



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

“ANÁLISIS ESTADÍSTICO DE VISORES WEB OPEN SOURCE
PARA LA IMPLEMENTACIÓN DE LA INFRAESTRUCTURA DE
DATOS ESPACIALES EN LA ESPOCH”

TESIS DE GRADO

Previa la obtención del título de:
INGENIEROS EN SISTEMAS INFORMÁTICOS

Presentado por:

Damaris Elizabeth Chinlli Pomaquero
Lorena del Pilar Morales Rivera

RIOBAMBA – ECUADOR

2012

AGRADECIMIENTO

A Dios, por estar siempre con nosotras dándonos la fortaleza necesaria para seguir adelante y así haber permitido que culminemos una etapa más en nuestras vidas.

A la ESPOCH por acogernos como estudiantes que aspiraban cumplir una meta y darnos la oportunidad de educarnos en esta prestigiosa institución, además por incentivarlos a la investigación con el fin de que nuestro desempeño en la vida laboral sea excelente.

También queremos agradecer a nuestra directora de tesis Ing. Lorena Aguirre, por guiarnos durante este trabajo de investigación mediante sus conocimientos y experiencia.

Agradecemos de igual forma al Ing. Jorge Huilca e Ing. Fernando Romero por su disposición de tiempo al contribuir por medio de sus conocimientos al desarrollo y ejecución de esta tesis.

Y por último pero no menos importante a nuestros buenos amigos quienes nos acompañaron y ayudaron durante nuestra etapa universitaria.

DEDICATORIA

A mis padres que por medio de sus consejos, comprensión y apoyo en todo sentido velaron por mi futuro.

A mis hermanos por motivarme y apoyarme siempre para que cumpla un objetivo más en mi vida.

Damaris Elizabeth Chinlli Pomaquero

A Dios quien me dio la fe, la fortaleza, la salud y la esperanza para terminar este trabajo.

A mis padres quienes con su apoyo incondicional me enseñaron a luchar para alcanzar mis objetivos.

A mis hermanos, quienes supieron brindarme su comprensión y cariño el cual me ha ayudado e impulsado a seguir adelante a pesar de las adversidades.

Lorena del Pilar Morales Rivera

FIRMAS DE RESPONSABLES Y NOTA

NOMBRES	FIRMAS	FECHA
Ing. Iván Menes
DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA		
Ing. Raúl Rosero
DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS		
Ing. Lorena Aguirre
DIRECTOR DE TESIS		
Ing. Jorge Menéndez
MIEMBRO DEL TRIBUNAL		
Tlgo. Carlos Rodríguez.....
Dir. Dpto. CENTRO DOCUMENTACIÓN		
NOTA DE LA TESIS.....		

“Nosotras, Damaris Elizabeth Chinlli Pomaquero y Lorena del Pilar Morales Rivera, somos las responsables de las ideas, doctrinas y resultados expuestos en esta tesis, y el patrimonio intelectual de la misma pertenece a la Escuela Superior Politécnica De Chimborazo”.

Damaris Elizabeth Chinlli Pomaquero

Lorena del Pilar Morales Rivera

ÍNDICE DE ABREVIATURAS

CGI: Common Gateway Interface.

CSW: Web Catalogue Service

CRC: Clase, Responsabilidad, Colaboración,

DHTML: HTML Dinámico.

GDAL: Geospatial Data Abstraction Library.

GML: Lenguaje de Marcado Geográfico

GEOS: Geometry Engine Open Source.

IDE: Infraestructura de Datos Espaciales.

OGC: Open Geospatial Consortium

SIG: Sistema de Información Geográfica.

SHP: Shapefile

WMS: Web Map Service

WFS: Web Features Service

WCS: Web Coverage Service

XP: Xtreme Programming o Programacion Extrema.

.

ÍNDICE GENERAL

AGRADECIMIENTO

DEDICATORIA

ÍNDICE DE ABREVIATURAS

INTRODUCCIÓN

CAPÍTULO I

1. MARCO REFERENCIAL	19
1.1. ANTECEDENTES	19
1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS O MEMORIA	22
1.2.1. Justificación Teórica.....	22
1.2.2. Justificación Práctica	23
1.3. OBJETIVOS	23
1.3.1. Objetivos Generales	23
1.3.2. Objetivos Específicos	23
1.4. HIPÓTESIS	24

CAPÍTULO II

2. MARCO TEÓRICO	26
2.1. SISTEMA DE INFORMACIÓN GEOGRÁFICA.....	26
2.1.1. Evolución	26
2.1.2. Definición.....	26
2.1.3. Importancia de los SIG.....	27
2.1.4. Funcionamiento de un SIG	28
2.1.5. Características de los SIG	29
2.1.6. Ventajas de los SIG	30
2.1.7. Desventajas de los SIG.....	30

2.1.8.	Componentes de un SIG	30
2.1.9.	El dato geográfico	32
2.1.10.	Modelos de representación	33
2.1.11.	Aplicaciones de los SIG	34
2.2.	INFRAESTRUCTURA DE DATOS ESPACIALES	35
2.2.1.	Introducción	35
2.2.2.	Definición.....	36
2.2.3.	Principios de una IDE	37
2.2.4.	Lugares de implantación de una IDE	37
2.2.5.	Objetivos IDE	38
2.2.6.	Razones para implantar una IDE.....	38
2.2.7.	Componentes de una IDE	38
2.2.8.	Componentes técnicos del geoportal.....	44
2.3.	CARTOWEB	56
2.3.1.	Características.....	56
2.3.2.	Arquitectura.....	57
2.3.3.	Estructura del directorio de Global	58
2.3.4.	Funcionamiento general.....	60
2.3.5.	Lenguajes de programación utilizadas	61
2.4.	OPENLAYERS.....	66
2.4.1.	Características.....	67
2.4.2.	Arquitectura.....	68
2.4.3.	Ventajas de OpenLayers.....	68
2.4.4.	Tipos de capas que se puede cargar con OpenLayers.....	69
2.4.5.	Manejo de herramientas	70
2.4.6.	Lenguajes de programación utilizadas	73
2.4.7.	Clases básicas	74

2.4.8.	Controles	76
2.4.9.	Capas.....	79
2.4.10.	Eventos.....	82
2.4.11.	Problemas en accesos locales	82
2.5.	VISOR CUENCA	83
2.5.1.	Introducción	83
2.5.2.	Características.....	84
2.5.3.	Tipos de capas que se puede cargar con el Visor Cuenca	84
2.5.4.	Manejo de herramientas	85
2.5.5.	Lenguajes de programación y clases usadas.....	87

CAPÍTULO III

3.	CONFIGURACIÓN Y DESARROLLO DEL PROTOTIPO	26
3.1.	Instalación y uso de CartoWeb.....	26
3.2.	Instalación y uso de OpenLayers	104
3.3.	Instalación y uso del Visor Cuenca	112

CAPÍTULO IV

4.	ANÁLISIS ESTADÍSTICO Y COMPROBACIÓN DE LA HIPÓTESIS	130
4.1.	Descripción de prototipo	130
4.2.	Definición de los parámetros de comparación	130
4.3.	Definición de los indicadores	130
4.4.	Criterios de evaluación	131
4.5.	Análisis de los parámetros de comparación	132
4.6.	Resumen general.....	149
4.7.	Comprobación de la hipótesis.....	151
4.7.1.	Hipótesis	151
4.7.2.	Tipo de hipótesis	151
4.7.3.	Población y Muestra	151
4.7.4.	Determinación de las variables	151
4.7.5.	Operacionalización Conceptual	152
4.7.6.	Operacionalización Metodológica	153
4.7.7.	Demostración de la Hipótesis.....	155

CAPÍTULO V

5.	IMPLEMENTACIÓN DEL GEOPORTAL	156
5.1.	Metodología Programación Extrema (XP)	156
5.1.1.	Planificación	157
5.1.2.	Diseño	166
5.1.3.	Desarrollo	175
5.1.4.	Pruebas.....	203

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

TABLA II.I. TABLA DE COMPARACIÓN MODELO VECTORIAL VS. MODELO RÁSTER	34
TABLA II.II. PARÁMETROS OBJETO MAP	50
TABLA II.III. PARÁMETROS OBJETO REFERENCE	51
TABLA II.IV. PARÁMETROS OBJETO SCALEBAR	52
TABLA II.V. PARÁMETROS OBJETO LAYER	53
TABLA II.VI. PARÁMETROS OBJETO CLASS	53
TABLA II.VII. PARÁMETROS OBJETO STYLE.....	54
TABLA II.VIII. PARÁMETROS OBJETO LABEL.....	54
TABLA II.IX. VENTAJAS DE OPENLAYERS FRENTE A GOOGLEMAPS	69
TABLA II.X. TIPOS COMUNES DE CAPAS USADOS EN OPENLAYERS.....	70
TABLA II.XI. RESUMEN OPENLAYERS	70
TABLA II.XII. TIPOS COMUNES DE CAPAS USADOS EN VISOR CUENCA.....	85
TABLA III.XIII. PUERTOS DE LAS APLICACIONES	112
TABLA IV.XIV. PARÁMETROS DE COMPARACIÓN	130
TABLA IV.XV. INDICADORES	131
TABLA IV.XVI. CRITERIOS DE EVALUACIÓN GENERAL	131
TABLA IV.XVII. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO COMPATIBILIDAD ..	132
TABLA IV.XVIII. PARÁMETRO COMPATIBILIDAD.....	133
TABLA IV.XIX. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO SOPORTE DE LENGUAJE.....	134
TABLA IV.XX. PARÁMETRO SOPORTE DE LENGUAJE	134
TABLA IV.XXI. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO INSTALACIÓN	135
TABLA IV.XXII. PARÁMETRO INSTALACIÓN.....	136
TABLA IV.XXIII. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO USABILIDAD	137
TABLA IV.XXIV. PARÁMETRO USABILIDAD	137
TABLA IV.XXV. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO INTEGRACIÓN DE WMS.....	139
TABLA IV.XXVI. PARÁMETRO INTEGRACIÓN DE WMS	139
TABLA IV.XXVII. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO GESTIÓN DE CAPAS	140
TABLA IV.XXVIII. PARÁMETRO GESTIÓN DE CAPAS	140
TABLA IV.XXIX. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO EXPORTACIÓN DE ARCHIVOS.....	142
TABLA IV.XXX. PARÁMETRO EXPORTACIÓN DE ARCHIVOS.....	142
TABLA IV.XXXI. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO SOPORTE DE SERVICIOS	143
TABLA IV.XXXII. PARÁMETRO SOPORTE DE SERVICIOS.....	143
TABLA IV.XXXIII. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO TIPO DE FORMATOS ACEPTADOS	145
TABLA IV.XXXIV. PARÁMETRO TIPO DE FORMATOS ACEPTADOS.....	145
TABLA IV.XXXV. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO TIEMPO DE EJECUCIÓN.....	146
TABLA IV.XXXVI. PARÁMETRO TIEMPO DE EJECUCIÓN.....	146
TABLA IV.XXXVII. CRITERIOS DE EVALUACIÓN PARA EL PARÁMETRO PLATAFORMAS SOPORTADAS	147
TABLA IV.XXXVIII. PARÁMETRO PLATAFORMAS SOPORTADAS	148
TABLA IV.XXXIX. COMPARATIVA GENERAL - PROTOTIPO	149
TABLA IV.XL. SUMA TOTAL DEL PROTOTIPO Y ENCUESTA.....	150
TABLA IV.XLI. COMPARATIVA GENERAL	150
TABLA IV.XLII. OPERACIONALIZACIÓN CONCEPTUAL	152
TABLA IV.XLIII. OPERACIONALIZACIÓN METODOLÓGICA.....	153

TABLA V.XLIV. REQUERIMIENTOS FUNCIONALES.....	158
TABLA V.XLV. TARJETA HISTORIA DE USUARIO 1	159
TABLA V.XLVI. TARJETA HISTORIA DE USUARIO 2.....	159
TABLA V.XLVII. TARJETA HISTORIA DE USUARIO 3	160
TABLA V.XLVIII. TARJETA HISTORIA DE USUARIO 4	160
TABLA V.XLIX. TARJETA HISTORIA DE USUARIO 5	161
TABLA V.L. TARJETA HISTORIA DE USUARIO 6.....	161
TABLA V.LI. TARJETA HISTORIA DE USUARIO 7.....	162
TABLA V.LII. TARJETA HISTORIA DE USUARIO 8	162
TABLA V.LIII. TARJETA HISTORIA DE USUARIO 9	163
TABLA V.LIV. VELOCIDAD DEL PROYECTO.....	165
TABLA V.LV. TARJETA CRC ADMINISTRADOR	166
TABLA V.LVI. TARJETA CRC CAPA.....	166
TABLA V.LVII. IDENTIFICACIÓN DEL RIESGO	168
TABLA V.LVIII. VALORACIÓN DEL RIESGO.....	168
TABLA V.LIX. PROBABILIDAD	168
TABLA V.LX. IMPACTO DEL RIESGO.....	169
TABLA V.LXI. RIESGO – IMPACTO.....	169
TABLA V.LXII. EXPOSICIÓN AL RIESGO	170
TABLA V.LXIII. IMPACTO – PROBABILIDAD.....	170
TABLA V.LXIV. RESUMEN DEL RIESGO	170
TABLA V.LXV. PRIORIDADES DEL RIESGO.....	171
TABLA V.LXVI. GESTIÓN DEL RIESGO 1	172
TABLA V.LXVII. GESTIÓN DEL RIESGO 2	173
TABLA V.LXVIII. GESTIÓN DEL RIESGO 3	174
TABLA V.LXIX. GESTIÓN DEL RIESGO 4	175

ÍNDICE FIGURAS

FIGURA II.1. SIG.....	27
FIGURA II.2. TIPOS DE INFORMACIÓN GEOGRÁFICA	29
FIGURA II.3. COMPONENTES DE UN SIG	32
FIGURA II.4. COMPONENTE ESPACIAL – COMPONENTE TEMÁTICO.....	32
FIGURA II.5. MODELO RÁSTER VS. MODELO VECTORIAL.....	34
FIGURA II.6. COMPONENTES DE UNA IDE	39
FIGURA II.7. SERVICIOS IDE	41
FIGURA II.8. TECNOLOGÍA GEOPORTAL	45
FIGURA II.9. FUNCIONAMIENTO MAPSERVER.....	47
FIGURA II.10. RECURSOS MAPSERVER.....	48
FIGURA II.11. ESTRUCTURA ARCHIVO .MAP	49
FIGURA II.12. ESTRUCTURA PLUGINS.....	58
FIGURA II.13. ESQUEMA REPRESENTATIVO DE LA ARQUITECTURA DE OPENLAYERS ..	68
FIGURA II.14. LAYER SWITCHER	71
FIGURA II.15. PAN ZOOM BAR	71
FIGURA II.16. OVERVIEW MAP	72
FIGURA II.17. MOUSE TOOLBAR.....	72
FIGURA II.18. SCALE LINE	72
FIGURA II.19. MOUSE POSITION.....	73
FIGURA II.20. ESQUEMA DE ACCESO A DATOS OPENLAYERS	83
FIGURA II.21. AMPLIAR UNA ZONA.....	85
FIGURA II.22. ZOOM OUT	85
FIGURA II.23. ZOOM IN	85
FIGURA II.24. PANEO.....	85
FIGURA II.25. ESCALA.....	86
FIGURA II.26. IMPRIMIR	86
FIGURA II.27. POSICIÓN DEL MOUSE.....	86
FIGURA II.28. VOLVER AL INICIO.....	86
FIGURA II.29. MAPA DE REFERENCIA	86
FIGURA III. 30. COMANDO INSTALACIÓN CARTOWEB	94
FIGURA III. 31. ERROR DEL SERVIDOR	94
FIGURA III. 32. MODIFICACIÓN DE ARCHIVO HTTPD.CONF.....	94
FIGURA III. 33. INSTALACIÓN CORRECTA.....	95
FIGURA III. 34. ARCHIVOS INICIALES PROTOTIPO	95
FIGURA III. 35. SCRIPT DE INSTALACIÓN PROTOTIPO	96
FIGURA III. 36. MODIFICACIÓN ARCHIVO PROTOTIPO.PHP	96
FIGURA III. 37. EJECUCIÓN DE COMANDO PARA LA RECONSTRUCCIÓN DEL PROYECTO PROTOTIPO	96
FIGURA III. 38. PROYECTO PROTOTIPO INICIAL.....	97
FIGURA III. 39. DIRECTORIO CLIENT_CONF Y SERVER_CONF	97
FIGURA III. 40. DIRECTORIO HTDOCS	97
FIGURA III. 41. MAPA DE REFERENCIA	98
FIGURA III. 42. PROTOTIPO.MAP - MAPA DE REFERENCIA.....	98
FIGURA III. 43. INFORMACIÓN ALMACENADA EN DIRECTORIO DATA	98
FIGURA III. 44. REFERENCIA DE LA CAPA BASE ECUADOR	98
FIGURA III. 45. DIRECTORIO COREPLUGINS Y PLUGINS	99
FIGURA III. 46. CAPAS EN EL ARCHIVO PROTOTIPO.MAP	99
FIGURA III. 47. CREACIÓN DE LEYENDA EN EL ARCHIVO LAYERS.INI	99

