value="Ecuador"/>
Ciudad:	<input type="text" value="Cuenca"/>		Dirección:	<input type="text" value="Cdla. Totoracoha"/>	
Lugar de nacimiento:	<input type="text" value="Quito"/>		Documento de identificación:	<input type="text" value="010675367-3"/>	
<input type="button" value="Guardar"/>					
Teléfono:	<input type="text" value="0998765432"/>		E-mail:	<input type="text" value="perezcastillo@gmail.com"/>	

Imagen 297. Sección Datos personales del currículum del usuario en la aplicación web.

4.3.2.3.10.2 Estudios

En esta sección de Estudios, el usuario podrá ingresar las instituciones donde ha realizado los estudios y además los títulos obtenidos.

Estudios que serán divididos en: Educación Primaria, Educación Secundaria, Educación Superior y Postgrados, como se puede observar a continuación:



Imagen 298. Sección Estudios del currículum del usuario en la aplicación web.

Como se observa, existen los 4 principales estudios, y para acceder a cada uno de ellos bastará con hacer click sobre cada estudio y se desplegarán opciones para ingresar la institución y los títulos obtenidos en aquel Estudio.

Educación Primaria:

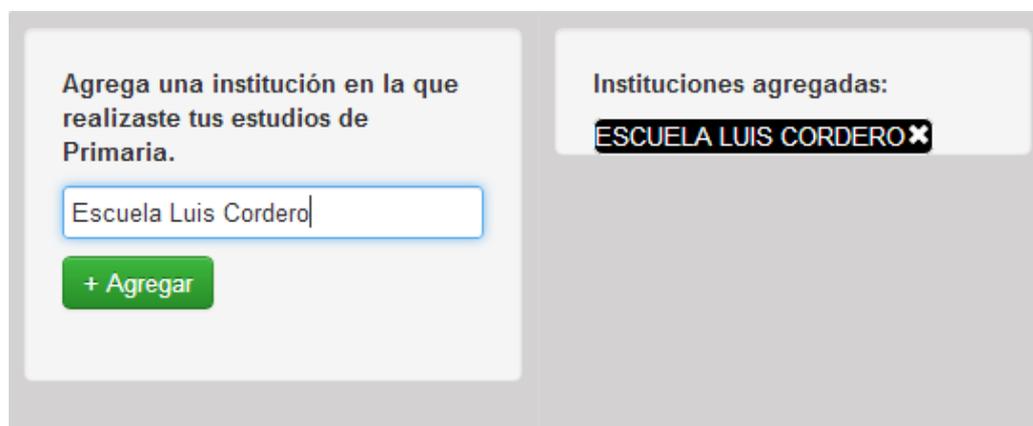


Imagen 299. Sección Educación primaria del currículum del usuario en la aplicación web.

Como se muestra en la imagen anterior, para agregar una institución de la Primaria solo hay que ingresar el nombre de la institución y dar un click sobre el botón “+Agregar” y se agregará a lista de instituciones.

Cuando se agregue las instituciones, se creará un signo “X” para poder eliminar la institución que se quiera, con solo dar click.

Educación Secundaria:

The image shows a web interface for adding secondary education information. It is divided into two main sections: institutions and titles. Each section has an input field and a green '+ Agregar' button. The right side of the interface displays the added items in a list, each with a black 'X' icon for removal.

Section	Input Field	Added Item
Instituciones	Colegio Técnico Salesiano	COLEGIO TÉCNICO SALESIANO X
Títulos	Informático	INFORMÁTICO X

Imagen 300. Sección Educación secundaria del currículum del usuario en la aplicación web.

Como se muestra en la imagen anterior, para agregar una institución de la Secundaria solo hay que ingresar el nombre de la institución y dar un click sobre el botón “+Agregar” y se agregará a lista de instituciones.

De igual manera para agregar un título obtenido en la Secundaria se debe ingresar el título y dar click en “+Agregar”.

Cuando se agregue las instituciones o los títulos obtenidos, se creará un signo “X” para poder eliminarlos cuando se quiera, con solo dar click.

Educación Superior:

The image displays a user interface for adding higher education information. It is divided into four quadrants. The top-left quadrant is titled 'Agrega una institución en la que realizaste tus estudios Superiores.' and contains a text input field with 'Universidad del Azuay' and a green '+ Agregar' button. The top-right quadrant is titled 'Instituciones agregadas:' and lists two items: 'UNIVERSIDAD DE CUENCA' and 'UNIVERSIDAD DEL AZUAY', each with a black 'X' icon for removal. The bottom-left quadrant is titled 'Agrega tus títulos obtenidos de tu educación Superior.' and contains a text input field with 'Ingeniero de Sistemas' and a green '+ Agregar' button. The bottom-right quadrant is titled 'Títulos agregados:' and lists one item: 'INGENIERO DE SISTEMAS' with a black 'X' icon for removal.

Imagen 301. Sección Educación superior del currículum del usuario en la aplicación web.

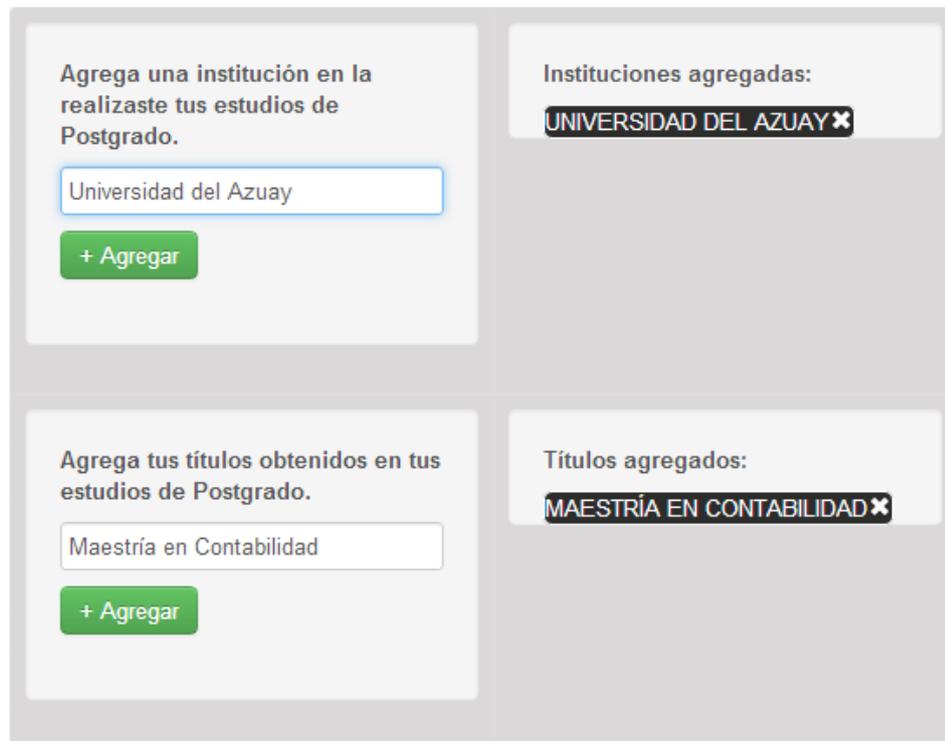
Como se muestra en la imagen anterior, para agregar una institución Superior solo hay que ingresar el nombre de la institución y dar un click sobre el botón “+Agregar” y se agregará a lista de instituciones.

De igual manera para agregar un título obtenido en la educación superior se debe ingresar el título y dar click en “+Agregar”.

Algo que es importante notar, como se observa en la imagen anterior, es que se puede ingresar más de una institución. Igualmente se puede agregar más de un título obtenido.

Cuando se agregue las instituciones o los títulos obtenidos, se creará un signo “X” para poder eliminarlos cuando se quiera, con solo dar click.