FIGURA III. 48. RESULTADO LEYENDA.....	100
FIGURA III. 49. PROTOTIPO.INI.....	100
FIGURA III. 50. RESULTADO MAPA DEL ECUADOR	100
FIGURA III. 51. DIRECTORIO HTDOCS Y TEMPLATES	100
FIGURA III. 52. ARCHIVO LAYERS.TPL	101
FIGURA III. 53. LEYENDA.....	101
FIGURA III. 54. ACTIVACIÓN DE AJAX.....	101
FIGURA III. 55. ARCHIVO CARTOCLIENT_AJAXHEADER.TPL	102
FIGURA III. 56. USO DE AJAX	102
FIGURA III. 57. PARÁMETROS DE CONFIGURACIÓN PARA EXPORTACIÓN PDF	102
FIGURA III. 58. EXPORTACIÓN PDF.....	103
FIGURA III. 59. ARCHIVO PDF CON EL MAPA	103
FIGURA III. 60. PROTOTIPO CARTOWEB.....	104
FIGURA III. 61. API OPENLAYERS-2-10.....	105
FIGURA III. 62. REFERENCIA A LA API DE OPENLAYERS	105
FIGURA III. 63. URL PARA COMPROBRAR ACCESO A OPENLAYERS	106
FIGURA III. 64. ARCHIVO PROTOTIPO.MAP	106
FIGURA III. 65. EXTENSIÓN DEL MAPA EN PROTOTIPO.MAP	106
FIGURA III. 66. CAPAS DEL MAPA EN PROTOTIPO.MAP	107
FIGURA III. 67. DIRECTORIO DE INFORMACIÓN GEOESPACIAL	107
FIGURA III. 68. DIRECTORIO DE INFORMACION GEOESPACIAL	107
FIGURA III. 69. ESTILOS APLICADOS A LOS OBJETOS DE OPENLAYERS.....	108
FIGURA III. 70. FUNCIÓN INIT().....	108
FIGURA III. 71. LLAMAR A LA FUNCION INIT().....	109
FIGURA III. 72. OBJETO MAP	109
FIGURA III. 73. CONTROLES	109
FIGURA III. 74. CONTROLES	110
FIGURA III. 75. CONTROLES	110
FIGURA III. 76. CAPAS	111
FIGURA III. 77. CAPAS	111
FIGURA III. 78. PROTOTIPO OPENLAYERS	112
FIGURA III. 79. INSTALACIÓN VISOR	113
FIGURA III. 80. ARCHIVO PARA AGREGAR CLAVE	113
FIGURA III. 81. CLAVE DE GOOGLE.....	113
FIGURA III. 82. DIRECTORIOS APLICACIÓN WEB.....	114
FIGURA III. 83. PATH IDEESPOCH.MAP	116
FIGURA III. 84. CREACION IDEESPOCH.MAP	116
FIGURA III. 85. DIRECTORIO ARCHIVO SHAPES	116
FIGURA III. 86. PATH DE SERVERS_GOOGLE.JS	116
FIGURA III. 87. CAPAS Y DIRECCIÓN DEL SERVIDOR.....	117
FIGURA III. 88. PATH DE ARBOL .JS.....	117
FIGURA III. 89. CREACIÓN DEL ÁRBOL	117
FIGURA III. 90. VISUALIZAR EL ÁRBOL.....	117
FIGURA III. 91. PATH DE INDEX.JS.....	118
FIGURA III. 92. PATH DE INDEX.JSP	118
FIGURA III. 93. REFERENCIA A LAS LIBRERIAS UTILIZADAS	118
FIGURA III. 94. DEFINICIÓN DE LA PROYECCIÓN	118
FIGURA III. 95. DEFINICIÓN DEL BOUND	119
FIGURA III. 96. CARGAR MAPAS DE GOOGLE.....	119
FIGURA III. 97. CONTROLES DEL MAPA.....	119
FIGURA III. 98. REFERENCIA DEL MAPA	119
FIGURA III. 99. CENTRAR LAS CAPAS SOBRE UNA ZONA.....	120
FIGURA III. 100. CREACIÓN DE LA LEYENDA	120
FIGURA III. 101. PATH DE BUSQUEDAS.JS.....	120
FIGURA III. 102. BÚSQUEDA.....	121

FIGURA III. 103. IMPRESIÓN.....	121
FIGURA III. 104. PATH DE SERVIDORES.JS.....	121
FIGURA III. 105. CARGA LA CAPA AL VISOR.....	121
FIGURA III. 106. PROTOTIPO VISOR CUENCA.....	122
FIGURA III. 107. PATH DEL AIDEUCUENCA.....	122
FIGURA III. 108. PATH DE AIDEUCUENCA.MAP.....	123
FIGURA III. 109. CONFIGURACIÓN DE AIDEUCUENCA.MAP.....	123
FIGURA III. 110. PATH CONFIGURACION_EDITOR.JSP.....	123
FIGURA III. 111. ARCHIVO USUARIOS.DBF.....	124
FIGURA III. 112. DIRECTORIOS APLICACIÓN WEB.....	124
FIGURA III. 113. PATH DE EDICION.JSP.....	126
FIGURA III. 114. AUTENTICACIÓN DE USUARIO.....	126
FIGURA III. 115. CREACIÓN DE UN SHAPE.....	126
FIGURA III. 116. ELIMINACIÓN DE UN SHAPE.....	127
FIGURA III. 117. MODIFICACIÓN DE UN SHAPE.....	127
FIGURA III. 118. SUBIR UN SHAPE Y VISUALIZARLO.....	127
FIGURA III. 119. DESCARGAR UN SHAPE.....	128
FIGURA III. 120. EDITOR DE OBJETOS.....	128
FIGURA IV. 121. PARÁMETRO COMPATIBILIDAD NAVEGADORES.....	133
FIGURA IV.122. PARÁMETRO SOPORTE DE LENGUAJE.....	135
FIGURA IV. 123. PARÁMETRO INSTALACIÓN.....	136
FIGURA IV. 124. PARÁMETRO USABILIDAD.....	138
FIGURA IV. 125. PARÁMETRO DE INTEGRACIÓN DE WMS.....	139
FIGURA IV.126. PARÁMETRO DE GESTIÓN DE CAPAS.....	141
FIGURA IV. 127. PARÁMETRO DE EXPORTACIÓN DE ARCHIVOS.....	142
FIGURA IV. 128. PARÁMETRO DE SOPORTE DE SERVICIOS.....	144
FIGURA IV. 129. PARÁMETRO DE FORMATOS ACEPTADOS.....	145
FIGURA IV. 130. PARÁMETRO DE TIEMPO DE EJECUCIÓN.....	147
FIGURA IV. 131. PARÁMETRO DE PLATAFORMAS SOPORTADAS.....	148
FIGURA IV. 132. RESUMEN DE LA COMPARATIVA GENERAL ENTRE CARTOWEB, OPENLAYERS Y VISOR CUENCA.....	150
FIGURA IV. 133. DEMOSTRACIÓN DE LA HIPÓTESIS.....	155
FIGURA IV. 134. VELOCIDAD DEL PROYECTO.....	165
FIGURA V. 135. EJECUCIÓN DE INSTALADOR.....	177
FIGURA V. 136. UBICACIÓN DEL DIRECTORIO MAPSERVER.....	177
FIGURA V. 137. PUERTO ABIERTO A PETICIÓN.....	178
FIGURA V. 138. SERVICIO LEVANTADO.....	178
FIGURA V. 139. COMPROBACIÓN DE PAQUETE.....	178
FIGURA V. 140. INSTALACIÓN DE HTTPD.....	179
FIGURA V. 141. PRUEBA DE HTTPD INSTALADO.....	179
FIGURA V. 142. INSTALACIÓN DE PHP.....	180
FIGURA V. 143. DESCARGA E INSTALACIÓN DE PHP.....	180
FIGURA V. 144. INSTALACIÓN DE MYSQL.....	180
FIGURA V. 145. DESCARGA E INSTALACIÓN DE MYSQL.....	181
FIGURA V. 146. REINICIANDO APACHE.....	181
FIGURA V. 147. INSTALACION DE LA DB.....	181
FIGURA V. 148. LEVANTAMIENTO DE SERVICIO.....	182
FIGURA V. 149. REINICIANDO MYSQLD.....	182
FIGURA V. 150. CONFIGURACION DE MYSQL.....	182
FIGURA V. 151. INGRESO DE PASSWORD PARA LA DB.....	183
FIGURA V. 152. REINICIANDO MYSQLD.....	183
FIGURA V. 153. EJECUCIÓN DE JOOMLA.....	183
FIGURA V. 154. SELECCIÓN DE IDIOMA.....	184
FIGURA V. 155. CONTRATO DE LICENCIA.....	184
FIGURA V. 156. CONFIGURACIÓN DE BASE DE DATOS.....	184

FIGURA V. 157. CONFIGURACIÓN DEL SITIO WEB	185
FIGURA V. 158. INSTALACIÓN FINALIZADA	185
FIGURA V. 159. BACKEND DE JOOMLA	186
FIGURA V. 160. PERMISOS AL INSTALADOR	186
FIGURA V. 161. ACEPTACIÓN DE LICENCIA	186
FIGURA V. 162. COMPROBACIÓN DE INSTALACIÓN CORRECTA	187
FIGURA V. 163. INGRESO AL ARCHIVO PROFILE	187
FIGURA V. 164. CONFIGURACIÓN DEL ARCHIVO PROFILE	187
FIGURA V. 165. ACTUALIZACIÓN ARCHIVO PROFILE	188
FIGURA V. 166. INGRESO DE RUTA PARA EJECUTAR SERVIDOR	188
FIGURA V. 167. EJECUCIÓN DEL SERVIDOR	188
FIGURA V. 168. INGRESO DE RUTA PARA CONFIGURAR EL PUERTO	188
FIGURA V. 169. CONFIGURAR PUERTO	189
FIGURA V. 170. CAMBIO DE PUERTO	189
FIGURA V. 171. INGRESO DE RUTA PARA CONFIGURAR USUARIO Y CONTRASEÑA	189
FIGURA V. 172. CONFIGURACIÓN DE CREDENCIALES	189
FIGURA V. 173. INGRESO DE CREDENCIALES	189
FIGURA V. 174. AUTENTICACIÓN	190
FIGURA V. 175. INGRESO AL DIRECTORIO PARA CREAR UN SCRIPT	190
FIGURA V. 176. COMANDO PARA EDITAR EL ARCHIVO TOMCAT	190
FIGURA V. 177. INGRESO DE CÓDIGO	191
FIGURA V. 178. PERMISOS PARA EJECUTAR EL CÓDIGO	191
FIGURA V. 179. AGREGAR SCRIPT A SERVICIOS DE SISTEMA	191
FIGURA V. 180. VERIFICAR MODIFICACIONES	192
FIGURA V. 181. LINK SIMBOLICO	192
FIGURA V. 182. PARAR SERVICIO TOMCAT	192
FIGURA V. 183. INICIAR SERVICIO TOMCAT	192
FIGURA V. 184. AGREGAR PUERTO 8085	193
FIGURA V. 185. DIRECTORIO DE PUBLICACIÓN VISOR DE CUENCA	193
FIGURA V. 186. UBICACIÓN SERVERS_GOOGLE.JS	193
FIGURA V. 187. ARCHIVO SERVERS_GOOGLE.JS	194
FIGURA V. 188. UBICACIÓN DE ARCHIVO ÁRBOL.JS	194
FIGURA V. 189. CÓDIGO JAVASCRIPT RELACIONADO CON LA LEYENDA	195
FIGURA V. 190. ARCHIVO IDEUCUENCA.MAP	195
FIGURA V. 191. CAPAS DE LA PROVINCIA DE CHIMBORAZO	196
FIGURA V. 192. PATH DE EDITOR DE MAPAS (AIDEUCUENCA)	196
FIGURA V. 193. USUARIOS.DBF	196
FIGURA V. 194. PATH DE INDEX.HTML E INDEX1.HTML	197
FIGURA V. 195. PATH DEL ARCHIVO CONFIGURACION_EDITOR.JSP	197
FIGURA V. 196. ARCHIVO CONFIGURACION_EDITOR.JSP	198
FIGURA V. 197. ARCHIVO AIDEUCUENCA.MAP	198
FIGURA V. 198. ARCHIVO .JSP	199
FIGURA V. 199. PÁGINA DE EDITOR DE OBJETOS	199
FIGURA V. 200. EJECUCIÓN DE INSTALADOR GEONETWORK	200
FIGURA V. 201. ACEPTACIÓN DE LICENCIA	200
FIGURA V. 202. RUTA DE GEONETWORK	200
FIGURA V. 203. PROGRESO DE INSTALACIÓN	201
FIGURA V. 204. FIN DE INSTALACIÓN	201
FIGURA V. 205. ENLACES SIMBÓLICOS	202
FIGURA V. 206. COMPROBACIÓN GEONETWORK BAJO TOMCAT	202
FIGURA V. 207. GEONETWORK INSTALADO	202
FIGURA V. 208. METADATOS CREADOS	203
FIGURA V. 209. GEOPORTAL	203
FIGURA V. 210. PANTALLA INICIO GEOPORTAL	204
FIGURA V. 211. DIFUSIÓN DE CAPAS DE CHIMBORAZO	205

FIGURA V. 212. PUBLICACIÓN DE INFORMACIÓN GENERAL	206
FIGURA V. 213. INTEGRACIÓN VISOR DE MAPAS.....	206
FIGURA V. 214. INTEGRACIÓN EDITOR DE CAPAS.....	207
FIGURA V. 215. CATÁLOGO DE GEONETWORK	208
FIGURA V. 216. GALERÍA DE FOTOS.....	209

INTRODUCCIÓN

La información geográfica siempre ha formado parte de la sociedad, este ha sido empleado en ambientes o aplicaciones de escritorio, diseñadas para el uso especialista y académico. Sin embargo como complemento a esto en los últimos años han surgido nuevas tecnologías web que aportan a la difusión de la información geográfica a través del internet para el público general.

Los visores web open source aparecen dentro de estas tecnologías y conjuntamente con la necesidad de propagar información geoespacial de la provincia de Chimborazo a través del internet, lo que motivo al desarrollo de este trabajo de investigación, ya que mediante el mismo se seleccionó la herramienta más adecuada para la difusión de información.

A continuación se describirá el desarrollo de la tesis contenido en 5 capítulos:

El capítulo I abarca el marco referencial donde se describe el motivo de la investigación, justificación, objetivos, e hipótesis.

El capítulo II incluye el marco teórico en el cual se mencionan los conceptos, terminologías, información relacionada al objeto de estudio.

El capítulo III encierra el desarrollo del prototipo con cada visor web que en el capítulo siguiente será indispensable.

El capítulo IV comprende el análisis estadístico a través de un prototipo desarrollado para cada visor web que en este caso son 3 los estudiados, cada prototipo será sometido a parámetros establecidos que ayuden a determinar el más adecuado para la visualización de información, finalizando así con la comprobación de la hipótesis.

El capítulo V engloba la construcción de la parte aplicativa de la tesis que es el desarrollo del geoportal con el visor seleccionado, mediante el uso de la metodología XP.

CAPÍTULO I

1. MARCO REFERENCIAL

En este capítulo se presentará el marco referencial, en el cual se encuentra la descripción de los antecedentes, la justificación del proyecto de tesis, el objetivo general y los específicos y el planteamiento de la hipótesis.

1.1. ANTECEDENTES

El mundo de los SIG evoluciona rápidamente, como sucede con cualquier tecnología estrechamente vinculada al mundo de la informática, e igual sucede con los elementos que la componen: software, datos, equipos, recursos humanos, etc. Uno de los elementos que está sufriendo una mayor evolución en los últimos tiempos es la información disponible para su empleo en SIG, es decir: los datos. Esta evolución “reciente” consiste principalmente en la potenciación del acceso de los usuarios a la información disponible por parte de los diferentes agentes públicos o privados que poseen información cartográfica de interés para la gestión ambiental, territorial, así como también la promoción del desarrollo sostenible, social y económico y las políticas gubernamentales.

La información espacial generada por instituciones requiere en estos momentos de un sistema informativo integral que asegure la adecuada captación, procesamiento y flujo informativo, que constituye un instrumento importante en la evaluación de los progresos o retrocesos del estado del medio ambiente, naciendo así la Infraestructura de Datos Espaciales (IDE), este es un sistema informático integrado por un conjunto de servicios como catálogos, visor y editor de mapas, que ayudan a gestionar Información Geográfica, disponibles en Internet, que cumplen una serie de condiciones de interoperabilidad, tales como normas, especificaciones, protocolos, interfaces, etc. que permiten que un usuario, utilizando un navegador acceda y haga uso de la información geográfica a nivel local, nacional y regional, mediante la formulación de políticas, la estandarización de los datos y la transferencia y aplicación de tecnologías, que permiten actualizar los métodos convencionales de gestión de datos espaciales y satisfacer las necesidades de los usuarios actuales potenciales.

Para especificar más en si el problema de no contar con una infraestructura de información de datos espaciales, a continuación se mencionan los conflictos de mayor peso:

- La información espacial existente está dispersa y no estandarizada.
- No existen normas para la transferencia de los datos entre los participantes.
- Hay dificultades para la obtención de datos (por el nivel de acceso, por el precio, por las restricciones de los propietarios, etc.).
- No se cuenta con la suficiente integración en la tenencia de los datos y las bases cartográficas (incluyendo datos de sensores remotos).
- El universo de la información espacial es abierto y variable. Los portadores de información son muchos, variados e impredecibles.
- No existe total identificación de los responsables de los datos, dentro y fuera de los subsistemas.

En la actualidad la ESPOCH no cuenta con una infraestructura de datos espaciales que le permita al usuario acceder a información espacial de la provincia de Chimborazo.

Como ya se mencionó un visor forma parte en la construcción de un IDE, este es un cliente ligero web SIG que ha adquirido gran importancia en los últimos años debido a la optimización de recursos y a las nuevas tecnologías desarrolladas para mejorar la experiencia de los usuarios en los navegadores web.

Los visores web o clientes ligeros web no requieren de una instalación como en caso de los clientes pesados que sí, ya que para los clientes ligeros solo se requiere un navegador compatible para posibilitar el acceso a la información sencilla y rápida.

El mundo de los visores web es el más dinámico y fértil dentro de las diferentes categorías de tecnologías geoespaciales.

Actualmente existen muchos clientes ligeros web o visores web para SIG, entre ellos los que se mencionan a continuación:

Gmap, Ka-Map, OpenLayers, MapBender, MapBuilder, CartoWeb, Chameleon, visor web de la universidad de Cuenca.

Para el estudio se seleccionaron las más conocidas y/o utilizadas en el ambiente geográfico como CartoWeb, OpenLayers y el visor web del IDE de la universidad Cuenca.

Hay ya muchos países o regiones que cuentan con este tipo de infraestructuras de datos espaciales, los mismos que serán mencionados a continuación:

- IDE ANDALUZA – España.
- IDE DEL CABILDO DE PALMA.
- IDE DE VENEZUELA.
- IDESF. IDE DE SANTA FE – Argentina.
- IDEP- IDE PERU.

Así mismo en el Ecuador existe la iniciativa de la IDE Red CEDIA (Consortio Ecuatoriano para el Desarrollo de Internet Avanzado) que tiene como objeto proporcionar conocimiento geográfico a nivel nacional, el cual lo están conformando algunas universidades como las siguientes:

- IDE CUENCA.
- IDE UTPL.
- IDE UNIANDES.
- IDE UPS.
- IDE ESPE.

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS O MEMORIA

1.2.1. Justificación Teórica

El motivo principal para el desarrollo del análisis estadístico de los visores web Open Source nace del resultado de la observación y posterior reflexión de las necesidades y problemas que surgen a causa de no tener implementado una infraestructura de datos espaciales en la ESPOCH, razón por la cual se va hacer un análisis estadístico de visores web como CartoWeb, OpenLayers y el visor web del IDE de la universidad de Cuenca, permitiendo establecer el más adecuado, en base a mejorar el tiempo de respuesta en el despliegue de información espacial utilizando un navegador, los mismos que permiten mostrar información creando entre el usuario y el sistema una interactividad de petición y visualización de datos teniendo como base la localización geográfica que viene dada mediante la programación y configuración dentro de una página web, además como complemento se deberá realizar un estudio general de recursos tecnológicos como servidor de mapas, entre otros, permitiendo así implementar el IDE enmarcado siempre bajo software libre, ya que implica una menor inversión económica y es la vía más indicada para sistematizar la gestión de los datos espaciales, porque su implementación resuelve necesariamente una serie de aspectos fundamentales que caracterizan hoy en día la información generada en los resultados de los proyectos y que dificultan la transferencia, localización y el trabajo con la información espacial.

Con la Implementación del IDE en la ESPOCH se obtendrá los siguientes beneficios:

- Conocer la disponibilidad de datos espaciales y facilitar su acceso e intercambio.
- Hacer uso de información geográfica estandarizada a nivel local, o regional.
- Brindar soporte al gobierno nacional apoyando las actividades de planeación del uso del territorio, trayendo consigo ser un apoyo para la toma de decisiones objetiva y para una sólida política de administración de tierras.
- Favorecer al país en términos de conocimientos, prosperidad y desarrollo.

1.2.2. Justificación Práctica

Al haber conocido con exactitud los resultados que se produjeron al finalizar el análisis estadístico en un prototipo, permitirá conocer la manera que tiene cada visor web de mostrar información espacial, en cuanto a los parámetros que se hayan establecido dando lugar a seleccionar el más adecuado para ser utilizado en la implementación de un geoportal en la Escuela Superior Politécnica del Chimborazo, ya que mediante su ejecución permitirá tener información geográfica local organizada, integrada y disponible para el uso del usuario de manera que se mejore en lo posible el tiempo de respuesta cuando la información geográfica sea mostrada en el geoportal.

Para la implementación del geoportal con herramientas y servicios que proporcionara la IDE en la ESPOCH se debe realizar la elección de las herramientas tecnológicas libres, que permitan el levantamiento del mismo ofreciendo a los usuarios el acceso y visualización a una serie de recursos y servicios basados en la información geográfica, utilizando un navegador estándar.

1.3. OBJETIVOS

1.3.1. Objetivos Generales

Realizar un análisis estadístico de visores web Open Source para la implementación de la infraestructura de datos espaciales en la ESPOCH.

1.3.2. Objetivos Específicos

- Estudiar los visores web para SIG como CartoWeb, OpenLayers y el visor web del IDE de la universidad de Cuenca y los componentes que tiene una Infraestructura de Datos Espaciales (IDE).
- Determinar los parámetros de comparación entre CartoWeb, OpenLayers y el visor web de la universidad de Cuenca, para establecer el más adecuado en la implementación de una infraestructura de datos espaciales.
- Construir un prototipo para realizar el análisis estadístico entre CartoWeb, OpenLayers y el visor web del IDE de la universidad de Cuenca.
- Aplicar el visor web seleccionado para implantar un geoportal con herramientas y servicios que proporcionara la IDE en la ESPOCH.

1.4. HIPÓTESIS

El análisis estadístico de visores web permitirá establecer el más adecuado para mostrar información espacial en una infraestructura de datos espaciales.

CAPÍTULO II

2. MARCO TEÓRICO

El presente capítulo menciona definiciones conceptuales relacionadas con los visores web: CartoWeb, OpenLayers y el Visor web de la universidad de Cuenca, así como también conceptos involucrados en una infraestructura de datos espaciales.

2.1. SISTEMA DE INFORMACIÓN GEOGRÁFICA

Los Sistemas de Información Geográfica se han constituido durante los últimos veinte (20) años en una de las más importantes herramientas de trabajo para investigadores, analistas y planificadores, etc., en todas las actividades que tienen como insumo el manejo de la información (Bases de Datos), relacionada con diversos niveles de agregación espacial georeferenciada. Aunque los SIG son una valiosa herramienta tecnológica, estos no pueden existir por sí mismos, deben obedecer a una correcta planificación y organización, personal capacitado y un equipamiento con debidas especificaciones, implementación y sostenimiento, adicionalmente este debe tener garantizados los recursos para su mantenimiento.

2.1.1. Evolución

La historia de los SIG se remonta a los inicios de la Humanidad, pero con la llegada de

los años 60's y 70's se empezó a aplicar la tecnología del computador digital al desarrollo de tecnología automatizada.

Excluyendo cambios estructurales en el manejo de la información, la mayoría del desarrollo de tecnología estuvo dirigido hacia la automatización del trabajo cartográfico. "Un Poco de Historia". Solo algunos pocos exploraron nuevos métodos para el manejo de información espacial, y se siguieron básicamente dos tendencias:

- Producción automática de dibujos con alta calidad pictórica
- Producción de información basada en el análisis espacial pero con el costo de una baja calidad gráfica.

La producción automática de dibujos se basó en la tecnología de diseño asistido por computador (Computer Aided Design). El CAD se utilizó en cartografía para aumentar la productividad en la generación y actualización de mapas. El modelo de datos CAD maneja la información espacial como dibujos electrónicos compuestos por entidades gráficas organizadas en planos de visualización o capas.

Cada capa contiene información de cada punto en la pantalla (píxel) que debe encender para la representación por pantalla, estos conjuntos de puntos organizados por planos de visualización se guardan en un formato vectorial.

2.1.2. Definición

SIG, es un conjunto de métodos, herramientas y datos que están diseñados para actuar coordinada y lógicamente para capturar, almacenar, analizar, transformar y presentar toda la información geográfica y de sus atributos con el fin de satisfacer múltiples propósitos.

Los SIG son una nueva tecnología que permite gestionar y analizar la información espacial y que surgió como resultado de la necesidad de disponer rápidamente de información para resolver problemas y contestar a preguntas de modo inmediato. Existen otras muchas definiciones de SIG, algunas de ellas acentúan su componente de base de datos, otras sus funcionalidades y otras enfatizan el hecho de ser una herramienta de apoyo en la toma de decisiones, pero todas coinciden en referirse a un SIG como un sistema integrado para trabajar con información espacial, herramienta esencial para el análisis y toma de decisiones en muchas áreas vitales para el

desarrollo nacional, incluyendo la relacionada con la infraestructura de un municipio, estado o incluso a nivel nacional.

Los SIG o Geographic Information System (GIS) son una nueva tecnología que forma parte de los conocidos como sistemas de información. El contexto general por el que surgen es la **disponibilidad rápida** de **información** para resolver problemas y contestar a las preguntas de modo inmediato.

La amplia difusión de estos sistemas ha propiciado la existencia de numerosas definiciones del término. A modo de ejemplo definiremos los Sistemas de Información Geográfica como un “conjunto de instrumentos y métodos especialmente dispuestos para capturar, almacenar, analizar y presentar información territorial georeferenciada del mundo real”, como se muestra en la figura II.1.

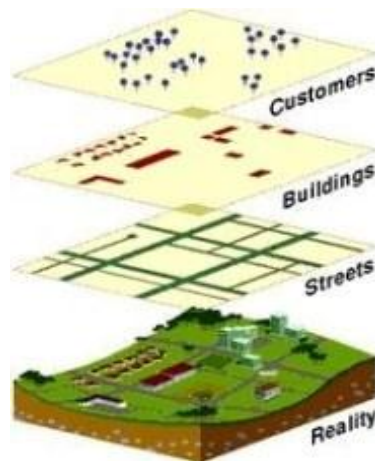


Figura II.1. SIG

Fuente: <http://www.todosig.es/1-1-que-es-un-gis.html>

2.1.3. Importancia de los SIG

Las soluciones para muchos problemas frecuentemente requieren acceso a varios tipos de información que sólo pueden ser relacionadas por geografía o distribución espacial. Sólo la tecnología SIG permite almacenar y manipular información georeferenciada¹ para poder, analizar patrones, relaciones y tendencias en la información.

¹ **Georeferencia:** Es la capacidad de leer y ubicar el posicionamiento de un objeto espacial en su posición correcta.

2.1.4. Funcionamiento de un SIG

La construcción e implementación de un SIG en cualquier organización es una tarea siempre progresiva, compleja, laboriosa y continúa. Los análisis y estudios anteriores a la implantación de un SIG son similares a los que se deben realizar para establecer cualquier otro sistema de información; sin embargo, en los SIG hay que considerar las características especiales de los datos utilizados y sus correspondientes procesos de actualización.

Es indiscutible que los datos son el principal activo de cualquier sistema de información. Por ello el éxito y la eficacia de un SIG se miden por el tipo, la calidad y vigencia de los datos con los que opera.

Los esfuerzos y la inversión necesaria para crear las bases de datos y tener un SIG eficiente y funcional no son pequeños, aunque tampoco significa una gran inversión. Es un esfuerzo permanente por ampliar y mejorar los datos almacenados, utilizando las herramientas más eficientes para tal propósito.

La información geográfica contiene una referencia territorial explícita como latitud y longitud o una referencia implícita como domicilio o código postal. Las referencias implícitas pueden ser derivadas de referencias explícitas mediante geocodificación.

Los SIG funcionan con dos tipos diferentes de información geográfica: el modelo vector y el modelo ráster.

El modelo ráster funciona a través de una retícula que permite asociar datos a una imagen; es decir, se pueden relacionar paquetes de información a los píxeles de una imagen digitalizada, como se muestra en la Figura II.2.

En el modelo vector, la información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas x,y. La ubicación de una característica puntual, pueden describirse con un sólo punto x,y. Las características lineales, pueden almacenarse como un conjunto de puntos de coordenadas x,y. Las características poligonales, pueden almacenarse como un circuito cerrado de coordenadas, como se muestra en la figura II.2.

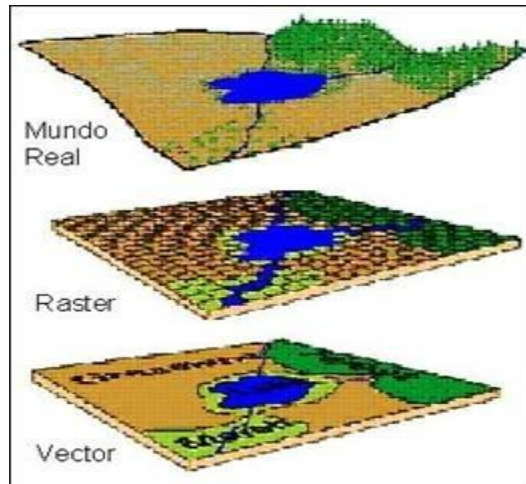


Figura II.2. Tipos de información geográfica
Fuente: Carlos Brenes SERIO-PREPAC

Hoy en día el condicionante principal a la hora de afrontar cualquier proyecto basado en SIG lo constituye la disponibilidad de datos geográficos del territorio a estudiar, mientras que hace diez años lo era la disponibilidad de computadoras potentes que permitieran realizar los procesos de cálculo involucrados en el análisis de datos territoriales.

Pero además de ser un factor limitante, la información geográfica es a su vez el elemento diferenciador de un Sistema de Información Geográfica frente a otro tipo de Sistemas de Información; así, la particular naturaleza de este tipo de información contiene dos vertientes diferentes: por un lado está la vertiente espacial y por otro la vertiente temática de los datos. Mientras otros Sistemas de Información contienen sólo datos alfanuméricos (nombres, direcciones, números de cuenta, etc.), las bases de datos de un SIG integran además la delimitación espacial de cada uno de los objetos geográficos.

2.1.5. Características de los SIG

- Manejo de grandes volúmenes de información.
- Posibilidad de información de distintas fuentes y escalas.
- Rapidez en el procesamiento de la información y obtención de productos cartográficos.
- Capacidad de modelar información.
- Maneja información georeferenciada.

2.1.6. Ventajas de los SIG

- Datos físicamente almacenados en forma completa.
- El mantenimiento y recuperación de datos pueden ser realizados a costos más bajos.
- Posibilita una gran variedad de modelos cartográficos con una mínima inversión de tiempo y dinero.
- Datos espaciales y no espaciales pueden ser analizados simultáneamente en una forma racional.
- Los modelos conceptuales pueden ser probados rápida y repetidamente, facilitando su evaluación.
- Los análisis de cambios temporales pueden ser efectuados eficientemente.
- La adquisición de datos, análisis espacial y procesos de toma de decisiones son integrados en un contexto común de flujo de información.

2.1.7. Desventajas de los SIG

- Costos y problemas técnicos para convertir datos analógicos en formato digital.
- Necesita de especialistas para mantener datos en forma digital en computadoras.
- Falso sentimiento de una mayor confiabilidad y precisión.
- Alto costo de adquisición de equipos y programas necesarios.

2.1.8. Componentes de un SIG

Los componentes necesarios para llevar a cabo las tareas de un GIS, son principalmente, los siguientes:

- **Técnico especialista en SIG o usuario**

Es el operador básico de todo programa informático. Ni el mejor programa ni el mejor ordenador del mercado, resultan útiles si el usuario no goza de una formación adecuada. Sin embargo es necesario señalar que cualquier técnico en GIS debe estar lo suficientemente formado para el correcto funcionamiento del sistema.

- **Datos**

La disponibilidad y exactitud de los datos influyen de manera directa en el resultado o análisis final. No debemos confundir datos con bases de datos (BBDD), correspondiendo esta última la mejor forma de almacenarlos y organizarlos. Es conveniente recordar que ArcGIS permite trabajar con BBDD ya instaladas.

- **Equipo informático o hardware**

La capacidad del procesador, la velocidad RAM, afectan a la velocidad de procesamiento del archivo de salida. Los SIG funcionan en un amplio rango de equipos centralizados o en configuraciones individualizadas o de red.

- **Software**

ArcGIS Desktop incluye un conjunto de módulos: ArcMap, ArcCatalog y ArcToolbox. Utilizando estos tres módulos, se puede realizar cualquier tarea SIG por complicada que resulte. Se puede acceder a ArcGIS utilizando tres productos software: ArcView, ArcEditor y ArcInfo. Los principales componentes de cualquier plataforma GIS, son:

- Sistema de manejo de bases de datos.
- Un sistema de Interfaz gráfica de usuario.
- Herramientas para la captura y manejo de la información.
- Herramientas para soporte de consultas, análisis y visualización de datos geográficos.