Postgrados:



The image displays a web application interface for managing postgraduate information. It is divided into four quadrants:

- Top-Left:** A section titled "Agrega una institución en la realizaste tus estudios de Postgrado." It contains a text input field with "Universidad del Azuay" and a green "+ Agregar" button.
- Top-Right:** A section titled "Instituciones agregadas:" showing a list with one item: "UNIVERSIDAD DEL AZUAY" with a close button (X).
- Bottom-Left:** A section titled "Agrega tus títulos obtenidos en tus estudios de Postgrado." It contains a text input field with "Maestría en Contabilidad" and a green "+ Agregar" button.
- Bottom-Right:** A section titled "Títulos agregados:" showing a list with one item: "MAESTRÍA EN CONTABILIDAD" with a close button (X).

Imagen 302. Sección Postgrados del currículum del usuario en la aplicación web.

Como se muestra en la imagen anterior, para agregar una institución de Postgrados solo hay que ingresar el nombre de la institución y dar un click sobre el botón “+Agregar” y se agregará a lista de instituciones.

De igual manera para agregar un título obtenido en la educación de Postgrados se debe ingresar el título y dar click en “+Agregar”.

Algo que es importante notar, como se observa en la imagen anterior, es que se puede ingresar más de una institución. Igualmente se puede agregar más de un título obtenido.

Cuando se agregue las instituciones o los títulos obtenidos, se creará un signo “X” para poder eliminarlos cuando se quiera, con solo dar click.

4.3.2.3.10.3 Méritos

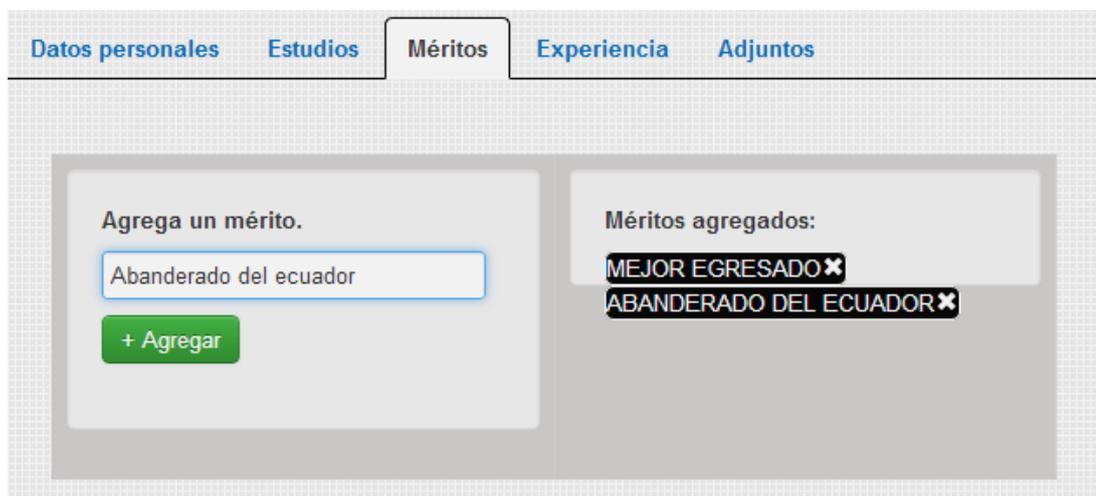


Imagen 303. Sección Méritos del currículum del usuario en la aplicación web.

En esta sección el usuario podrá agregar los méritos que ha conseguido, para llenarlo en su currículum. Con solo ingresar el nombre o título del mérito y dar click en el botón “+Agregar”, como se muestra en la imagen anterior.

Cuando se agregue los méritos obtenidos, se creará un signo “X” para poder eliminarlos cuando se quiera, con solo dar click.

4.3.2.3.10.4 Experiencia

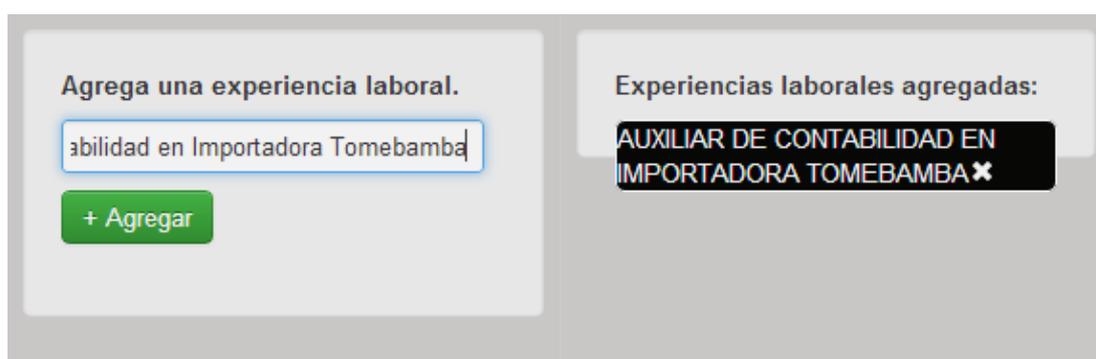


Imagen 304. Sección Experiencia del currículum del usuario en la aplicación web.

En esta sección el usuario podrá agregar las experiencias que haya tenido (laboral, habilidades, etc.), para llenarlo en su currículum. Con solo ingresar el nombre o título

de la experiencia y dar click en el botón “+Agregar”, como se muestra en la imagen anterior.

Cuando se agregue las experiencias, se creará un signo “X” para poder eliminarlos cuando se quiera, con solo dar click.

4.3.2.3.10.5 Adjuntos

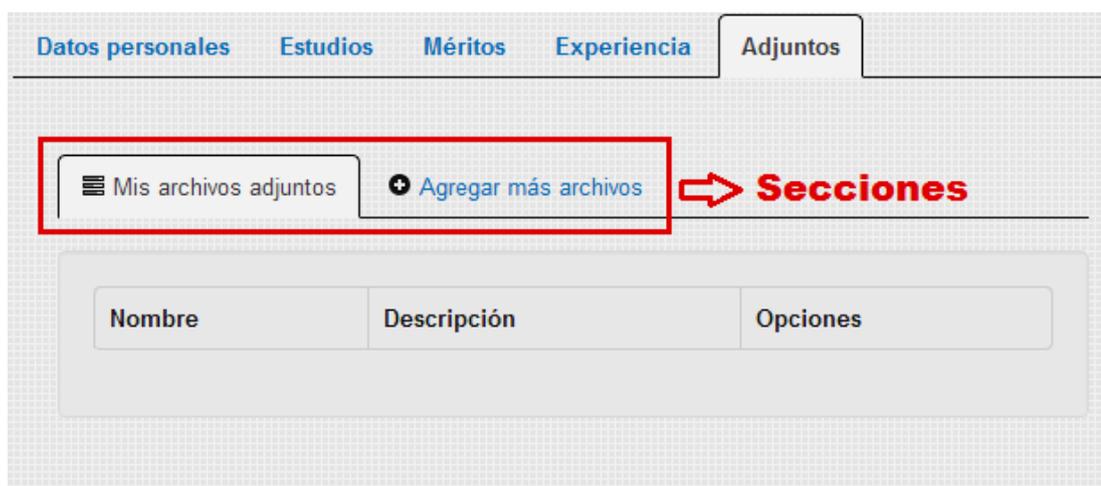


Imagen 305. Sección Adjuntos del currículum en la aplicación web.

Esta sección se utiliza para agregar archivos adjuntos al currículum, y como se puede observar está dividida en dos secciones más que son: Mis archivos adjuntos y Agregar más archivos.

Mis archivos adjuntos:



Imagen 306. Sección Mis archivos adjuntos del currículum en la aplicación web.

Esta sección, como se muestra la imagen anterior, mostrará una tabla con los archivos que el usuario tenga agregado a su currículum. Una tabla con el nombre del archivo, una descripción y 2 opciones: Eliminar archivo y Descargar el archivo, como se puede ver un ejemplo a continuación:

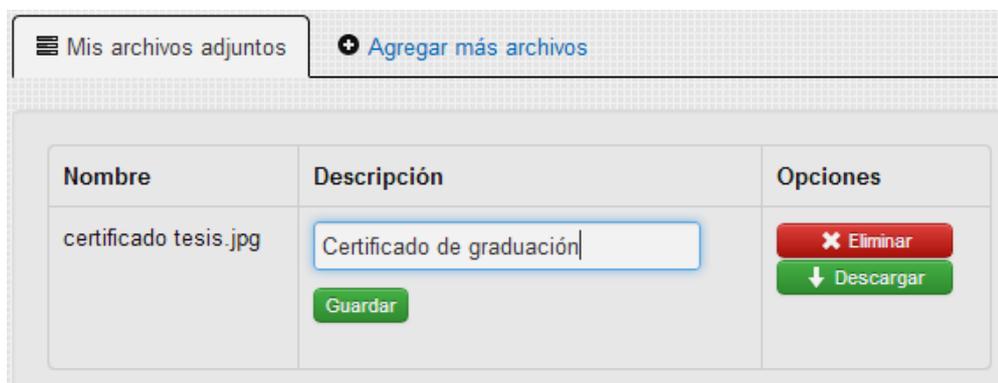


Imagen 307. Ejemplo de los archivos del usuario en el currículum de la aplicación web.

Como se muestra en la imagen anterior, existe un archivo que el usuario ha agregado a su currículum, y se puede ingresar una descripción haciendo click sobre el campo de datos y dando click en el botón “Guardar”.

También existen dos opciones (botones) para eliminar y descargar el archivo si se desea.

Agregar más archivos:



Imagen 308. Sección Agregar más archivos del currículum del usuario en la aplicación web.

En esta sección se pueden agregar los archivos adjuntos para el currículum. Se puede hacer de dos maneras:

- 1) Arrastrando el ó los archivos desde la ubicación original de la PC hasta la Zona de arrastre que se muestra en la imagen anterior.
- 2) Haciendo click en el botón “Elegir archivos” para elegir los archivos, y después haciendo click en el botón “Subir”.

En la siguiente imagen se muestra cuando los archivos han sido elegidos para ser subidos:

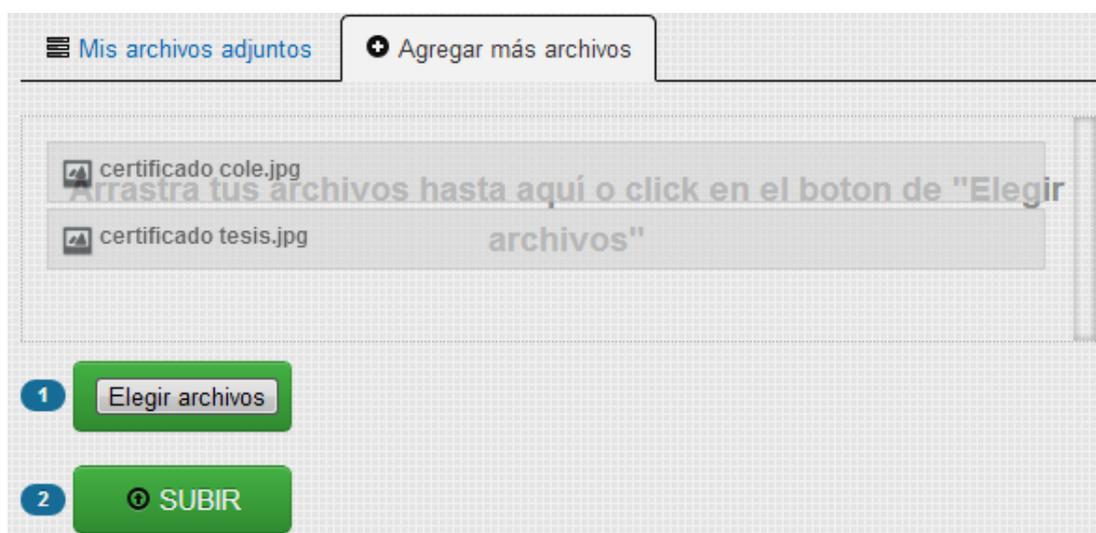


Imagen 309. Ejemplo de 2 archivos listos para agregar al currículum del usuario en la aplicación web.

Se podrán subir archivos, máximo de 2 Mega Bytes.

4.3.2.3.11 Función Entretenimiento

Esta función contiene un juego simple de entretenimiento para los usuarios, se trata de un juego denominado: “Juego de la víbora”, que consiste en una pequeña víbora o serpiente que no debe chocar con el filo del área de juego, y debe ir cogiendo puntos (comida) blancos para ir creciendo y aumentando los puntos.

A continuación se muestra una imagen del juego:

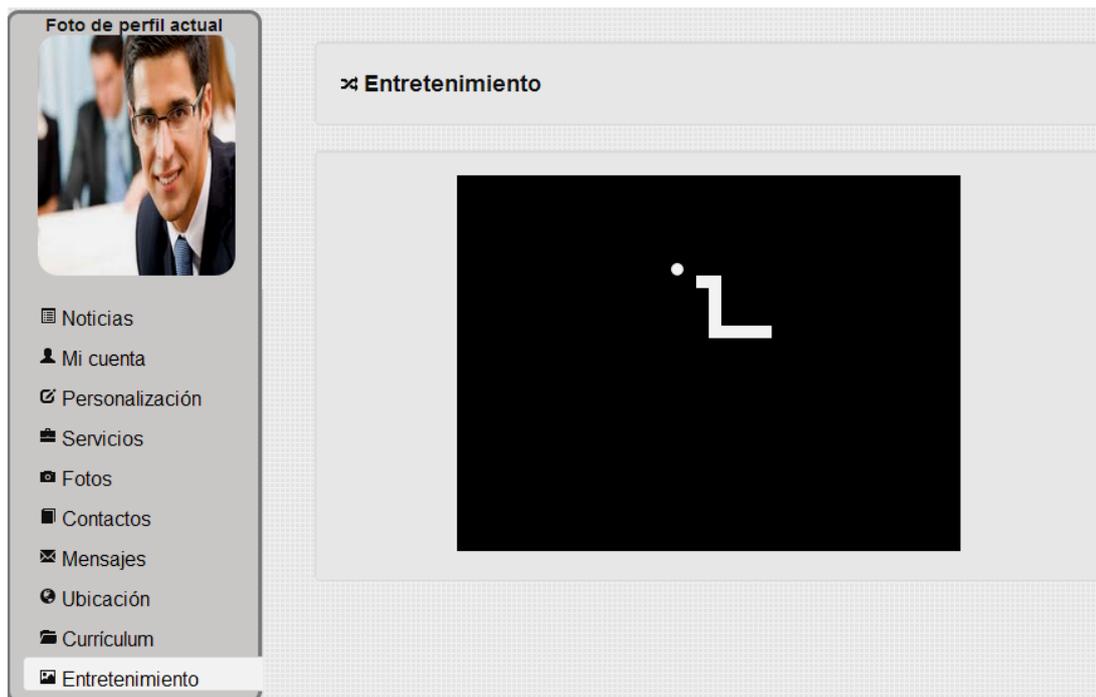


Imagen 310. Función Entrenimiento del usuario en la aplicación web.

Para iniciar ó pausar el juego se debe presionar la tecla “ESC” y para dirigir a la víbora, se utiliza las teclas de dirección del teclado (flecha arriba, abajo, derecha, izquierda).