- **Procedimientos o métodos**

Para el correcto funcionamiento de un SIG es necesario un procedimiento o conjunto de los mismos correctamente definido, de forma que permita una retroalimentación para detectar posibles disfuncionalidades, como se muestra en la figura II.3.



Figura II.3. Componentes de un SIG

Fuente: <http://www.todosig.es/1-1-que-es-un-gis.html>

2.1.9. El dato geográfico

La potencia y funcionalidad de los SIG está basada en su capacidad para realizar operaciones de consulta, manipulación y análisis de datos geográficos.

Un dato geográfico, es decir el contenido de la información, se puede referenciar sobre una entidad geográfica (soporte del contenido) que representa una simplificación de un elemento real. Un dato geográfico tiene dos componentes, por un lado la componente espacial que se refiere a una entidad geográfica y por otro lado una componente temática que compone la parte descriptiva del dato geográfico (atributos y/o información asociada a la entidad geográfica). Por ejemplo, el dato geográfico de una provincia está compuesto por un componente espacial, la propia geometría de la provincia, y un componente temático que puede ser el nombre, población, área, etc., como se muestra en la figura II.4.



Figura II.4. Componente espacial – Componente temático

Fuente: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/749/1/00775tfc.pdf>

Esta característica del dato geográfico nos permite tratarlo de tres formas distintas:

- Análisis de la componente temática. De manera análoga a como se haría en una BD corriente.
- Análisis de la parte espacial. Es decir, analizando solo sus características topológicas (adyacencia, proximidad, etc.).
- Análisis de ambos componentes. Esta es la más habitual y la que caracteriza a los SIG.

Es necesario que la componente espacial de un dato geográfico se pueda relacionar con otras componentes espaciales de otros datos geográficos. Esto se consigue mediante la georeferenciación. Una vez que un dato geográfico está georeferenciada, podemos hablar de su relación con los datos geográficos de su alrededor. Esta relación se denomina relación espacial o topológica.

2.1.10. Modelos de representación

Un SIG ha de ser capaz de representar y almacenar entidades geográficas del mundo real mediante la representación y almacenamiento de las entidades gráficas en un sistema informático, es decir convertir los datos geográficos en registros discretos que puedan ser tratados informáticamente.

Existen dos formas de representar la información geográfica en un sistema informático, el modelo *ráster* y el modelo *vectorial*.

La diferencia entre uno y otro está en la manera de almacenar la información geográfica. Mientras que el modelo vectorial almacena las coordenadas de las formas geométricas que definen cada entidad, el modelo ráster almacena una matriz de posiciones que adoptan el valor de la entidad que se representa en cada posición, como se muestra en la figura II.5.

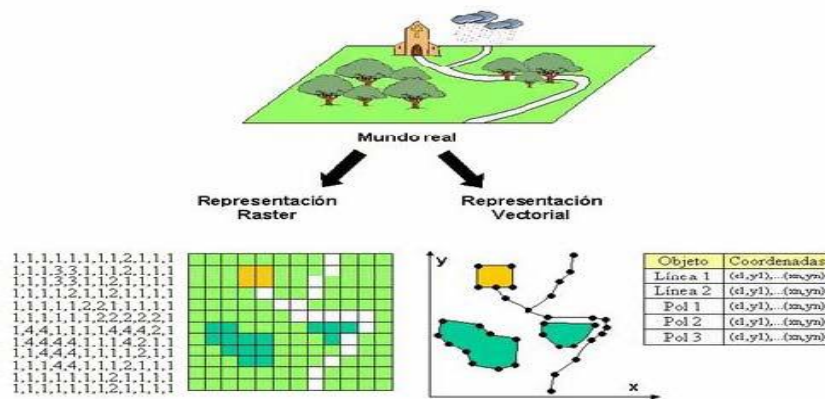


Figura II.5. Modelo ráster vs. Modelo vectorial

Fuente: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/749/1/00775tfc.pdf>

A la hora de elegir entre el modelo vectorial y el modelo ráster se ha de tener en cuenta que cada modelo se adapta particularmente bien a determinados escenarios, Por ejemplo, si la variable a modelizar se comporta de forma continua en el espacio se usa de forma habitual el modelo ráster. Sin embargo si la variable es discreta se usan ambos modelos no pudiéndose decir que uno sea mejor que otro de manera genérica.

La tabla II.I nos muestra una comparativa entre el formato vectorial y ráster:

Tabla II.I. Tabla de comparación modelo Vectorial vs. Modelo Ráster

Modelo Vectorial	Modelo raster
Ventajas	
La ocupación de espacio de memoria es menor	La captura de datos es más sencilla
La definición de entidades y el cálculo de magnitudes geométricas es más precisa	La estructura de datos es más simple
La representación de las relaciones topológicas es más óptima	La manipulación y gestión de la información resulta más sencilla
La representación gráfica está optimizada	
Inconvenientes	
La captura de datos es más costosa	La precisión en el cálculo de áreas y longitudes es menor
La estructura de datos es más compleja	La ocupación de espacio de memoria es mayor
La realización de ciertas operaciones (Comparación de mapas) es más dificultosa	La representación de ciertas relaciones topológicas es más dificultosa

2.1.11. Aplicaciones de los SIG

- Cartografía temática automatizada: mapa de temperaturas, mapa de precipitaciones, mapa de suelos.
- Estudios hidrológicos: ordenación de cuencas, cálculo de colectores.
- Diseño de infraestructura: redes de agua, saneamiento y electricidad.
- Ordenación territorial: espacios protegidos, Plan General de Ordenación

Urbana.

- Cálculo de mapas de peligro potencial: incendios, erosión, inundaciones.
- Estudio y valoración del paisaje: fragmentación, sinuosidad, diversidad.
- Valoraciones económicas y urbanísticas: zonas de expansión, rentas.
- Estudios de geomarketing y de viabilidad: grandes superficies comerciales, entidades financieras.
- Gestión cinegética: caza mayor, caza menor.
- Estudios de equipamiento social: centros de enseñanza, hospitales.

2.2. INFRAESTRUCTURA DE DATOS ESPACIALES

2.2.1. Introducción

Las Infraestructuras de datos espaciales definen un nuevo paradigma en la producción, y gestión de la cartografía. Tienen como objetivo el integrar a través de Internet los datos, metadatos, servicios e información de tipo geográfico, facilitando a todos los usuarios potenciales la localización, identificación, selección y acceso a tales recursos.

Las IDEs son producto de la iniciativa europea INSPIRE, y de acuerdo con las directrices marcadas por el Open GIS Consortium, siguen unos estándares que permiten tener disponibles en Internet una gran parte de la información geográfica generada por las entidades, tanto públicas como privadas que se dedican a la generación de cartografía, todo ello únicamente empleando exclusivamente un navegador.

En el ámbito de la infraestructura de datos espaciales, tienen especial importancia dos iniciativas, una a nivel Europeo y otra mundial. A nivel europeo encontramos la iniciativa Inspire y a nivel mundial el OGC.

Inspire

Inspire nació como uno de los principios de la Unión Europea para lograr una mayor transparencia de las administraciones públicas, así como una mayor participación y acceso a la información, especialmente en temas relacionados con políticas territoriales y ambientales. La Directiva europea INSPIRE, una importante referencia mundial en materia de IDE, establece que los datos deben cumplir una serie de requisitos como, por ejemplo, ser visualizados en un contexto apropiado y utilizados a través de interfaces sencillas y amigables que faciliten su comprensión e

interpretación, permitiendo así la disposición de información geográfica de forma que se permita la formulación, la implementación, la monitorización y la evaluación de las políticas de impacto o de dimensión territorial de la UE. En resumidas cuentas, esta iniciativa tiene como objetivo poner a disposición de los ciudadanos una gran cantidad de información geográfica producida por las administraciones públicas.

Open Geospatial Consortium

Es un consorcio sin ánimo de lucro fundada en 1994, formado por empresas de software GIS, administraciones públicas y universidades, que tiene como objetivo la “definición de estándares abiertos e interoperables dentro de los Sistemas de Información Geográfica”. Esto quiere decir, la definición de unos estándares de datos y servicios que cualquier software SIG pueda manejar, independientemente de cuál sea el fabricante del software. Esto facilita la interoperabilidad y el intercambio de información geográfica de unos softwares a otros, todo ello para un claro beneficio de los usuarios.

Dentro de los estándares de datos definidos por el OGC encontramos los ficheros .GML y entre los servicios encontramos los WMS y WFS, que denominaremos servidores de mapas.

Estos servidores permiten a un usuario conectado a internet “solicitar” y recibir información geográfica en formato imagen (*.jpg o *.png) para el caso del WMS, o en formato vectorial (*.GML) para el caso del WFS, a partir de los datos geográficos que existan en el servidor.

2.2.2. Definición

Una IDE o Infraestructura de Datos Espaciales es un sistema informático integrado por un conjunto de recursos como servidores, programas, datos, aplicaciones, recursos humanos, dedicados a gestionar Información Geográfica y que están disponibles en Internet a través de un geoportal, que deben cumplir una serie de condiciones de interoperabilidad, como normas, especificaciones, protocolos, interfaces, que permitan facilitar la producción, el acceso y uso de la información espacial regional, nacional o local para el apoyo al desarrollo social, económico y ambiental de los pueblos, haciendo uso solamente de un simple navegador de manera fácil, cómoda, eficaz, confiable y a bajo costo sin necesidad de que el usuario posea

grandes conocimientos técnicos.

2.2.3. Principios de una IDE

- Un principio fundamental de la IDE es que para que se pueda constituir como tal, las normas OGC indican que por lo menos deben cumplir con tres servicios que son el Web Map Service, servidor de catálogos (metadatos) y el Web Features Service, por medio de los cuales se debe acceder fácil, cómoda y eficazmente a los datos geográficos existentes.
- Otro principio importante es que la Infraestructura de Datos Espaciales es una colección de políticas, marcos o acuerdos institucionales, estándares, tecnología y factor humano que facilitan la disponibilidad y acceso a la información espacial.

2.2.4. Lugares de implantación de una IDE

Por lo general las IDE se establecen en un ámbito territorial, como:

- Estatal,
- Autonómico,
- Local, o incluso global.

Dentro de los ámbitos que se mencionaron, específicamente pueden implementarse en una empresa, un centro de investigación, un organismo oficial, como ayuda para la gestión de su propia información espacial, y también puede implantarse como servicio público y/o enlazando otras IDE.

Para el establecimiento de una IDE, ya sea a nivel local, regional, estatal o global, lógicamente se requiere del acuerdo de los productores, integradores y usuarios de datos espaciales del ámbito territorial en el que se establezca, para que se proporcionen los recursos adecuados para su puesta en marcha y mantenimiento, incentivar la producción y recogida de metadatos y asegurar la igualdad de presencia y la transparencia de acceso.

En resumen pretenderá responder a las preguntas que probablemente todos nos hayamos hecho alguna vez, tales como ¿Qué información geográfica hay?, ¿Dónde encontrarla?, ¿Quién la produce?, ¿Cuáles son sus características? o ¿Cómo obtenerla?

2.2.5. Objetivos IDE

Los objetivos son claros y ambiciosos:

- Facilitar el acceso y la integración de la información espacial, tanto a nivel institucional y empresarial como de los propios ciudadanos, lo que permitirá extender el conocimiento y el uso de la información geográfica y la optimización de la toma de decisiones;
- Promover los metadatos estandarizados como método para documentar la información espacial, lo que permitirá la reducción de costos y evitar la duplicación de esfuerzos; y animar a la cooperación entre los agentes, favoreciendo un clima de confianza para el intercambio de datos.

2.2.6. Razones para implantar una IDE

Hay muchas razones, entre las cuales se mencionara las de mayor peso:

- La información espacial existente está dispersa y no estandarizada.
- Existen problemas con la coincidencia espacial de las entidades geográficas registradas.
- No existen normas para la transferencia de los datos entre los participantes.
- Hay dificultades para la obtención de datos (por el nivel de acceso, por el precio, por las restricciones de los propietarios, etc.).
- No se cuenta con la suficiente integración en la tenencia de los datos y las bases cartográficas (incluyendo datos de sensores remotos).
- El universo de la información geoespacial² es abierto y variable. Los portadores de información son muchos, variados e impredecibles.
- No existe total identificación de los responsables de los datos, dentro y fuera de los subsistemas.

2.2.7. Componentes de una IDE

Para ello se consideran los siguientes componentes esenciales en una IDE, como se muestra en la figura II.6.

² **Geoespacial:** Es información sobre objetos o entidades abstraídos del espacio geográfico.

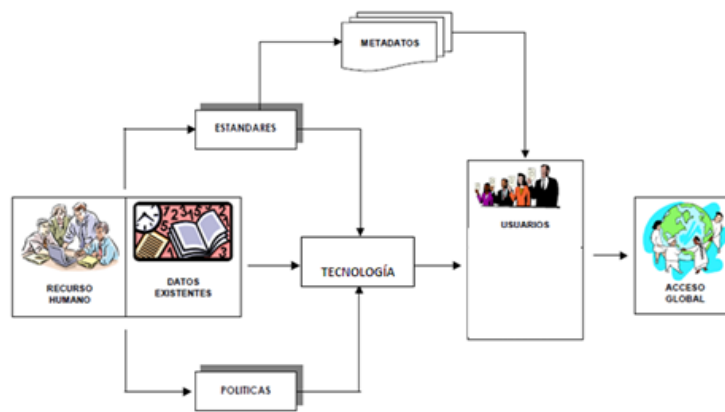


Figura II.6. Componentes de una IDE

Fuente: [http:// www.ccidep.gob.pe/archiv/doc/DocDifusion_IDEP.pdf](http://www.ccidep.gob.pe/archiv/doc/DocDifusion_IDEP.pdf)

Recurso humano

Es la participación de grupos de trabajo multidisciplinarios que elaboran las diferentes actividades y el desarrollo técnico de la IDE.

Para que la IDE pueda funcionar es necesario que los involucrados se comprometan a proporcionar los recursos necesarios o adecuados para poner en marcha el proyecto.

Datos

Son aquellos sin los cuales es imposible construir información lógica, consistente, exacta, racional e intercambiable. Deben permitir el análisis, y además deben cumplir con las normas y especificaciones declaradas para la información geográfica.

Estándares

Estos se constituyen en la piedra angular de la creación de la infraestructura de los datos espaciales, en virtud de que sin estándares se seguirá trabajando como islas con la información espacial.

Los estándares establecen normas que permiten ajustar la información geográfica, así como también intercambiar y compartirla con los diferentes agentes sin problemas.

En lo referente a la información espacial se estandariza las abstracciones que representan los objetos espaciales, es decir los diferentes elementos con sus atributos y procesos.

El aporte de los estándares está en:

- Proveer definiciones comunes para la información, de tal manera que se facilite su uso, conocimiento y automatización.
- Homogenizar los atributos y sobre todo los valores que estos puedan tomar.
- Homogenizar los procesos para las operaciones de captura, almacenamiento y presentación de la información.

Políticas

Las políticas promueven la creación y accesibilidad a datos esenciales.

Dado que es necesario que todos los agentes implicados usen unas referencias comunes de forma que se puedan manipular los datos sin ambigüedad, van a ser necesarias unas políticas concretas para asegurar la existencia de datos de referencia. Una operación tan sencilla como la de superponer dos capas de información exige que ambas estén referidas un mismo sistema geodésico y utilicen la misma proyección.

Tecnología

La tecnología es necesaria para el funcionamiento del sistema; para que la IDE funcione correctamente es imprescindible que la tecnología utilizada cumpla con las condiciones necesarias.

La tecnología WebMapping consiste básicamente en un sistema que permite la administración de datos a través de Internet, lo que posibilita el manejo, visualización y consulta de la información geográfica existente de manera remota.

Metadatos

Los metadatos consisten en información que caracterizan datos, brindando al usuario información acerca de las características de los datos publicados. Los metadatos son utilizados para suministrar información sobre esencia.

Su utilización describe la información geográfica, a través de las respuestas a las preguntas siguientes:

- ¿Qué describe el conjunto de datos?
- ¿Quién ha producido el conjunto de datos?
- ¿Qué finalidad tienen los datos?

- ¿Cómo se creó el conjunto de datos?
- ¿Por qué se creó este conjunto de datos?
- ¿Quién escribió los metadatos?
- ¿Fecha de la creación de datos?

La gestión de los metadatos por parte del geoportal se realiza a través de un Servicio Web llamado Catálogo de Metadatos.

Donde cabe destacar que su consulta se realiza mediante un formulario que tiene cada geoportal. El usuario debe poder preguntar por datos en función del momento de su creación o actualización, en función de su posición o en función de palabras clave de contenidos o de productor.

Servicios

Además de los datos y metadatos, las IDE cuentan con servicios que cumplen una serie de condiciones de interoperabilidad.

A continuación se mencionara algunos de los servicios que están definidos en el OGC, como se muestra en la figura II.7.

- CSW.
- WMS - Mapas (imágenes).
- WFS - Datos (fenómenos).
- WCS.
- Nomenclátor.