4.3.2.4 Barra de información

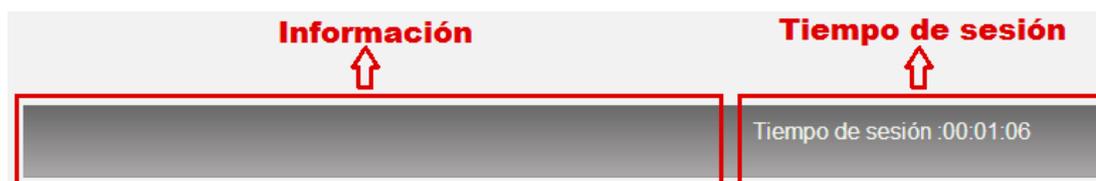


Imagen 311. Barra de información de la aplicación web.

Esta barra se encuentra en la parte inferior de la aplicación web, y su función es mostrar al usuario información sobre las interacciones de él con la aplicación. Se divide en dos partes importantes: Información y Tiempo de sesión.

Información:

En esta parte de la barra se informará las interacciones del usuario con la aplicación, además de notificaciones de actividades de los contactos del usuario como: Si el

contacto se ha conectado o se ha desconectado, si tiene algún mensaje por leer, o si tiene nuevas noticias por leer.

Por ejemplo si el usuario editara su número de teléfono de sus Datos personales, en la barra de información se mostrará lo siguiente:



Imagen 312. Ejemplo de la barra de información cuando el usuario edite su teléfono en la aplicación web.

Otro ejemplo puede ser al recibir notificaciones de algún contacto por ejemplo:

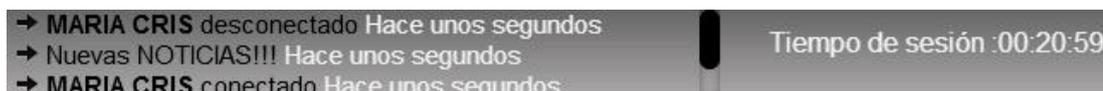


Imagen 313. Ejemplo de la barra de información cuando el usuario reciba notificaciones en la aplicación web.

4.3.2.5 Herramienta de chat

Esta herramienta es muy importante en **PreSrvi** ya que permite a los usuarios comunicarse con sus contactos. A continuación se puede ver en la imagen la herramienta de chat, ya con 2 contactos agregados:

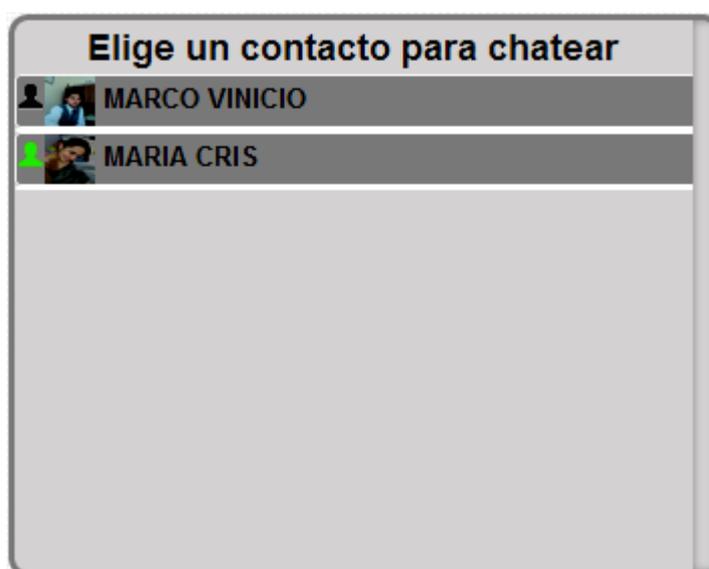


Imagen 314. Herramienta de chat del usuario en la aplicación web

Como se observa en la imagen anterior, existen 2 contactos en la lista del usuario, con los que se intentará “chatear” online.

Cabe diferenciar que un contacto no conectado tendrá un icono de usuario de color gris, como por ejemplo:



Imagen 315. Ejemplo de un contacto desconectado en la aplicación web.

Por otro lado un contacto conectado (online) tendrá un icono de color verde, como por ejemplo:



Imagen 316. Ejemplo de un contacto conectado en la aplicación web.

Para poder “chatear” con los contactos se debe tener presente lo siguiente:

- 1) El contacto al que se le desea enviar un mensaje de chat debe estar conectado (online), si no lo está el mensaje le llegará solo a su carpeta de mensajes (Función Mensajes).
- 2) El usuario que intenta enviar el mensaje de chat (chatear), debe estar en la lista de contactos del usuario al que se le envía el mensaje, caso contrario le llegará solo a su carpeta de mensajes (Función Mensajes)

Para enviar un mensaje de chat, se debe elegir el contacto y dar click sobre él, y se desplegará un cuadro para poder escribir un mensaje, como se muestra a continuación:

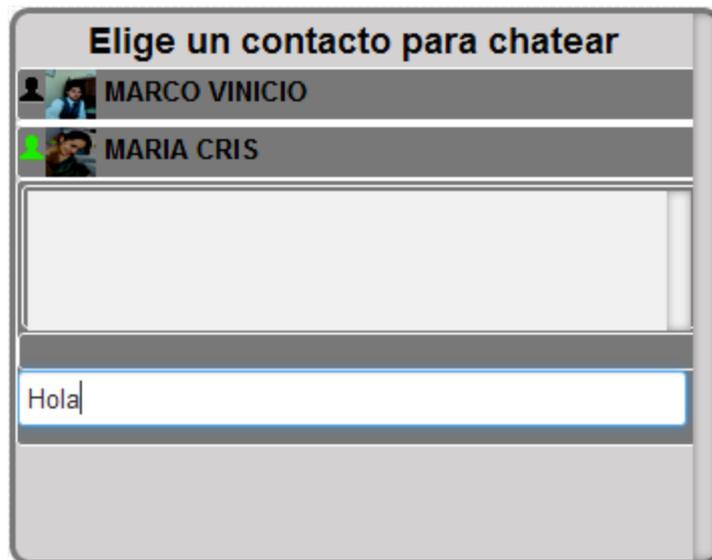


Imagen 317. Ejemplo para elegir un contacto y enviarle un mensaje de chat en la aplicación web.

Ahora se puede escribir y con para enviar el mensaje se debe presionar la tecla “ENTER” y se visualizará el mensaje que se envíe de esta manera:



Imagen 318. Ejemplo de envío de un mensaje de chat a un contacto en la aplicación web.

Se observa en la imagen anterior que se le ha enviado un mensaje de chat a un usuario llamado “MARIA CRIS”, pues entonces en la página de “MARIA CRIS” se le avisará que tiene un mensaje de la siguiente forma:

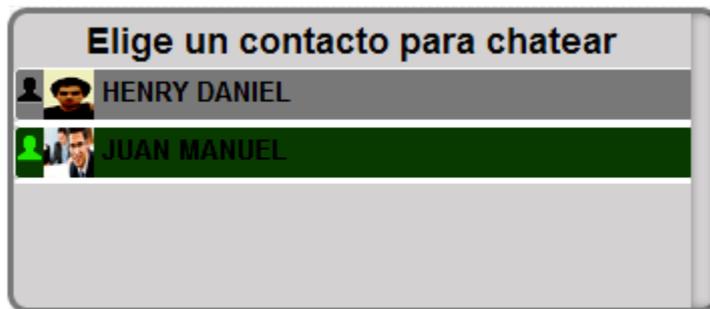


Imagen 319. Ejemplo de aviso cuando al usuario le llegue un mensaje de chat en la aplicación web.

Como se observa “JUAN MANUEL” es quien le ha enviado el mensaje de chat a “MARIA CRIS”, y solo deberá hacer click sobre el contacto para leer el o los mensajes que “JUAN MANUEL” le ha enviado, como se muestra a continuación:



Imagen 320. Ejemplo de cómo leer un mensaje que le ha llegado al usuario en la aplicación web.

Si el usuario al que le llega el mensaje tiene el volumen de su PC encendido, pues podrá escuchar un sonido de aviso y si tiene activado la opción de mostrar Notificaciones de escritorio, podrá observarse en su PC lo siguiente:

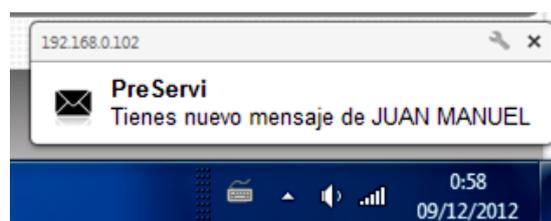


Imagen 321. Ejemplo de notificación de escritorio en la aplicación web.

4.3.3 Perfil visible a los demás usuarios

Este es el perfil público de los usuarios, ya que cuando un usuario encuentre a una persona y revise su perfil esto es lo que observará:



Imagen 322. Perfil visible a los demás usuarios en la aplicación web.

En la imagen anterior se muestra la página de perfil del usuario, esta página será vista cuando se encuentre a algún usuario, con el Buscador de usuarios o personas.

El perfil de usuarios contiene toda la información de cada uno de los usuarios como: Datos personales, Servicios que ofrece y busca, Ubicación geográfica, Fotos y el Currículum.

Como se observa en la imagen anterior, existen 4 secciones o partes principales del perfil como: Botón agregar/eliminar contacto, Menú de navegación, Foto de perfil y el Contenido.

4.3.3.1 Botón agregar/eliminar contacto:

La función de este botón es agregar al usuario del que se está revisando el perfil a la lista de contactos ó a su vez si el usuario ya pertenece a la lista de contactos eliminar al usuario.

Si el usuario ya pertenece a la lista de contactos se mostrará el siguiente botón:



Imagen 323. Botón eliminar contacto en la aplicación web.

Si el usuario no pertenece aún a la lista de contactos se mostrará el siguiente botón:



Imagen 324. Botón agregar contacto en la aplicación web.

4.3.3.2 Foto de perfil:

En esta sección se mostrará la foto de perfil del usuario:



Imagen 325. Foto de perfil en el perfil del usuario en la aplicación web.

4.3.3.3 Contenido:

Esta sección será el contenido de información referente al usuario, por ejemplo el contenido del perfil, de la ubicación, de las fotos o del currículum en sí.

4.3.3.4 Menú de navegación:



El menú que se muestra en la imagen anterior, se utiliza para navegar entre la información del usuario, revisando el Perfil, la Ubicación, las Fotos y el Currículum, todo esto se mostrará en la sección Contenido.

Perfil:



Imagen 326. Perfil del usuario, sección Perfil de la aplicación web.

Esta sección mostrará los Datos personales y los servicios que busca y ofrece el usuario.

Ubicación:

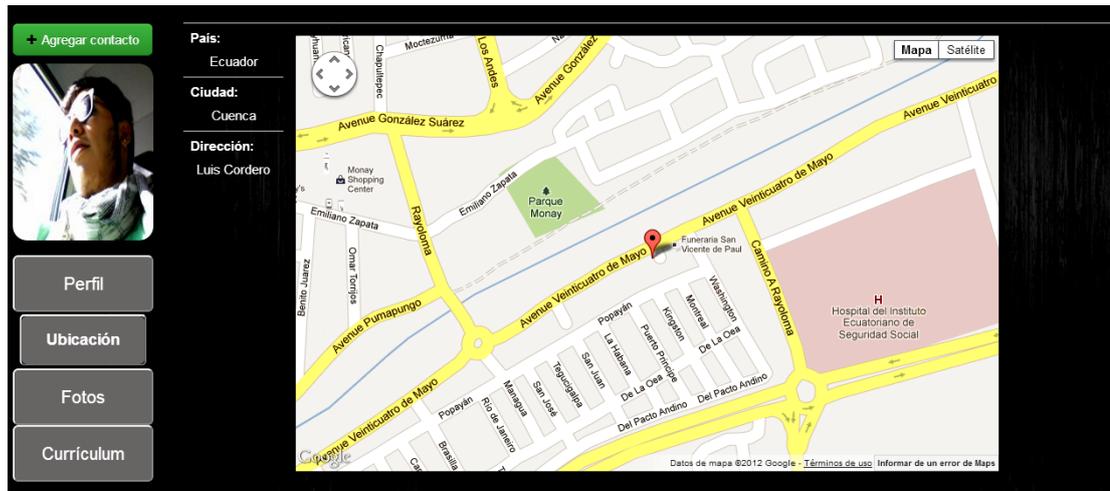


Imagen 327. Perfil del usuario, sección Ubicación de la aplicación web.

Esta sección mostrará la Ubicación del usuario: Ubicación geográfica, país, ciudad y dirección.

Fotos:

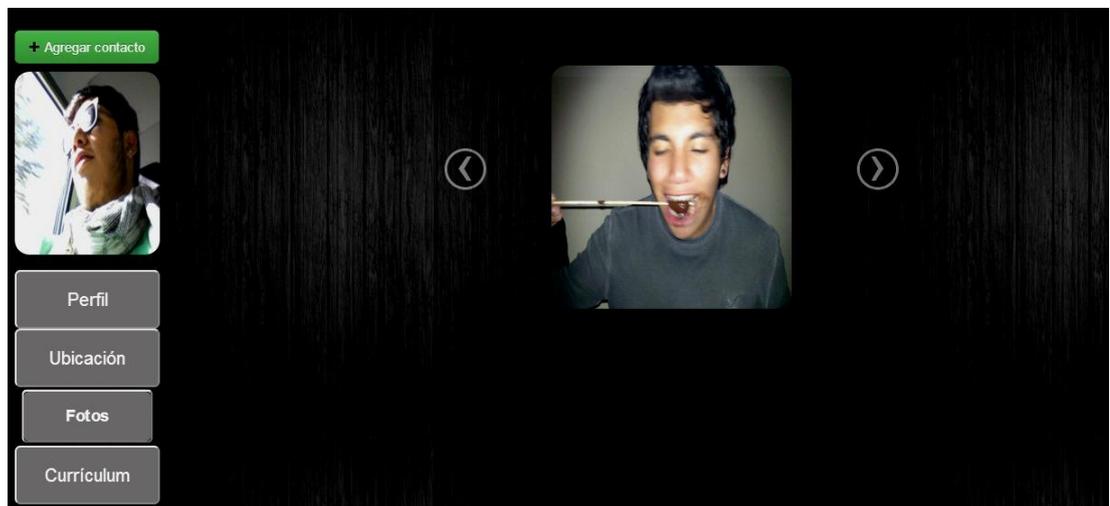


Imagen 328. Perfil del usuario, sección Fotos de la aplicación web.

Esta sección mostrará las fotos del usuario.

Currículum:



Imagen 329. Perfil del usuario, sección Currículum de la aplicación web.

Esta sección mostrará el currículum del usuario, como se observa en la imagen anterior existe un Menú del currículum, para revisar cada parte del currículum como: Datos personales, Estudios, Méritos, Experiencia laboral y Adjuntos, a continuación se muestra el menú:



Imagen 330. Menú del currículum dl usuario en la aplicación web.

Se puede acceder a cada parte del currículum, haciendo click sobre ésta en el menú.