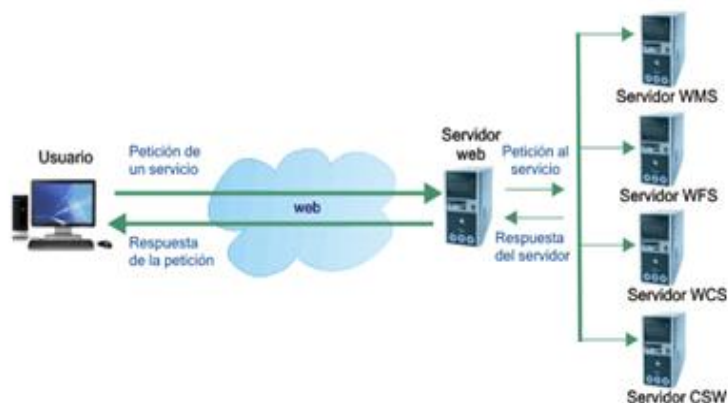


Figura II.7. Servicios IDE

Fuente: <http://idet.tucuman.gob.ar>

CSW - Servidor de catálogos

Es un servicio de búsqueda que ofrece mediante un formulario o plantilla la posibilidad de localizar conjuntos de datos geográficos (mapas, hojas topografías, fotos, ortofotos, documentos, etc) que hay disponibles a una escala determinada, de una zona particular, sobre un tema específico, o en una fecha o intervalo de fechas.

Como resultado de la búsqueda se permite: visualizar o descargar los metadatos, que describen el recurso encontrado; o ver el producto localizado en el visualizador (si existe un servicio de mapas disponible).

WMS - Servicio de mapas web

Es un servicio que produce mapas de datos espaciales en forma de un archivo de imagen digital conveniente para la exhibición en una pantalla de ordenador referidos de forma dinámica a partir de la Información Geográfica producida, que puede provenir de datos de un SIG, un mapa digital, una ortofoto³, una imagen de satélite, etc.

Los mapas producidos por WMS se generan normalmente en un formato de imagen como PNG, GIF o JPEG, y ocasionalmente como gráficos vectoriales en formato SVG⁴ o WebCGM⁵.

Las operaciones WMS pueden ser invocadas usando un navegador estándar realizando peticiones en la forma de URL⁶. El contenido de tales URLs depende de la operación solicitada. Concretamente, al solicitar un mapa, la URL indica qué información debe ser mostrada en el mapa, qué porción de la tierra debe dibujar, el sistema de coordenadas de referencia, y la anchura y la altura de la imagen de salida. Cuando dos o más mapas se producen con los mismos parámetros geográficos y tamaño de salida, los resultados se pueden solapar para producir un mapa compuesto. El uso de formatos de imagen que soportan fondos transparentes (e.g., GIF o PNG) permite que los mapas subyacentes sean visibles. Además, se puede solicitar mapas individuales de diversos servidores.

³ **Ortofoto:** Es un conjunto de imágenes aéreas tomadas desde un avión o satélite.

⁴ **SVG:** Scalable Vector Graphics

⁵ **WCGM:** Web Computer Graphics Metafile

⁶ **URL:** Uniform Resource Locators

El WMS tiene establecidas tres tipos de peticiones:

- **GetCapabilities:** investiga las capacidades del servidor de mapas interrogado mediante un mensaje XML. El servidor le devuelve la información mediante otro mensaje XML.
- **GetMap:** conociendo las capacidades del servidor, requiere una mapa mediante un mensaje XML y el servidor interrogado devuelve un mapa en formato ráster (PNG, JPEG, GIF). Estos mapas pueden superponerse al definir colores transparentes.
- **GetFeatureInfo:** sobre el mapa devuelto se puede interrogar al servidor remoto sobre información asociada a algún elemento (que se puede seleccionar, por ejemplo, mediante un clic sobre un pixel del elemento). Tanto la pregunta como la respuesta se vuelven a realizar mediante mensajes XML.

Como se puede apreciar, el WMS interroga y recibe información textual, pero además puede confeccionar un mapa ráster a partir de una interrogación y de la información espacial contenida en el servidor.

WFS - Servicios de fenómenos en web

Es un servicio estándar, que ofrece un interfaz de comunicación, el cual permite interactuar con los mapas servidos por el estándar WMS, facilitando el acceso, consulta y edición de datos y los atributos de un fenómeno geográfico como un río, una ciudad o un lago, en definitiva ofrece servicio WMS siguiendo criterios geográficos. Habitualmente los datos proporcionados están en formato GML que deriva del XML, que es el estándar a través del que se transmiten las órdenes WFS, pero cualquier otro formato vectorial puede ser válido. Un WFS permite no solo visualizar la información tal y como permite un WMS, sino también consultarla libremente y por lo tanto permite desarrollar todo tipo de aplicaciones de análisis de los datos: gestión de redes, caminos mínimos, análisis superficial, etcétera. Por añadidura, al dar acceso a todos los atributos almacenados, abre la puerta para la descarga de la información.

WCS - Servicio de coberturas en web

Pertenece a los servicios de la OGC. Este servicio permite obtener e intercambiar información geoespacial en forma de coberturas que corresponden a objetos de tipo

vectorial, ráster o modelos digitales además de no solo visualizar información ráster, como ofrece un WMS, sino además consultar el valor del atributo o atributos almacenados en cada píxel, y descargar el fichero completo en un formato estándar, como GML.

Nomenclátor

Se define como un servicio que a partir de la entrada de un nombre o de una característica de un fenómeno, devuelve la localización de un fenómeno geográfico, mediante unas coordenadas del fenómeno en cuestión. Adicionalmente, la consulta por nombre permite fijar otros criterios como la extensión espacial en que se desea buscar o el tipo de fenómeno dentro de una lista disponible (río, montaña, población, etcétera).

Una vez localizado un nombre, permite la visualización de su ubicación y del entorno geográfico que le rodea.

Geoportal

El acceso a la información y datos geográficos es esencial en las IDEs, es por aquello que por lo general se hace uso de un geoportal, el cual es un sitio web cuya finalidad es ofrecer a los usuarios el acceso a una serie de recursos y servicios basados en la información geográfica, permitiendo el descubrimiento, acceso y visualización de los datos geoespaciales, utilizando un navegador estándar, posibilitando la integración, interoperabilidad y el intercambio de información entre las diversas instituciones, colectivos profesionales, empresas de servicios, etcétera.

2.2.8. Componentes técnicos del geoportal

La figura II.8 muestra los componentes tecnológicos mayormente utilizados para la implementación de un geoportal son:

- Servidor de mapas: MapServer.
- Servidor web: Apache.
- Visor de mapas: OpenLayers, CartoWeb, entre otros.
- Base de datos espacial: PostgreSQL+PostGIS.
- Servidor de catálogo: Geonetwork.
- Cliente IDE avanzado: GVSIG, ArcGis.

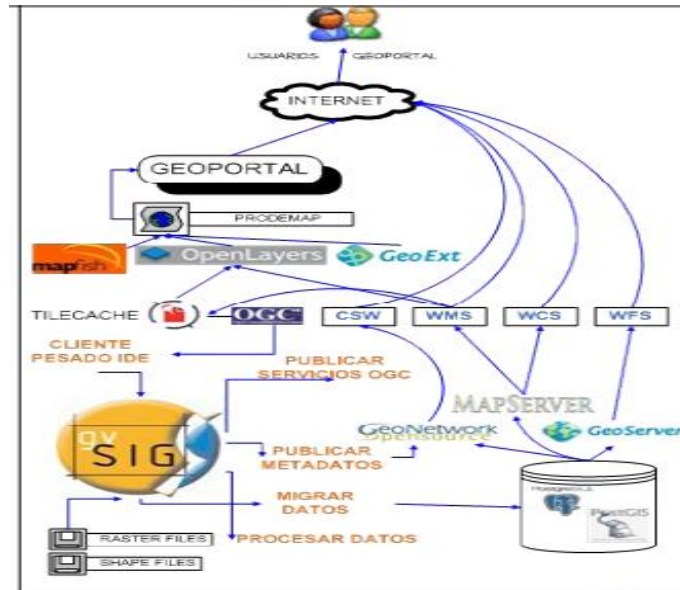


Figura II.8. Tecnología geoportal

Fuente: <http://dugi-doc.udg.edu/bitstream/handle/10256/3459/a5.pdf>

A continuación se menciona la definición de cada componente del geoportal.

MapServer

Definición

Es una herramienta para publicación de mapas en Internet. Generalmente MapServer funciona como una aplicación CGI⁷ de carácter libre, la cual corre bajo plataformas Linux/Apache, Windows 2000/2003, desarrollada por la Universidad de Minnesota para construir aplicaciones que sirvan mapas a través de Internet.

Su modo de funcionamiento está basado en la generación de lado del servidor web de imágenes estáticas (JPEG, GIF, PNG, etc.) como resultado del proceso de las peticiones realizadas por los clientes. Estas imágenes son referenciadas posteriormente dentro de la interfaz de usuario que se le envía al cliente (código HTML).

Características de MapServer

Entre las características de MapServer destacan las siguientes:

- Generación automática de leyendas.

⁷ CGI: Es un estándar que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web.

- Símbolo y color adaptable.
- Acceso en función de las características a datos sobre atributos.
- Salida cartográfica avanzada.
 - Dibujo capas de información dependiendo de la escala.
 - Dibujo de etiqueta evitando la colisión entre ellas.
 - Plantillas de salida totalmente personalizables.
 - Elementos del mapa automáticos, como son escala gráfica, mapa de referencia y leyenda.
- Distintas plataformas: Linux, Windows, Mac OS X, Solaris, y otros.
- Multitud de formatos de datos ráster y vector.

Librerías MapServer

Las principales librerías de software Libre, para la edición, manipulación y visualización de datos geográficos ráster y/o vectorial, que son necesarias para la explotación de MapServer son las siguientes:

- **GDAL/OGR** Es una biblioteca que permite acceder de forma transparente a una enorme cantidad de formatos de imagen ráster, para ser más específicos soporta más de 40 formatos ráster.
OGR: Es una librería Open Source para el procesamiento de datos vectoriales. GDAL se emplea para visualizar datos ráster o imágenes y OGR para los vectoriales.
- **PROJ4:** Librería de administración de proyecciones cartográficas.
- **GEOS:** Traducción de la librería JTS al lenguaje C.
- **GD:** Es una biblioteca para generar imágenes de forma eficiente, es requerida por el MapServer.

Funcionamiento de MapServer

El esquema simplificado de funcionamiento de MapServer se describe a continuación, como se muestra en la figura II.9.

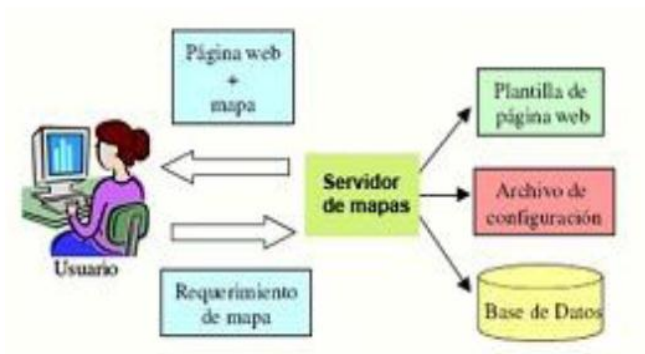


Figura II.9. Funcionamiento MapServer

Fuente: http://idesc.cali.gov.co/download/curso_fundamentos_ide/10.Manual_PDF_MapServer.pdf

El servidor recibe un requerimiento de un usuario, ya sea para presentar un mapa nuevo o realizar alguna acción sobre uno ya existente. Utilizando la información geográfica que se encuentra en la Base de Datos y un archivo de configuración que indica que capas debe presentar y de qué modo, se genera un nuevo mapa.

Luego utilizando una plantilla de página Web, resuelve el requerimiento retornando una página web y la imagen del mapa.

Arquitectura de las aplicaciones de MapServer

La figura II.10 muestra que una aplicación de MapServer consiste de:

- Un servidor HTTP.
- Mapserver CGI.
- Una fuente de datos SIG.
- Un archivo plantilla.
- Un archivo mapfile (.map).



Figura II.10. Recursos MapServer

Fuente:http://idesc.cali.gov.co/download/curso_fundamentos_ide/10.Manual_PDF_MapServer.pdf

Servidor HTTP

Proporciona las páginas HTML cuando son requeridas por el navegador del usuario. Se requiere un servidor como Apache o Microsoft Internet Information Server.

MapServer CGI

Es el archivo binario o ejecutable, que recibe solicitudes y retorna datos, imágenes, etc. que se encuentra en el directorio cgi-bin o scripts del servidor HTTP, por razones de seguridad no debe estar en la raíz del web y se denomina mapserv.

Fuente de datos GIS

La información o fuente de datos SIG que utiliza MapServer pueden ser de dos tipos.

- Los formatos de datos vectoriales que soporta son ESRI shapefiles como formato vector por defecto, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL y otros a través de la librería OGR.
- Los formatos de datos ráster soportados son: TIFF/GeoTIFF, EPPL7, ECW y muchos otros a través de la utilización de la librería GDAL.

El archivo plantilla

Controla como saldrán los mapas, leyendas y cualquier otra respuesta de desde MapServer hacia la página HTML, opera como cualquier otro archivo HTML excepto porque contiene celdas que pueden ser modificadas por el CGI de MapServer. El

archivo Plantilla permite al autor colocar el mapa y la leyenda en una página, y determina la manera en que el usuario interactúa con MapServer (navegar, consultar, hacer zoom, etc.). La información o fuente de datos SIG que utiliza MapServer es SHAPEFILE⁸ como formato vector por defecto, en formato ráster se puede utilizar archivos geoTiff y archivos Tiff.

Archivo MapFile

El archivo principal de configuración de MapServer es un archivo de texto, normalmente con extensión “.map”, en él se incluye una serie de parámetros que definen las capas disponibles en el servicio, el estilo con que se representarán, su simbología, formato en que se generará la imagen, el sistema de referencia, las consultas, etc.

El archivo .map consta de varias secciones. Cada sección se inicia con el nombre de la sección y termina con la palabra END. El contenido de las secciones consiste en la definición de determinados parámetros del tipo atributo - valor.

La sección principal es el objeto .map, la cual anida a otras secciones, como se observa en la siguiente figura II.11.

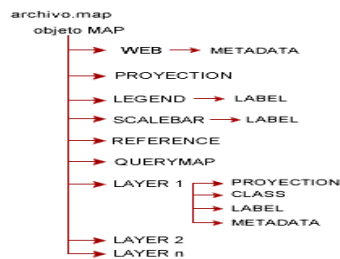


Figura II.11. Estructura archivo .map

- **Análisis del archivo mapfile (.map)**

El archivo mapfile (.map) es el mecanismo básico de configuración de MapServer, está hecho de "objetos" y cada objeto tiene una palabra clave u otros objetos. Cada objeto comienza con el nombre del objeto y termina con "END". El contenido del archivo mapfile es similar a lo siguiente:

⁸ **Shapefile:** Es un formato de almacenamiento digital que permite guardar datos correspondientes a la localización de los elementos geográficos así como sus atributos.

```
NAME Ejemplo # Comienzo del archivo mapfile
STATUS ON
SIZE 400 300
EXTENT 356829 9427649 1192815 10191151
UNITS METERS
SHAPEPATH "data"
IMAGECOLOR 255 255 255
FONTSET "fonts/fonts.txt"
LAYER #Inicio definición de capas
  NAME ecuador
  TYPE POLYGON
  STATUS ON
  DATA provi_ecuador
  CLASSITEM "pro_descri"
  LABELITEM "pro_descri"
  CLASS
  STYLE
  COLOR 240 240 240
  END # Style
  END # Class
END # Layer
END # Map File
```

A continuación se mostrara algunos ejemplos de los parámetros más importantes en una archivo .map.

Objeto MAP

La tabla II.II indica las palabras claves (parámetros) dentro del objeto MAP dividido por secciones, algunos de los cuales son obligatorios, mientras que otros son opcionales o tienen un valor asignado por defecto.

Tabla II.II. Parámetros objeto MAP

NAME	[name]. Nombre del archivo .map.
STATUS	[on off]. Establece si el mapa está activo o no.
SIZE	[int X] [int Y]. En píxeles ancho y alto de la imagen de salida.
EXTENT	[minX][minY][maxX][maxY]. Este parámetro especifica los límites geográficos del mapa generado, debe estar en las mismas unidades de los datos geográficos.
UNITS	[feet inches kilometers meters miles dd]. Unidades de las coordenadas del mapa, usado para el cómputo de la escala gráfica y escala numérica.
SHAPEPATH	[path]. Nombre del directorio donde se almacenan los datos geográficos. Esta es la ruta a los archivos de la capa.
IMAGECOLOR	[int R][int G][int B]. Establece color de fondo con el que se inicializará la imagen de salida.
FONTSET	[filename]. Nombre completo del archivo y directorio que contiene el conjunto de fuentes disponibles para usar.

Objeto REFERENCE

La tabla II.III define como será creado el mapa de referencia. Este es un mapa que comprende la extensión total y representa una marca en la zona que se visualiza actualmente, actualizándose interactivamente. También es posible realizar un clic en un determinado sector del mapa de referencia y MapServer generará el mapa de dicha zona.

```
REFERENCE
IMAGE imagenes/keymap.png
EXTENT 356829.92 9427649.64 1192815.51 10191151.79
SIZE 120 90
STATUS ON
COLOR -1 -1 -1 # Transparente
OUTLINECOLOR 255 0 0 # Rojo
END
```

Tabla II.III. Parámetros objeto REFERENCE

IMAGE	[filename]. Nombre completo del archivo de la imagen que será usada para generar el mapa de referencia.
EXTENT	[minX][minY][maxX][maxY]. Extensión espacial de la imagen de referencia.
SIZE	[int X][int Y]. Tamaño en píxeles de la imagen de referencia.
STATUS	[on off] on - el mapa de referencia será generado. off - el mapa de referencia no será generado. El valor por defecto es off.
COLOR	[int R][int G][int B]. Color en que se dibujará el recuadro de referencia. Para que dicho rectángulo o marca no se encuentre relleno deberá colocarse -1 -1 -1.
OUTLINECOLOR	[int R][int G][int B]. Color de la línea exterior del recuadro de referencia. Para no incluir línea exterior debe colocarse -1 -1 -1.

Objeto SCALEBAR

La tabla II.IV define como se construirá la escala gráfica.

```
SCALEBAR
STATUS ON
LABEL
SIZE SMALL
END
SIZE 300 5
UNITS kilometers
INTERVALS 5
END
```

Tabla II.IV. Parámetros objeto SCALEBAR

LABEL	Señal de comienzo de la sección LABEL en la que se definirá las características de Representación de las etiquetas. (color, tamaño, etc.)
SIZE	[int X][int Y]. Tamaño en píxeles de la escala gráfica. El etiquetado no está considerado dentro de estos valores.
UNITS	[feet inches kilometers meters miles]. Unidades de la escala gráfica.
INTERVALS	[integer] . Número de intervalos en que se dividirá la escala gráfica. Por defecto es 4.
STATUS	[on off embed]. on - la escala gráfica será generada.

Objeto LAYER

Para cada capa de información que contendrá la aplicación, deberá definirse un objeto LAYER. Puede especificar tantas capas como desee, como se muestra en la tabla II.V.

```
LAYER
NAME ecuador
TYPE POLYGON
STATUS ON
DATA provi_ecuador
CLASSITEM "pro_descri"
CLASS
STYLE
COLOR 240 240 240
END
END #class
END # layer
```

Tabla II.V. Parámetros objeto LAYER

NAME	[string]. Nombre corto para la capa. Este nombre es el vínculo entre el archivo map y la interfaz web, deben ser idénticos.
TYPE	[point line polygon circle annotation raster query]. Especifica como los datos podrían ser dibujados.
STATUS	[on off default]. Configura el estado actual de la capa.
DATA	[filename] [sde parameters] [postgis table/column] [oracle table/column]. Nombre completo del archivo de datos espaciales a ser procesado. Si se trata de archivos shapefile, no es necesario incluir la extensión. MapServer soporta formatos vector diferentes al formato shape a través del uso de la librería OGR (parte del paquete de software GDAL).
CONNECTION	[string]. Cadena de conexión a bases de datos para acceder a datos remotos.
CONNECTIONTYPE	Tipo de conexión. Por defecto es local. Este parámetro debe incorporarse en el caso que se desee incluirse una capa remota.
CLASS	Señal de comienzo del objeto CLASS.
CLASSITEM	[atributte]. Nombre del item en tabla de atributos a usar como filtro para aplicar el objeto CLASS.
LABELITEM	[atributte]. Nombre del item en tabla de atributos a usar como anotación.
TEMPLATE	Nombre del archivo plantilla. Página web visible por el usuario.

Objeto CLASS

La tabla II.VI define clases temáticas para las capas. Cada capa debe tener al menos una clase. A través del uso de expresiones puede darse distintos estilos a distintos atributos de una capa.

```

CLASS
  STYLE
    COLOR 240 240 240
  END # Style
END # Class
    
```

Tabla II.VI. Parámetros objeto CLASS

LABEL	Señal de comienzo del objeto LABEL.
STYLE	Señal de comienzo del objeto STYLE.

Objeto STYLE

La tabla II.VIII describe los elementos del objeto de estilo determinan cómo los símbolos serán presentados. Una clase puede contener El estilo múltiple se opone, que son aplicado en la secuencia.

```

STYLE
  SYMBOL "cuadrado"
  SIZE 8
  COLOR 232 121 0
  OUTLINECOLOR 0 0 0
END
    
```

Tabla II.VII. Parámetros objeto STYLE

SYMBOL	[int string filename]. Especifica el símbolo usado para las características de dibujo, identificado por el nombre o el número. El nombre está asociado con la palabra clave asociada en el archivo de definición de símbolos.
SIZE	[int N]. Establece el tamaño del objeto referido.
COLOR	[int R][int G][int B]. Establece el color usado para rellenar las características del objeto.
OUTLINECOLOR	[int R][int G][int B]. Establece el color del borde usado para los objetos. Los valores se encuentran en formato RGB.
SIZE	[int N]. Establece el tamaño del objeto referido.

Objeto LABEL

Es usado para colocar una etiqueta en el mapa, a partir de datos alfanuméricos. Comienza con la palabra LABEL y termina con END, como se observa en la tabla II.VIII.

```

LABEL
  COLOR 0 0 0
  MINDISTANCE 150
  MINFEATURESIZE 120
  SIZE SMALL
  TYPE BITMAP
  POSITION CC
END
    
```

Tabla II.VIII. Parámetros objeto LABEL

COLOR	[int R][int G][int B]. Color del texto o etiqueta a ser visualizada.
MINDISTANCE	[integer]. Define la distancia mínima en píxeles entre etiquetas.) en el que un elemento será etiquetado.
MINFEATURESIZE	[integer auto]. Especifica el tamaño mínimo (en el que un elemento será etiquetado.
POSITION	[ul uc ur cl cc]. Indica la posición donde se coloca el texto en relación con los puntos de etiquetado. ul - superior izquierda ; uc - superior centro ; ur - superior derecha; cc – centro.
SIZE	[integer][tyny small medium large giant]. Establece el tamaño del texto.
TYPE	[bitmap truetype]. Especifica qué tipo de fuente se va a usar.

Visor

Un visor sirve para la visualización de información espacial a través de los servicios web existentes en esta y en otras infraestructuras de datos espaciales es necesario la utilización de visores. Estos deben de ser capaces de consultar servicios estándares

de mapas cumpliendo los especificaciones de la OGC y normativas ISO⁹.

Los visores de información espacial se pueden clasificar en clientes ligeros o clientes pesados en función de si es posible su utilización y acceso mediante un navegador web (Internet Explorer, Firefox, Opera, Safari, etc) o es necesaria una instalación en local.

En la siguiente lista mencionaremos algunos visores de mapas libres para la web:

- OpenLayers
- CartoWeb
- Chameleon
- ka-Map. entre otros.

Base de datos (PostgreSQL y PostGIS)

Es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica, aportándole la capacidad a PostgreSQL de almacenar, consultar y manipular tipos de datos espaciales. Las funciones más avanzadas de PostGIS se basan en un conjunto de librerías GEOS. Permite manipular los datos directamente en la BD, empleando SQL.

GeoNetwork

GeoNetwork es una aplicación informática gratuita de Software Libre (código abierto), basada en estándares, desarrollada por la FAO¹⁰, de catalogación para recursos referenciados al espacio geográfico, permite un servicio de búsqueda remota, como el mencionado estándar CSW. Esta aplicación se conecta con la base de datos donde están almacenados los metadatos y los envía a las aplicaciones cliente cuando recibe las peticiones utilizando Internet.

gvSig

Es un programa informático libre con una interfaz amigable desarrollado en lenguaje

⁹ ISO: Organización internacional para la estandarización.

¹⁰ FAO: Organización para la Agricultura y la Alimentación.

de programación java, orientada al manejo de la información geográfica manejando datos tanto vectoriales como ráster.

2.3. CARTOWEB

Trabajar con MapServer, si bien es simple, se vuelve medio complejo y muy limitado para trabajos grandes porque todo está en el .map. La mayor desventaja está en que todo se hace de a pie, como definir cada color en una tematización, y por esto surgen herramientas como CartoWeb, que trabaja sobre MapServer pero trae plugins contruidos.

Cartoweb3 es una solución producida por CamptoCamp y está disponible desde el 30 de marzo 2005 en software libre de licencia GPL¹¹. Es un software bastante complejo.

Se basa en la cartografía del motor UMN¹² MapServer y se puede aprovechar la gran mayoría de formatos de datos espaciales (ráster y vector) corrientes. También da la posibilidad de conectarse a bases de datos geográfica de PostgreSQL / PostGIS Spatial. Muchas de las funciones y los módulos están pre-desarrollados, y la gestión de los flujos de datos. Estos artículos se pueden personalizar e integrar en un proyecto personal.

2.3.1. Características

A continuación se pueden mencionar las características potencialmente explotables:

- El acceso a muchos formatos de datos raster: TIFF (File), GeoTIFF, GIF, PNG, JPEG.
- El acceso a muchos formatos de datos vectoriales: Shapefiles, etc.
- El acceso a bases de datos espaciales: PostgreSQL / PostGIS.
- Compatible con Open Geospatial Consortium WMS, WFS, WCS.
- Características avanzadas de visualización modo de DHTML¹³.
- Navegación: zoom, movimiento.

¹¹ **Licencia GPL:** Es una licencia publica General que está orientada principalmente a proteger la libre distribución, modificación y uso de software.

¹² **UMN:** Universidad de Minnesota.

¹³ **DHTML:** Es un término que describe el arte de hacer páginas web dinámicas e interactivas, combina html, JavaScript y CSS.

- Exportación a formato CSV, PDF.

2.3.2. Arquitectura

El motor de CartoWeb se basa en la UMN MapServer. Las interacciones que se dan entre CartoWeb y MapServer son gestionados por el módulo PHP MapServer / MapScript.

- **Modularidad**

La aplicación fue construida como un conjunto de ladrillos que interactúan entre sí. Cada módulo o plugins lleva a cabo un grupo de tareas especiales, tales como la exploración geográfica, niveles de gestión, las propiedades de las imágenes, los permisos de usuario, los objetos de consulta, la exportación y mucho más. Sus funcionalidades están empaquetadas en conjuntos lógicos llamados plugins, que aspiran a ser independientes entre sí, aunque algunas dependencias no se pueden evitar totalmente. Algunos plugins (complementos del núcleo) no pueden ser desactivados, mientras que los otros deben ser explícitamente cargado en el servidor y / o archivos de configuración del cliente.

- **Proyectos**

Los proyectos se utilizan para personalizar una aplicación CartoWeb a sus necesidades, separando claramente los archivos principales de los proyectos específicos. Al crear un nuevo proyecto el desarrollador puede invalidar las plantillas, los archivos de recursos (imágenes, hojas de estilo, archivos JavaScript, etc), archivos de configuración e incluso añadir nuevos plugins o modificar los existentes.

Se recomienda utilizar los proyectos al implementar una aplicación con CartoWeb. La razón principal es la necesidad de mantener los archivos originales sin cambios con el fin de aplicar fácilmente las actualizaciones de las aplicaciones sin problemas.

Los proyectos son en realidad una colección de espejos de directorios y archivos de la arquitectura original. Los archivos colocados en un proyecto en la mayoría de los casos se utilizan preferentemente a los archivos originales.

En un proyecto nuevo no es necesario duplicar la estructura CartoWeb. Sólo tienen que ser creados los directorios que contienen archivos fundamentales.

2.3.3. Estructura del directorio de Global

Después de la instalación, CartoWeb tiene la siguiente estructura de directorios, como se muestra en la figura II.12.

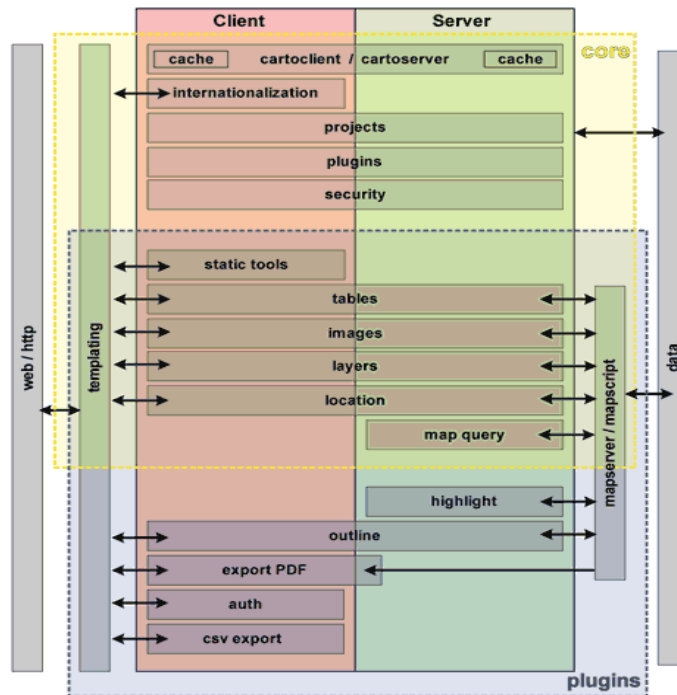


Figura II.12. Estructura Plugins
Fuente: <http://cartoweb.org/documentation.html>

- client: Los archivos del cliente.
- client_conf: Los archivos de configuración del cliente(client.ini.in)
- common: Contiene el código y los archivos comunes que tienen el servidor y el cliente.
- coreplugins: Estos plugins son inseparables e indispensables para el funcionamiento de CartoWeb. Su pertenencia a Cartoclient con CartoServer simplemente define su nivel de acción.
 - images: Este plugin define el formato de visualización de imágenes en la pantalla.
 - layers: Describe la estructura de capas de información.
 - location: Este parámetro interviene en la exploración y navegación en el mapa (zoom, pan).
 - mapquery: Realiza consultas basadas en un conjunto de Identificación del objeto seleccionado.

- query: Realiza consultas sobre capas.
- htdocs: Directorio del diseño web accesible
 - css: css
 - gfx: Los archivos de iconos
 - js: Los archivos de javascript
- include: Librerías usadas por CartoWeb.
- log: Registros utilizadas principalmente para el desarrollo y fines de depuración.
- plugins: Son completamente independientes y se puedan aplicar o no, según sea necesario. pero no es obligatorio
 - exportHtml: Plugin de exportación a HTML, permite exportar los mapas en formato HTML.
 - exportPdf: Plugin de exportación a PDF.
 - outline: Autoriza a las líneas de dibujo, puntos o polígonos en el mapa.
- projects: Proyectos CartoWeb.
- scripts: Guiones de mantenimiento y administración, contiene secuencias de comandos. Existe un script que gestiona la limpieza de los datos temporales.
- server: Archivos de código que se ejecutan en el lado del servidor.
- server_conf: Archivos de configuración del lado del servidor CartoWeb, contiene el archivo (server.ini) y está vinculado con un archivo de mapa.
- templates: Contiene el diseño específico de la plantilla.
- templates_c: Plantillas Smarty de los archivos almacenados en caché
- tests: Unidad de pruebas CartoWeb, utilizadas principalmente para fines de desarrollo y depuración.
- www-data: Este directorio web contiene subdirectorios con archivos temporales generados durante el uso de cartoweb, que luego pueden ser accedidos y modificados.
 - icons: Iconos de creación.
 - images: Imágenes de MapServer.
 - mapinfo_cache: Cache de configuración del lado del servidor, que contiene información del cliente.
 - mapresult_cache: Contiene las solicitudes del cliente y los resultados asociados con el servidor caché.
 - pdf: Contiene archivos de caché acerca de PDF generados.

2.3.4. Funcionamiento general

- El envío de la solicitud por el navegador al servidor, en este caso un servidor Apache.
- Apache analiza la solicitud y lo redirige a CartoWeb.
- CartoWeb procesa la información a través de una variedad de secuencias de comandos de php5, que se organizan en los plugins.
- Desde la solicitud enviada por el cliente, CartoWeb construyó su propia queja de MapServer.
- Para los scripts se pueden ejecutar como módulo de Apache utiliza php5.
- Una serie de funciones que se utilizan en secuencias de comandos CartoWeb, destinados específicamente para la comunicación con MapServer. Estas funciones son parte de la biblioteca de funciones MapScript.
- Funciones de MapScript y se interpretan para reflejar la consulta MapServer producido por CartoWeb.
- MapServer recibe la consulta, analizar y responder al visitar Mapfile. - El archivo de asignaciones de archivos es algo así como un índice, por lo que MapServer sabe qué tipo de información que están disponibles en los datos y sabe lo que son. Con Mapfile, MapServer accede a los datos y luego recupera los solicitados por la consulta.
- Después de este primer paso que nos muestra cómo se lleva a cabo el acceso a los datos, en los siguientes pasos se mencionara cómo estos datos se procesan y se devuelve al cliente.
- Los datos son recuperados y devueltos al servidor MapServer. Modos de comunicación entre MapServer y datos, difieren dependiendo de la naturaleza de los datos (formatos de capas o imágenes), que se definen en el archivo map de MapServer adoptar y consecuencias en "el camino a seguir".
- Una vez que se recuperan los datos, MapServer trata de producir imágenes y metadatos.
- Las imágenes así producidas, y la información correspondiente se recupera por CartoWeb a través del uso, de nuevo, las funciones MapScript.
- CartoWeb procesa la información sobre la base de la solicitud original enviado por el cliente y enviar el conjunto a un subconjunto de CartoWeb que organiza los datos en la vista de la pantalla.
- Antes de devolver al cliente, los scripts php se interpretan para generar

páginas HTML. Su diseño está dirigido por un conjunto de archivos llamado plantillas, posiblemente acompañados de archivos CSS (estilos y el formato). Finalmente funciones javascript se puede unir a la página para hacerla más dinámica.- Por último, el conjunto devuelto por Apache para el cliente.