4.4 Manual para administrador

4.4.1 Página de Inicio

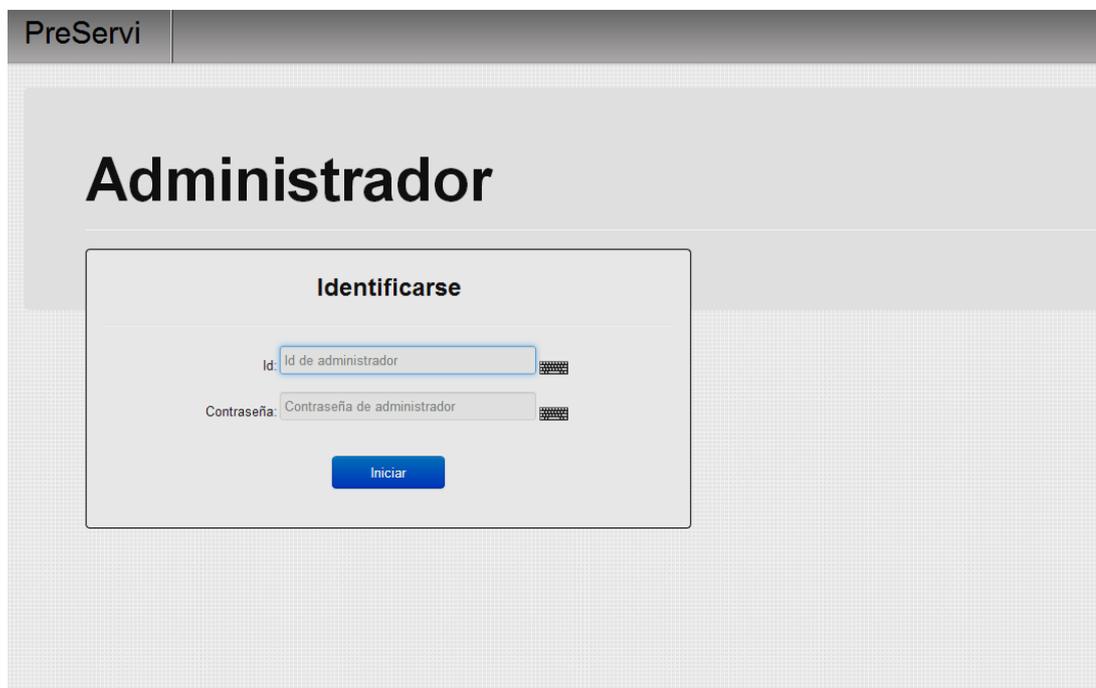


Imagen 331. Página de inicio del administrador de la aplicación web.

Lo que se muestra en la imagen anterior es la página de inicio del administrador, donde podrá identificarse mediante un ID y una Contraseña.

4.4.1.1 Identificarse

Para poder identificarse como administrador se deberá ingresar los siguientes datos:

ID: admin

CONTRASEÑA: adminpreservi

A continuación se muestra la zona de identificación del administrador, donde deberá ingresar los datos antes mencionados:



Imagen 332. Zona de identificación del administrador de la aplicación web.

Para ingresar esos datos en la página de inicio del administrador, se deberá hacer mediante un teclado virtual, ya que se ha desactivado el teclado por razones de seguridad.

En la imagen anterior se muestra 2 botones que al hacer click sobre ellos aparecerá un teclado virtual para ingresar el ID y Contraseña del administrador, como se muestra a continuación:



Imagen 333. Identificación del administrador con el teclado virtual en la aplicación web.

Cuando se haya ingresado los datos del ID y Contraseña del administrador mediante el teclado virtual, se podrá iniciar a la página principal del administrador, haciendo click en el botón “Iniciar”.

4.4.2 Página principal

Al ingresar los datos correctos de identificación del administrador y al hacer click en el botón “Iniciar”, aparecerá la página principal del administrador, para poder sacar reportes de las actividades de los usuarios, a continuación se muestra la página principal del administrador:



Imagen 334. Página principal del administrador de la aplicación web.

4.4.2.1 Tabla de reportes generada

En esta zona de la página se mostrará una tabla de los reportes generados por el usuario, una tabla que se podrá imprimir.

4.4.2.2 Tipos de Reportes

El administrador podrá generar reportes de las actividades que hayan realizado los usuarios de diferentes tipos como: Reportes por usuario, por fecha específica, por mes y año específico, y un reporte total. A continuación se muestra la sección donde elegir el tipo de reporte que se desee generar:



Imagen 335. Sección para escoger el tipo de reporte que generará el administrador de la aplicación web.

Al hacer click en cada uno de los tipos de reporte se accederá a su contenido y así generar el reporte deseado.

4.4.2.2.1 Reporte por usuario



Imagen 336. Reporte por usuario para el administrador en la aplicación web.

Este reporte generará las actividades de un usuario en específico y se podrá filtrar para generar:

- Todas las actividades del usuario
- Actividades en una fecha específica
- Actividades en un mes y año específico

Pasos:

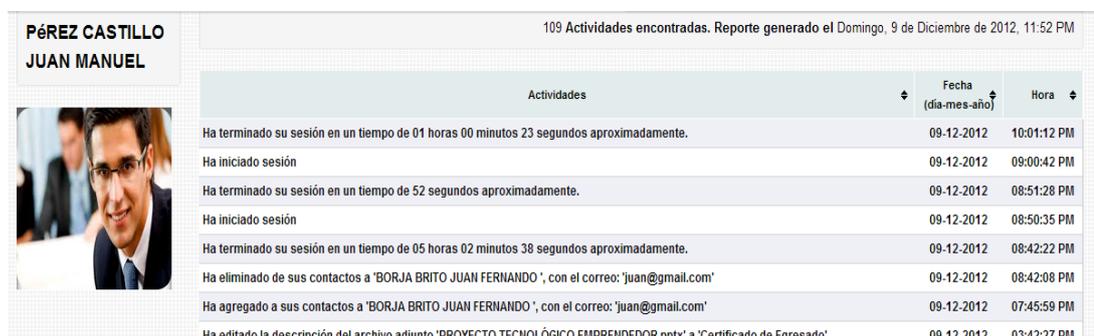
- 1) Ingresar el correo electrónico del usuario
- 2) Escoger el filtro para generar el reporte

Si se escoge el filtro de Reporte en fecha específica, se debe elegir la fecha.

Si se escoge el filtro Reporte en mes y año específico, se debe elegir el mes y el año

- 3) Dar click en el botón “Generar reporte”

Cuando se cumplan esos pasos el reporte comenzará a generarse y se mostrará algo así en la tabla de reportes generada:



Actividades	Fecha (día-mes-año)	Hora
Ha terminado su sesión en un tiempo de 01 horas 00 minutos 23 segundos aproximadamente.	09-12-2012	10:01:12 PM
Ha iniciado sesión	09-12-2012	09:00:42 PM
Ha terminado su sesión en un tiempo de 52 segundos aproximadamente.	09-12-2012	08:51:28 PM
Ha iniciado sesión	09-12-2012	08:50:35 PM
Ha terminado su sesión en un tiempo de 05 horas 02 minutos 38 segundos aproximadamente.	09-12-2012	08:42:22 PM
Ha eliminado de sus contactos a 'BORJA BRITO JUAN FERNANDO', con el correo: 'juan@gmail.com'	09-12-2012	08:42:08 PM
Ha agregado a sus contactos a 'BORJA BRITO JUAN FERNANDO', con el correo: 'juan@gmail.com'	09-12-2012	07:45:59 PM
Ha editado la descripción del archivo adjunto 'PROYECTO TECNOLÓGICO EMPRENDEDOR.pptx' a 'Certificado de Egresado'	09-12-2012	03:42:27 PM

Imagen 337. Ejemplo de reporte generado por el administrador de las actividades de un usuario en específico en la aplicación web.

Al momento que se genere el reporte, aparecerá una tabla con la descripción de cada actividad que el usuario ha realizado, con la fecha y hora.