2.3.5. Lenguajes de programación utilizadas

Php

Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor, es un lenguaje de código abierto e independiente de la plataforma, es muy popular especialmente adecuado para desarrollo web y que puede ser incrustado en HTML.

Uso

CartoWeb usa clases php que se utilizan para realizar una acción específica ya sea la navegación por el mapa, consultas, interfaz de búsqueda entre otros.

JavaScript

JavaScript es un lenguaje de programación utilizado para crear pequeños programitas encargados de realizar acciones dentro del ámbito de una página web.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Uso

La acción de JavaScript en CartoWeb, es que por una lado crea efectos especiales sobre la plantilla, además de crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo. Por el otro, JavaScript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas.

Css

Es la tecnología desarrollada por el World Wide Web Consortium (W3C), con el fin de separar la estructura de un documento de su presentación escrito en HTML o XML.

Tiene una sintaxis muy sencilla, que usa unas cuantas palabras claves tomadas del inglés para especificar los nombres de sus selectores, propiedades y atributos.

Una hoja de estilos CSS consiste en una serie de *reglas*. Cada regla consiste en uno o más *selectores* y un *bloque de estilos* con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato propiedad: valor.

Uso

Las hojas de estilos son utilizadas en la presentación de la plantilla del visor, ya que ahorran tiempo porque definen el aspecto de visualización de una etiqueta, con la hoja de estilo se puede lograr con una sola línea de código que todos los encabezamientos <h1> tengan un color o un tamaño determinado. O sólo un <h1> determinado. Pero si no se usara hojas de estilo, se tendría que definir cada vez que se use ese <h1>. Del mismo modo, si ese cambio que realizamos pasado un tiempo no nos gusta y lo queremos de otra forma. Con cambiar esa línea se aplicará en cascada a todos los documentos, mientras que de la otra manera implicaría un trabajo bastante más complicado.

2.3.6. Desarrollo una aplicación CartoWeb

Son 5 archivos en teoría los que deberían ser suficientes para definir una aplicación básica:

1. Client.ini
2. Server.ini
3. Mapfile.map
4. Mapfile.ini
5. Layer.ini

Los archivos de configuración. Ini - client.ini y server.ini

Los archivos de configuración (.ini) son los principales intermediarios para personalizar una aplicación. Hay uno para cada plugin, más uno para CartoServer y CartoClient, y otro para el archivo de asignaciones. Están ubicados en /project/prototipo/client_conf/, para los archivos de configuración de plugins que se ejecutan en el lado del cliente, y en /server_conf/prototipo/, archivos de configuración para los plugins que se ejecutan del lado del servidor y el archivo server.ini está dentro de /server_conf/.

1. Client.ini

Este archivo se encarga del funcionamiento de Cartoclient.

Los controles principales de este archivo son los siguientes:

- CartoserverDirectAccess = True/False: true significa que no utiliza el modo de SOAP, esto requiere que CartoServer y Cartoclient estén en la misma máquina.
- cartoclientUrl = URL: camino a la Cartoclient (client.php acceso a los archivos).
- cartoserverUrl = URL: camino a la CartoServer (cartoserver.wsdl.php acceso a los archivos).
- mapid = String: especifica el archivo de asignaciones a utilizar, es el nombre especificado por el parámetro NAME en el archivo de asignaciones (prototipo.map).

2. Server.ini

El archivo server.ini, contiene lo siguiente:

- imageUrl = cadena: ruta donde cartoserver genera imágenes que pueden ser accedidas.
- reverseProxyUrl = cadena: la URL del proxy inverso, en caso de ser usado.

3. **myMap.map**

Es un archivo de asignaciones donde se describen todas las capas, su simbolización y muchas otras cosas.

Un proyecto CartoWeb debe tener un archivo de asignaciones, conjuntamente con un archivo de configuración .ini. Es conveniente que el archivo de asignaciones y el proyecto lleven el mismo nombre.

4. **myMap.ini**

El archivo que contiene la información de configuración relacionada con un mapa o capa, es el mymap.ini, se encuentra en el mismo directorio que el archivo de asignaciones, pero tiene una extensión .ini. Estos archivos están en el directorio /server_conf/prototipo, este debe tener el mismo nombre que el archivo .map.

- mapInfo.loadPlugins = lista: lista de plugins del lado del servidor para cargar. Tenga en cuenta que la mayoría de los plugins del servidor también tienen un plugin de lado del cliente correspondiente, que debe estar cargado de igual manera en el lado del cliente.
- mapInfo.initialMapStates: [...].

MapStates iniciales

Inicia los Estados de un mapa, estableciendo los aspectos iniciales de la interfaz de selección de capas cuando se inicia con CartoClient.

Varios estados iniciales del mapa se pueden crear en myMap.ini, pero al menos una debe estar presente. Cada uno se identifica por un initialMapStateld único.

Propiedades disponibles y la sintaxis de las capas de "estados iniciales" son los siguientes:

- mapInfo.initialMapStates.initialMapStateld.layers.layerId. selected = true | false: true, si la capa se seleccionó inicialmente.
- mapInfo.initialMapStates.initialMapStateld.layers.layerId.development = true| false: true, el layerGroup se representa como un nodo desplegado (subcapas son visibles).
- mapInfo.initialMapStates.initialMapStateld.layers.layerId. hidden = true| false: si

es cierto, esta capa y sus hijos no se muestran en la lista de capas (pero todavía se muestran en el mapa si están activadas).

- `mapInfo.initialMapStates.initialMapStateId.layers.layerId.blocking = true| false:` true, esta capa y sus subcapas también están en la lista del árbol, pero sin casilla de verificación. Su estado de selección (que se define como "seleccionado") no se puede cambiar.

5. Layers.ini

Existen dos ejemplares de este archivo, uno configura el plugin para el lado CartoClient (ubicado en `/client_conf/`) y otro para el lado de CartoServer (ubicado en `/server_conf/prototipo`). Le permite configurar la estructura de las capas geográficas.

Del lado del cliente

Del lado del cliente hay un solo comando disponible.

- `ApplySecurity = True/False:` La gestión activa de la seguridad en capas de información geográfica.

Del lado del servidor

Hay dos tipos de capas en `layers.ini`: `Layers` y `LayerGroups`.

Layers

Tienen una correspondencia 1-a-1 a las capas de MapServer (como se define en la `layers.map`).

Aquí es la sintaxis para los parámetros de configuración diferentes de una capa.

- `layers.layerId.className = layer:` define el objeto como una capa; `layerId` es una cadena que identifica el objeto. Las normas generales de la sintaxis de un archivo ini deben ser respetados en la elección de la `layerId` (por ejemplo, no "-" se les permite).
- `layers.layerId.msLayer = cadena:` nombre de la correspondiente capa de MapServer en el archivo de asignaciones.
- `layers.layerId.label = cadena:` título de la capa en el árbol de capas en el cliente.
- `layers.layerId.icon = nombre:` nombre de la imagen estática que ilustra esta

capa en el árbol de capas. El archivo se espera que esté ubicado en /server_conf/prototipo/icons.

LayerGroups

Tiene subcapas u otros LayerGroups.

Parámetros de configuración de LayerGroups.

- layers.layerId.className = LayerGroup: define el objeto como un LayerGroup; layerId es una cadena que identifica el objeto. Las normas generales de la sintaxis de un archivo ini deben ser respetados en la elección de la layerId (por ejemplo, no "-" se les permite).
- layers.layerId.children = lista de layerIds: lista separada por comas de layerIds, estas subcapas pueden ser capas u otros LayerGroups.
- layers.layerId.aggregate = true | false: si es cierto, las subcapas no figuran en el árbol y no se pueden seleccionar individualmente. El valor predeterminado es falso.
- layers.layerId.label = cadena: título de la capa en el árbol de capas en el cliente.
- layers.layerId.icon = nombre: nombre de la imagen estática que ilustra esta capa en el árbol de capas. La ruta es relativa al myMap.ini.
- layers.layerId.rendering = tree | bloque | Radio | desplegable: indica cómo mostrar las subcapas.

2.4. OPENLAYERS

OpenLayers es un cliente Web-GIS ligero construido con clases de JavaScript, sin dependencia de servidores de mapas concretos. Ofrece una interfaz de usuario simplificado que hace uso de los servicios WMS de forma transparente para el usuario y desarrollador.

OpenLayers hace fácil poner un mapa dinámico en cualquier Web page. MetaCarta, la versión inicial de OpenLayers le dio al público el uso de la información geográfica de todas las clases. OpenLayers es un proyecto del Open Source Geospatial Foundation (OSGeo). Es JavaScript totalmente libre, lanzado bajo licencia BSD¹⁴.

¹⁴ **BSD:** Licencia de distribución de software libre flexible.

¡Para Los Desarrolladores!

OpenLayers es un conjunto de librerías escritas en varios lenguajes entre los que sobresalen JavaScript, XML, CSS los cuales permiten tener un producto de calidad y que permite exhibir datos del mapa en la mayoría de los browsers modernos de la web, sin dependencias del servidor. OpenLayers implementa JavaScript API¹⁵ para construir aplicaciones web para uso geográfico, similar al Google Maps y MNS Virtual Herat APIs, con una diferencia importante. OpenLayers es software libre, desarrollado para y por la comunidad del software de código libre.

La API JavaScript tiene objetos que permiten manejar todo tipo de capas (kml, geoJSON, wms,etc) y trabajar con eventos (hacer clic en el mapa).

OpenLayers implementa los métodos industry-standard para el acceso geográfico de los datos, tal como el OpenGIS Consortium's Web Mapping Service (WMS). OpenLayers se escribe en JavaScript orientado a objetos, usando componentes de Prototype.js y de la biblioteca de Rico. La base del código de OpenLayers tiene ya centenares de pruebas de unidad, a través de Test.AnotherWay framework.

Lenguajes imprescindibles para el desarrollo web con OpenLayers son: HTML, CSS y JavaScript que hace que las páginas web sean dinámicas con el objeto de mostrar una interfaz atractiva y novedosa para el usuario.

Las clases básicas de OpenLayers son:

Map.js: Es el objeto central de OpenLayers y contiene a todos los demás

Layer.js: Cada capa hereda los métodos básicos de esta capa

Control.js: Controles del Mapa. Tienen una relación 1..1 con los handlers y generalmente son los elementos que se suelen personalizar.

Handler.js: Son los manejadores de eventos. Están asociados a los eventos típicos de la web como son: "Click", "MouseWheel", "Hover"...

2.4.1. Características

Entre las principales características se puede mencionar las siguientes:

- Facilidad de uso.

¹⁵ **API:** Interfaz de programación de aplicaciones.

- Facilidad al momento de llamar al servidor de mapas ya sea de tipo local o algún servidor externo como google map, yahoo map entre otros.
- Permite la navegación de un mapa dentro una página web/wap.
- Visualiza hasta un punto mínimo y máximo del mapa, mediante la manipulación del zoom.
- Utiliza capas de tipo shp que están dentro de archivos (.map) y lo trata con si fuesen marcas.
- Brinda facilidad de utilizar varios objetos o capas de diferentes servidores y los muestra en una sola aplicación.

2.4.2. Arquitectura

OpenLayers se basa en la UMN MapServer ya que la aplicación realiza una petición al servidor Mapserver y se ejecutara los archivos .shp que están especificados en un .map y también permite cargar mapas de un servidor externo como google map, en la figura II.13 se muestra la arquitectura de OpenLayers.

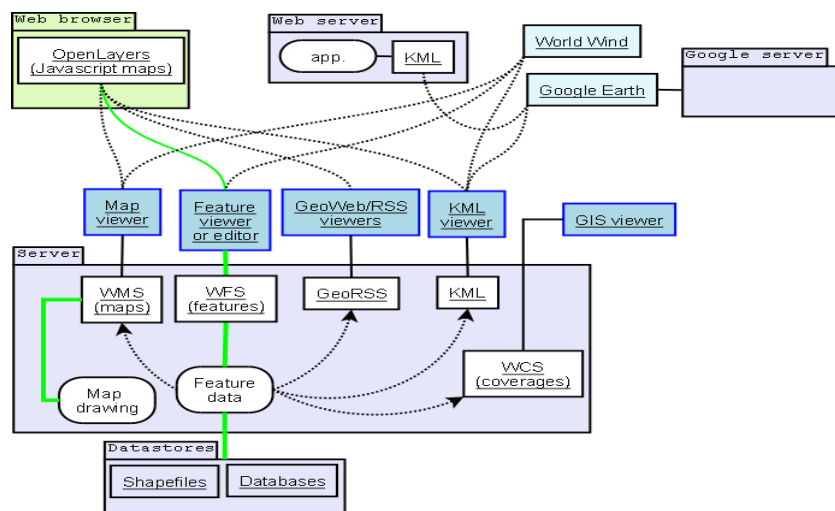


Figura II.13. Esquema representativo de la arquitectura de OpenLayers

Fuente: http://hierro.lazos.cl/lazos_cl/fileadmin/gestorarchivos/1248897960tesisSIGFinal.pdf

2.4.3. Ventajas de OpenLayers

La aparición de GoogleMaps, y sobre todo de la manera de utilizarlo gratuitamente en otras páginas mediante programación, ha revolucionado definitivamente el panorama geoespacial.

Los puntos más destacados del proyecto de GoogleMaps son:

- El buen diseño y documentación de su API.
- La cobertura, detalle y velocidad de sus mapas. La velocidad 'google', soportada por una infraestructura hardware inmensa, se ha convertido en la velocidad estándar requerida por cualquier usuario.
- La amplitud de la comunidad generada a su alrededor (con abundantes ejemplos en páginas, foros y blogs).
- Su creciente capacidad de integración y sinergia con otros servicios y tecnologías (del propio Google¹⁰ o de terceros).

Estos factores lo han convertido en la primera opción de manejo de información espacial para muchos individuos y organizaciones en todo el mundo. Entonces ¿por qué utilizar otra opción como OpenLayers?, en la tabla II.IX se muestra las ventajas de OpenLayers y googleMaps.

Tabla II.IX. Ventajas de OpenLayers frente a GoogleMaps

OpenLayers	GoogleMaps
Mayor flexibilidad para mapa base: GoogleMaps, BingMaps, OpenStreetMap, WMS...	Sólo capas base propias (callejero, satélite, híbrido y relieve). ¹¹
Implementación estándares OGC (WMS, WFS...)	Tecnología propia de Google
Mejor manejo de entidades vectoriales y estilos.	-
OpenSource, bajo osGeo y con licencia BSD ¹²	Gratuito pero no abierto, con código fuente no accesible.
Sinergias otros software libre como Geoserver.	-

2.4.4. Tipos de capas que se puede cargar con OpenLayers

OpenLayers trabaja, a diferencia de otros APIs, con un amplio número de orígenes de datos (servicios de mapa comerciales, servidores de mapa open source, servidores propietarios), y además lo hace siempre que sea posible de la mano de los estándares OGC, por lo que proporciona mucha flexibilidad al programador.

En OpenLayers existe el concepto de capa base, que es la capa principal de fondo activa, de la cual se toma el sistema de proyección y los niveles de zoom. Sobre ésta pueden añadirse más capas simultáneas, a modo de overlay, sean ráster o

vectoriales. Existen muchos tipos de capas, como se muestra en la tabla II.X.:

Tabla II.X. Tipos comunes de capas usados en OpenLayers

Capa	Descripción
Google	Raster. EPSG 900913 ¹⁷ . Varias opciones: satélite, mapa carreteras, híbrido y relieve.
Bing	Raster. EPSG 900913. Opciones: satélite, mapa carreteras e híbrido. Nota: hasta la v2.10 sólo está disponible el tipo de capa VirtualEarth (sin acceso directo a la caché de mapas de Microsoft).
WMS	Raster. EPSG variable. Permite seleccionar las capas del WMS.
OpenStreetMap	Raster. EPSG 900913.
GML	Vectorial. EPSG variable.
Vector	Vectorial. EPSG variable. Permite cargar formatos como GeoJSON, KML...

La tabla II.XI presenta un resumen de OpenLayers.