4.4.2.2 Reporte por fecha específica

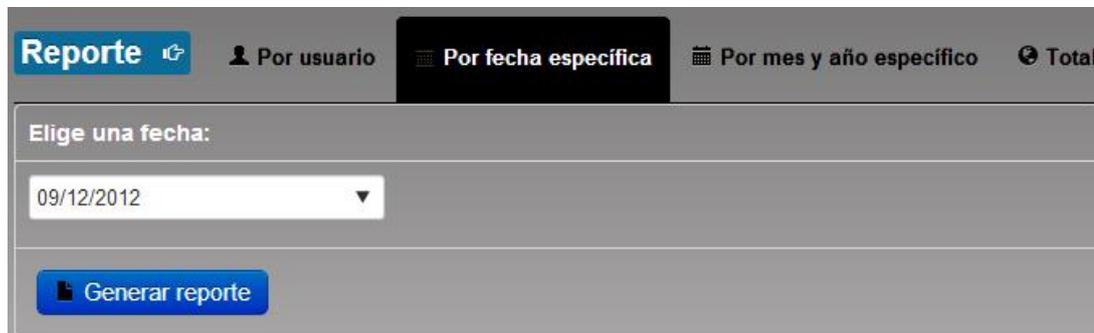


Imagen 338. Reporte por fecha específica de usuarios para el administrador en la aplicación web.

Este reporte generará las actividades de todos los usuarios que hayan realizado actividades en una fecha específica.

Pasos:

- 1) Elegir una fecha específica.
- 2) Dar click en el botón “Generar reporte”

Cuando se cumplan esos pasos el reporte comenzará a generarse y se mostrará algo así en la tabla de reportes generada:

52 Actividades encontradas. Reporte generado el Lunes, 10 de Diciembre de 2012. 00:26 AM				
Usuario	Correo	Actividades	Fecha (dia-mes-año)	Hora
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha terminado su sesión en un tiempo de 08 minutos 55 segundos aproximadamente.	09-12-2012	09:00:31 PM
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha editado la descripción del archivo adjunto 'Curriculum.zip' a 'Titulo de grado'	09-12-2012	09:00:28 PM
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha editado la descripción del archivo adjunto 'tgcmllog.txt' a 'Certificado de Egresado'	09-12-2012	09:00:16 PM
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha agregado un archivo con el nombre: 'Curriculum.zip' como adjunto a su curriculum	09-12-2012	08:59:57 PM

Imagen 339. Ejemplo de reporte generado por el administrador de las actividades de usuarios en una fecha específica en la aplicación web.

Al momento que se genere el reporte, aparecerá una tabla con el nombre de cada usuario, su correo electrónico, descripción de cada actividad que el usuario ha realizado, con la fecha y hora.

4.4.2.3 Reporte por mes y año específico

The screenshot shows a web interface for generating reports. At the top, there is a navigation bar with the title 'Reporte' and four filter options: 'Por usuario', 'Por fecha específica', 'Por mes y año específico' (which is selected and highlighted in black), and 'Total'. Below the navigation bar, there are two dropdown menus: 'Elige un mes:' with 'Diciembre' selected, and 'Elige un año:' with '2012' selected. At the bottom of the form is a blue button labeled 'Generar reporte'.

Imagen 340. Reporte por mes y año específico de usuarios para el administrador en la aplicación web.

Este reporte generará las actividades de todos los usuarios que hayan realizado actividades en un mes y año específico.

Pasos:

- 1) Elegir un mes específico.
- 2) Elegir un año específico.
- 3) Dar click en el botón “Generar reporte”

Cuando se cumplan esos pasos el reporte comenzará a generarse y se mostrará algo así en la tabla de reportes generada:

52 Actividades encontradas. Reporte generado el Lunes, 10 de Diciembre de 2012, 00:26 AM				
Usuario	Correo	Actividades	Fecha (día-mes-año)	Hora
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha terminado su sesión en un tiempo de 08 minutos 55 segundos aproximadamente.	09-12-2012	09:00:31 PM
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha editado la descripción del archivo adjunto 'Curriculum.zip' a 'Titulo de grado'	09-12-2012	09:00:28 PM
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha editado la descripción del archivo adjunto 'tgcmllog.txt' a 'Certificado de Egresado'	09-12-2012	09:00:16 PM
BORJA BRITO JUAN FERNANDO	juan@gmail.com	Ha agregado un archivo con el nombre: 'Curriculum.zip' como adjunto a su curriculum	09-12-2012	08:59:57 PM

Imagen 341. Ejemplo de reporte generado por el administrador de las actividades de usuarios en un mes y año específico en la aplicación web.

Al momento que se genere el reporte, aparecerá una tabla con el nombre de cada usuario, su correo electrónico, descripción de cada actividad que el usuario ha realizado, con la fecha y hora.

4.4.2.2.4 Reporte total

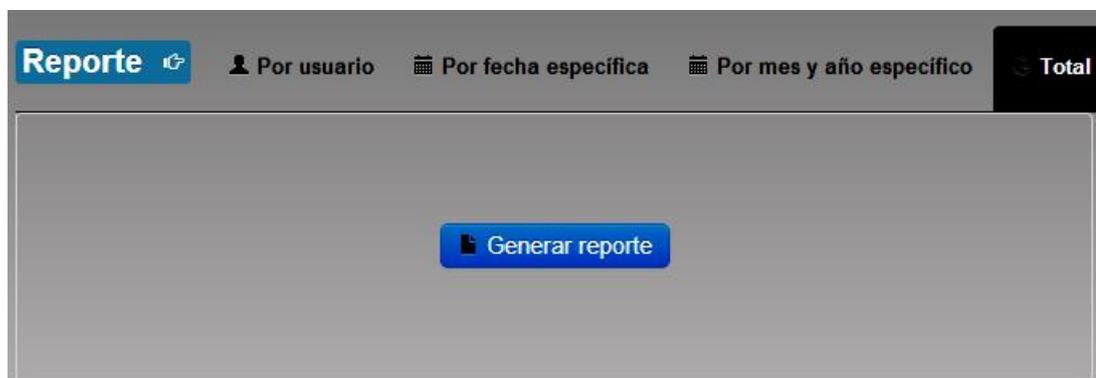


Imagen 342. Reporte total de usuarios para el administrador en la aplicación web.

Este reporte generará todas actividades de todos los usuarios registrados en **PreServi**.

Pasos:

- 1) Dar click en el botón “Generar reporte”

Cuando se cumpla ese paso el reporte comenzará a generarse y se mostrará algo así en la tabla de reportes generada:

523 Actividades encontradas. Reporte generado el Lunes, 10 de Diciembre de 2012, 00:56 AM

Usuario	Correo	Actividades	Fecha (día-mes-año)	Hora
ANDRADE CASTRO ADRIAN RENATO	andrade@gmail.com	Ha eliminado su cuenta de PreServi	28-10-2012	06:39:43 PM
ANDRADE CASTRO ADRIAN RENATO	andrade@gmail.com	Ha iniciado sesión	28-10-2012	06:39:06 PM
ANDRADE CASTRO ADRIAN RENATO	andrade@gmail.com	Se ha unido a PreServi	28-10-2012	06:15:15 PM
CABRERA RIOS DIEGO FERNANDO	cabra@gmail.com	Ha eliminado su cuenta de PreServi	28-10-2012	06:26:11 PM

Imagen 343. Ejemplo de reporte generado por el administrador de todas las actividades de usuarios en la aplicación web.

Al momento que se genere el reporte, aparecerá una tabla con el nombre de cada usuario, su correo electrónico, descripción de cada actividad que el usuario ha realizado, con la fecha y hora.

Cuando se genere cualquier tipo de reporte, aparecerá un botón de Imprimir, para poder imprimir todo el reporte.

4.4.2.3 Cantidad de usuarios registrados

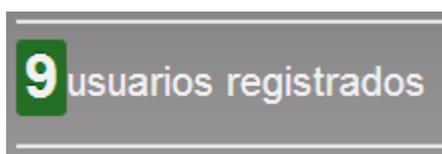


Imagen 344. Sección Cantidad de usuarios registrados en el administrador de la aplicación web.

En esta sección el administrador podrá observar la cantidad exacta de usuarios registrados en **PreServi**.

4.4.2.4 Salir del administrador



Imagen 345. Botón para salir del administrador en la aplicación web.

Esta sección simplemente cuenta con un botón para poder salir del administrador.

CONCLUSIONES

Todos los objetivos planteados en el diseño de esta tesis fueron cumplidos.

Durante el proceso de desarrollo de la aplicación web se pudo apreciar que utilizando los nuevos elementos de HTML5 se puede ahorrar tiempo al momento de crear páginas web, ya que lo que antes se realizaba con Javascript en varias líneas de código ahora se lo puede realizar en una sola, como por ejemplo la etiqueta “input” con la propiedad “type” y el valor “date” que despliega un calendario para elegir la fecha.

Las nuevas etiquetas section, article, aside, nav, footer y header permiten organizar de una mejor manera el contenido de una página web.

Una de las grandes ventajas de la base de datos MongoDB es que mediante el código de programación se la puede modificar de forma dinámica de acuerdo a como se ejecuta la aplicación de esta forma se adecua a las exigencias del momento.

En aplicaciones donde se manejan gran cantidad de datos con grandes volúmenes de escritura MongoDB es una alternativa a tomar en cuenta.

No todos los navegadores en la actualidad soportan HTML5 al cien por ciento, por eso hay que tener cuidado al momento de desarrollar en el mismo, para que pueda funcionar el aplicativo en la mayoría de los navegadores.

RECOMENDACIONES

Utilizar librerías en Javascript (HTML5.js) que ayuden a mejorar la compatibilidad en los navegadores, ya que actualmente todos los navegadores no soportan completamente el estándar HTML5.

Considerar la base de datos NoSQL en el momento de tomar una decisión en elegir una base de datos para un proyecto.

Revisar constantemente la documentación oficial de HTML5 para ir mejorando el manual creado en esta tesis.

BIBLIOGRAFÍA

REFERENCIAS ELECTRÓNICAS:

10 Gen – The MongoDB company. 2012. Introducción a NoSQL y MongoDB. [Disponible en: <http://www.10gen.com/events/webinar/Introduccion-al-NoSQL-y-MongoDB>]. [Consulta: 8 de Noviembre de 2012].

Adictos al Trabajo. 2012. HTML5: Almacenamiento Local. [Disponible en: <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=html5AlmacenamientoLocal>]. [Consulta: 17 de Octubre de 2012].

Bootstrap. [s.a.]. Bootstrap. [Disponible en: <http://twitter.github.com/bootstrap/>]. [Consulta: 4 de Julio de 2012].

GreyWyvern.com. 2012. Virtual Keyboard Interface. [Disponible en: <http://www.greywyvern.com/code/javascript/keyboard>]. [Consulta: 26 de Noviembre de 2012].

HTML5 Doctor. 2010. Introducing Web SQL Databases. [Disponible en: <http://html5doctor.com/introducing-web-sql-databases/>]. [Consulta: 15 de Octubre de 2012].

HTML5 Fácil. 2011. ¿Cómo utilizar web workers de JavaScript?. [Disponible en: <http://html5facil.com/tutoriales/como-utilizar-web-workers-de-javascript>]. [Consulta: 29 de Octubre de 2012].

HTML5Pattern. [s.a.]. HTML5Pattern. [Disponible en: <http://html5pattern.com/>]. [Consulta: 26 de Septiembre de 2012].

HTML5 Rocks. 2011. Creación de formas fabulosas con HTML5. [Disponible en: <http://www.html5rocks.com/es/tutorials/forms/html5forms/>]. [Consulta: 19 de Julio de 2012].

HTML5 Rocks. [s.a.]. ¿Por qué HTML5 es la estrella ahora?. [Disponible en: <http://www.html5rocks.com/es/why>]. [Consulta: 25 de Junio de 2012].

HTML5 Tutoriales. 2012. Atributos de Eventos HTML5. [Disponible en: <http://html5tutoriales.com/atributos-de-eventos-html5.html>]. [Consulta: 26 de Noviembre de 2012].

Inmensia. 2011. Web Notifications. [Disponible en: http://www.inmensia.com/blog/20110108/web_notifications.html]. [Consulta: 29 de Octubre de 2012].

JQuery. 2012. JQuery. [Disponible en: <http://jquery.com/>]. [Consulta: 6 de Julio de 2012].

JQuery User Interface. 2012. JQuery UI. [Disponible en: <http://jqueryui.com/>]. [Consulta: 16 de Octubre de 2012].

JQuery. 2008. Plugins Print Area. [Disponible en: <http://archive.plugins.jquery.com/project/PrintArea>]. [Consulta: 16 de Noviembre de 2012].

La Webera.com. 2011. Drag and Drop en HTML5. [Disponible en: <http://www.lawebera.es/disenio-web-html-5/drag-and-drop-en-html5.php>]. [Consulta: 20 de Octubre de 2012].

MongoDB. 2012. Zona para desarrolladores. [Disponible en: <http://es.wiki.mongodb.org/display/DOCS/Developer+Zone>]. [Consulta: 6 de Noviembre de 2012].

MongoDB. 2012. Installing MongoDB. [Disponible en: <http://docs.mongodb.org/manual/installation/>]. [Consulta: 6 de Noviembre de 2012].

MongoDB. 2012. Updating. [Disponible en: <http://api.mongodb.org/wiki/current/Updating.html>]. [Consulta: 6 de Noviembre de 2012].

NodeJs. 2012. NodeJs. [Disponible en: <http://nodejs.org/>]. [Consulta: 7 de Agosto de 2012].

PHP. 2012. Driver Nativo MongoDB. [Disponible en: <http://php.net/manual/es/book.mongo.php>]. [Consulta: 3 de Agosto de 2012].

TableSorter. [s.a.]. Documentation. [Disponible en: <http://tablesorter.com/docs/>]. [Consulta: 26 de Noviembre de 2012].

The World Wide Web Consortium (W3C). 2011. HTML. [Disponible en: <http://www.w3.org/community/webed/wiki/HTML>]. [Consulta: 27 de Noviembre de 2012].

The World Wide Web Consortium (W3C). 2012. HTML5.1 Nightly. [Disponible en: <http://www.w3.org/html/wg/drafts/html/master/single-page.html>]. [Consulta: 27 de Noviembre de 2012].

Ultra Mega Blog. 2011. Create Snake in JavaScript with HTML5 Canvas. [Disponible en: <http://www.ultramegatech.com/2011/08/create-snake-in-javascript/>]. [Consulta: 3 de Noviembre de 2012].

Ver Tutoriales – Formación Audiovisual. 2011. Tutorial HTML5 – Dibujando sobre el canvas. [Disponible en: <http://www.vertutoriales.com/index.php/tutorial-html5-dibujando-sobre-el-canvas/>]. [Consulta: 22 de Agosto de 2012].

Wikipedia La enciclopedia libre. 2012. JQuery. [Disponible en: <http://es.wikipedia.org/wiki/JQuery>]. [Consulta: 1 de Diciembre de 2012].

Wikipedia La enciplopedia libre. 2012. JQuery UI. [Disponible en: http://es.wikipedia.org/wiki/JQuery_UI]. [Consulta: 1 de Diciembre de 2012].

Wikipedia La enciplopedia libre. 2012. NodeJs. [Disponible en: <http://es.wikipedia.org/wiki/Nodejs>]. [Consulta: 1 de Diciembre de 2012].

W3 Schools. [s.a.]. HTML Event Attributes. [Disponible en: http://www.w3schools.com/tags/ref_eventattributes.asp]. [Consulta: 26 de Noviembre de 2012].

W3 Schools. [s.a.]. HTML Reference – (HTML5 Compliant). [Disponible en: <http://www.w3schools.com/tags/>]. [Consulta: 25 de Noviembre de 2012].

Youtube - Canal de Betabeers. 2012. Taller nodejs con nacho soto. [Disponible en: <http://www.youtube.com/watch?v=E1FcQPYoZZk>]. [Consulta: 20 de Junio de 2012].