Tabla II.XI. Resumen OpenLayers

Datos			
Nombre	OpenLayers	Sitio Web	http://openlayers.org/
Lenguaje de desarrollo	Javascript	Versión estable	2.7
Plataformas soportadas	Windows, GNU/Linux, Mac OsX, BSD,UNIX	Licencia	Libre: BSD
Promotores principales	MetaCarta		
Soporte			
Control de versiones	SVN http://svn.openlayers.org/	Lista de Correo	Activa
Documentación de usuario	Completa	Documentación para desarrolladores	Completa

2.4.5. Manejo de herramientas

2.4.5.1. Layer Switcher

Nos permite pasar de una capa a otra u ocultar o mostrar resultados. Se maneja dos categorías: BaseLayer y Overlays.

BaseLayer: capas que son mutuamente excluyentes, por lo cual solo se puede visualizar una a la vez.

Overlays (Superposición): capas que no son mutuamente excluyentes. OpenLayers puede mostrar muchas overlays al mismo tiempo.

LayerSwitcher nos permite pasar de BaseLayer a Overlays o viceversa, tomando en cuenta que un Layer es por defecto un Overlays. La ventana que contiene estas categorías puede ser maximizada o minimizada siendo esta última, la opción por defecto al cargar el controlador.

El usuario realiza la petición para seleccionar una capa el sistema responde de la siguiente forma: La aplicación realiza una petición al servidor MapServer y se ejecutará los archivos .shp y responderá a la petición del sistema, mientras que el sistema devolverá la petición que realizó el usuario y mostrará los resultados, como se muestra en la figura II.14.

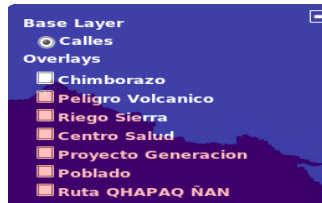


Figura II.14. Layer Switcher

Fuente: http://wiki.osgeo.org/wiki/Openlayers:_las_herramientas_disponibles_por_defecto#LayerSwitcher

2.4.5.2. Pan Zoom Bar

Nos permite manipular un mapa por medio de una barra de zoom, permitiendo ampliar o reducir el mismo a través de ZoomIn y ZoomOut.

El usuario realiza una petición al sistema para utilizar el zoom, la aplicación verifica y envía la respuesta de la petición para mostrar los resultados. El procedimiento sucede con la navegación dentro del mapa, se debe realizar la petición a la aplicación y el mostrará los resultados como respuesta de la petición, como se muestra en la figura II.15.

Aplicaciones

- Aumenta o disminuye el zoom (+, -).
- Permite desplazarnos en el mapa a nivel de zoom más cercano (slider).
- Desplazamiento del cursor con un zoom determinado.



Figura II.15. Pan Zoom Bar

Fuente: http://wiki.osgeo.org/wiki/Openlayers:_las_herramientas_disponibles_por_defecto#PanZoomBar

2.4.5.3. Overview Map

Nos permite tener un mapa con información general o relacionada a otro mapa. El mapa general se ubica en la parte inferior derecha del mapa con la opción a ser expandido.

En este mapa superpuesto (Overview) puede copiarse una capa base, de manera que este se convierta en el mapa principal.

Se debe tomar en cuenta que la capa de base para el mapa general debe estar en la misma proyección.

Cabe recalcar que OverviewMap no es mapa sino un control del mismo, como se muestra en la figura II.16.



Figura II.16. Overview Map

Fuente: http://wiki.osgeo.org/wiki/Openlayers:_las_herramientas_disponibles_por_defecto#OverviewMap

2.4.5.4. Mouse Toolbar

Permite seleccionar la forma de utilizar el mouse entre la navegación y el zoom.



Figura II.17. Mouse Toolbar

Fuente: http://wiki.osgeo.org/wiki/Openlayers:_las_herramientas_disponibles_por_defecto#MouseToolbar

2.4.5.5. Scale Line

Permite agregar una escala grafica al mapa, como se muestra en la figura II.18.

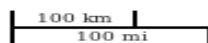


Figura II.18. Scale Line

Fuente: http://wiki.osgeo.org/wiki/Openlayers:_las_herramientas_disponibles_por_defecto#ScaleLine

2.4.5.6. Mouse Position

Permite visualizar en el mapa las coordenadas correspondientes a la ubicación del mouse, como se muestra en la figura II.19.

400627.01484, 10219376.62257

Figura II.19. Mouse Position

Fuente: http://wiki.osgeo.org/wiki/Openlayers:_las_herramientas_disponibles_por_defecto#MousePosition

2.4.6. Lenguajes de programación utilizadas

JavaScript

Son fragmentos de código que se los incrusta en una página web/wap cumpliendo la función por la cual fueron creados, dándole así una apariencia mejorada y un dinamismo a la misma.

Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado.

Con JavaScript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

Uso

JavaScript en OpenLayers, permite crear efectos especiales sobre la página web por ejemplo llamar a capas sobre el mapa o cualquier otro dinamismo de la misma. También permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas.

Css

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la

mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

Uso

Las hojas de estilos son utilizadas en la presentación de la plantilla del visor, ya que ahorran tiempo ya que se define el aspecto de visualización de una etiqueta.

HTML

HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento.

Uso

JavaScript en OpenLayers, permite la construcción de la interfaz del visor

.

2.4.7. Clases básicas

2.4.7.1. Tipos básicos de datos

OpenLayers está construida sobre JavaScript y por tanto dispone de los mismos tipos de datos. En JavaScript hay cinco tipos de datos primitivos: Undefined, Null, Boolean, Number y String.

Además se dispone de una colección de objetos nativos: Object, Boolean, Error, SyntaxError, Function, Number, EvalError, TypeError, Array, Date, RangeError, URIError, String, RegExp, y ReferenceError.

2.4.7.2. La clase OpenLayers.Map

OpenLayers.Map es la clase fundamental de la librería OpenLayers. Todo programa de OpenLayers tiene como objeto crear un mapa que se visualizará en pantalla. Los objetos de la clase OpenLayers.Map son una colección de capas, OpenLayers.Layer, que contienen la información que se quiere mostrar, y controles, OpenLayers.Control, que permiten interactuar con el usuario. El objeto Map también es el responsable de gestionar la visualización del mapa en cuanto a Zoom y Panning se refiere.

El constructor de la clase Map tiene la siguiente forma:

```
OpenLayers.Map( div : [DOMElement/ String], options: Object)
```

El primer parámetro es una referencia al elemento 'div' del documento HTML destinado a contener el mapa. En lugar de una referencia se puede pasar una cadena de texto con el 'id' del elemento.

El segundo parámetro es un Array de opciones en la forma 'key:value'. Las opciones son valores de las propiedades del objeto 'Map' que queremos fijar con un valor determinado en el constructor.

Propiedades del objeto Map

Básicamente un objeto Map es una colección de capas (Layer) y controles (Control). Veamos en primer lugar una serie de propiedades que son colecciones de objetos pertenecientes al mapa:

- layers: La propiedad 'layers' es un Array de objetos OpenLayers.Layer, que contiene la colección de capas del mapa. Para gestionar la colección de capas se dispone de los métodos: addLayer(), addLayers(), removeLayer(), getNumLayers(), entre otros.
- controls: Colección de controles asociados al mapa. Para manejar la colección se utilizan los métodos: addControl(), addControls(), entre otros.

2.4.8. Controles

2.4.8.1. Introducción a los controles

Los controles se utilizan para interactuar con el mapa. Permiten hacer zoom, mover el mapa, conocer las coordenadas del cursor, etc.

En OpenLayers V2.10 hay 40 clases de controles para utilizar con los mapas. De ellos, dos está desaconsejado su uso pues se van a suprimir en la próxima versión de OpenLayers, la versión 3.0. Los controles descatalogados son 'MouseDefaults' y 'MouseToolBar'. De los 38 controles restantes, dos son nuevos de esta versión, 'SLDSelect' y 'WMTSGetFeatureInfo'. El resto vienen de versiones anteriores, si bien el control 'OpenLayers.Control.Panel' ha sufrido modificaciones en su funcionamiento respecto de anteriores versiones.

Todos los controles derivan de la clase 'OpenLayers.Control', y todos los controles se añaden al objeto 'OpenLayers.Map', directa o indirectamente. La clase 'OpenLayers.Map' tiene una propiedad 'controls' que guarda la lista de controles del mapa. Para añadir un control al mapa se puede hacer mediante el método 'OpenLayers.Map.addControl()' o bien directamente en el constructor de 'OpenLayers.Map'.

El constructor de la clase OpenLayers.Map permite añadir controles en el momento de la creación del mapa. Si no se indica nada, el mapa añade los siguientes controles por defecto:

- OpenLayers.Control.Navigation
- OpenLayers.Control.PanZoom
- OpenLayers.Control.ArgParser
- OpenLayers.Control.Attribution

La lista de controles que podemos utilizar es la siguiente:

- Controles generales
 - ArgParser
 - Attribution
 - Button
 - Graticule
 - KeyboardDefaults
 - LayerSwitcher

- OverviewMap
- Panel
- PanPanel
- PermaLink
- SLDSelect (nuevo versión 2.10)
- Snapping
- Split
- Zoom, Panning, Position
 - DragPan
 - MousePosition
 - Navigation
 - NavigationHistory
 - NavToolBar
 - Pan
 - PanZoom
 - PanZoomBar
 - ZoomBox
 - ZoomIn
 - ZoomPanel
 - ZoomOut
 - ZoomToMaxExtent
- Features
 - EditingToolbar
 - DragFeature
 - DrawFeature
 - GetFeature
 - ModifyFeature
 - SelectFeature
 - TransformFeature
- Servicios de Mapas
 - WMSGetFeatureInfo.

2.4.8.2. La clase **OpenLayers.Control**

Propiedades:

- id : (String)
- map : (OpenLayers.Map) El mapa al que pertenece el control
- div : (DOMElement) Elemento 'div' que alberga el control
- type: (Number) Especifica el comportamiento del control, dentro del panel.

Puede tomar los valores:

OpenLayers.Control.TYPE_BUTTON.

2.4.8.3. Utilización de los controles

OpenLayers.Control.KeyboardDefaults

Este control permite hacer zoom mediante las teclas '+' y '-' y mover el mapa con las teclas de flecha.

Para activar el control solo hay que crear una instancia de la clase `OpenLayers.Control.KeyboardDefaults` y añadírsela a la colección de controles del mapa.

```
var ctrl = new OpenLayers.Control.KeyboardDefaults();  
map.addControl (ctrl);
```

OpenLayers.Control.Scale

El control `Scale` muestra la escala actual del mapa, en forma de ratio (1: 10000). El constructor tiene la siguiente signatura:

`OpenLayers.Control.Scale (element: DOMElement, options: Object).`

El aspecto y posición del control `Scale` viene definido por su atributo 'class' que es 'olControlScale'. Los valores de estilo por defecto para este control los podemos encontrar en el fichero `style.css` adjunto en el api de `OpenLayers`.

2.4.9. Capas

2.4.9.1. La clase `OpenLayers.Layer`

La clase `OpenLayers.Layer` es la clase base para todos los tipos de capas especializadas, que son las que realmente se añaden a los mapas.

El constructor de la clase `OpenLayers.Layer` tiene la siguiente signatura:

`OpenLayers.Layer` (name: String, options: Object).

El primer parámetro es una cadena que permitirá identificar a la capa, por ejemplo en el control `LayerSwitcher`. El segundo parámetro es un `Hashtag` de opciones adicionales que se añadirán a la capa. La forma de este parámetro es un `Array` asociativo de parejas 'key:value' separadas por comas.

La clase `OpenLayers.Layer` expone las siguientes propiedades:

- **id**: String.- El valor del atributo 'id' asignado al elemento 'div' de la capa
- **name**: String.- El nombre de la capa es una cadena que permite identificar a la capa en algunas situaciones, por ejemplo en el `Control LayerSwitcher`
- **div**: `DOMElement`.- Es una referencia al elemento 'div' que alberga la capa.
- **opacity**: Float.- Es un número entre 0 (= transparente) y 1 (= opaco) que indica el grado de transparencia de la capa.
- **alwaysInRange**: Boolean.- Se debe establecer en 'true' cuando la visualización de la capa no se debe basar en zoom.
- **events**: `OpenLayers.Events`.- La propiedad 'events' es la colección de eventos de la capa. Es una referencia a un objeto de la clase `OpenLayers.Events`.
- **map**: Es una referencia al mapa que contiene a la capa. Se establece en la función `addLayer()` del mapa o en la función `setMap()` de la capa. El objeto apuntado es un `OpenLayers.Map`.
- **isBaseLayer**: Es un Boolean que indica si se trata de una capa base. Por defecto es falso. Las clases derivadas especializadas establecen un valor por defecto para cada tipo de capa.
- **displayInLayerSwitcher**: Boolean que indica si el nombre de la capa debe de aparecer o no en el control `LayerSwitcher`. El valor por defecto es 'true'.
- **visibility**: Es un Boolean que indica si la capa es visible o no. El valor por

defecto es 'true'.

- **attribution**: Se trata de la cadena, String, que se mostrará en el control Attribution.
- **inRange**: Es un Boolean que indica si el valor de la resolución actual está entre el mínimo y el máximo de la capa (minResolution, maxResolution). Se establece cada vez que cambia el zoom.
- **options**: Se trata de un objeto mediante el cual se pueden pasar al constructor de la capa valores iniciales para cualquiera de las propiedades de la capa.
- **gutter**: (Integer) El valor del ancho del borde, si lo tiene. Por defecto es cero
- **projection**: (Object) Objeto OpenLayers.Projection con la proyección de la capa. Si se pasa en el objeto 'options' del constructor, se puede pasar como una cadena del tipo 'EPSG: 4326', pero durante la creación de la capa se construirá un objeto Projection. Cuando se utiliza esta opción suele ser necesario fijar también 'maxExtent', 'maxResolution' y 'units'.
- **units**: (String). Las unidades de medida de la capa. Por defecto son grados, y la variable 'units' tiene el valor 'degrees'. Los valores posibles son: 'degrees' (o 'dd'), 'm', 'ft', 'km', 'mi', 'inches'
- **scales**: (Array). Un array con las escalas del mapa para cada zoom en orden descendente. Este parámetro sólo tiene sentido si está bien calibrado con la resolución concreta del monitor en el que estemos trabajando. Además debe de estar bien definida la propiedad 'units' de la capa. En general es preferible utilizar la propiedad 'resolutions'.
- **resolutions**: (Array). Un array con las resoluciones del mapa para cada zoom en orden descendente. La resolución es el número de unidades de mapa por pixel. Si no se especifica al construir la capa, se calcula en base a otras propiedades de la capa (maxExtent, maxResolution, maxScale, etc).

2.4.9.2. Tipos de capas

OpenLayers.Layer.Markers

OpenLayers nos ofrece una capa especial para alojar nuestros marcadores: OpenLayers.Layer.Markers. Un marcador es una instancia de la clase OpenLayers.Marker, que es una combinación de un icono y una posición. El constructor de la clase OpenLayers.Markers tiene la siguiente forma:

OpenLayers.Markers (name: String, options: Object)

El primer parámetro es una cadena que permitirá identificar a la capa, por ejemplo en el control LayerSwitcher. El segundo parámetro es una Hashtable de opciones extra que queramos asignar a la capa. Las propiedades públicas de la capa Markers son:

- **isBaseLayer**: (Boolean = false) Las capas de marcadores no son capas base. Se sobrescribe la propiedad de la clase base OpenLayers.Layer con el valor false.
- **markers**: (Array(OpenLayers.Marker)) Lista de marcadores de la capa. Esta propiedad es un Array que guarda las referencias a los marcadores que se han ido añadiendo a la capa.
- **drawn** (Boolean): Nos indica si se ha dibujado la capa. En algunas situaciones al inicializar la capa o hacer zoom es necesario comprobar esta variable y si es necesario llamar al método 'draw()'.

OpenLayers.Layer.Vector

Las capas vectoriales están pensadas para alojar 'features' vectoriales, que serán instancias de las clases derivadas de OpenLayers.Feature.Vector. El constructor de la clase OpenLayers.Layer.Vector admite dos parámetros:

OpenLayers.Layer.Vector (name: String, options: Object);

El parámetro 'name' es una cadena de texto que sirva para identificar la capa y el parámetro 'options', que es opcional, es un objeto con propiedades de la capa que queramos establecer en un valor distinto del que se asigna por defecto. Una vez creada la capa se le pueden ir añadiendo las features que queramos visualizar mediante el método 'addFeatures()' que admite como parámetro un Array de features. Dichas 'features' habrán sido creadas previamente y serán instancias de la clase OpenLayers.Feature.Vector. La geometría y el estilo de visualización asignados a la feature serán las que marquen la forma en que se visualizarán las features en la capa. La capa, una vez creada, se añade al mapa mediante el método OpenLayers.Map.addLayer().

La capa WMS

Las capas `OpenLayers.Layer.WMS` están pensadas para albergar los datos provenientes de consultas a los Web Map Services de acuerdo con las especificaciones del OGC (Open Geospatial Consortium). Esta capa hereda de `OpenLayers.Layer.Grid`.

Para acceder a un WMS lo primero que debemos conocer es la url del servicio de mapas.

2.4.10. Eventos

Coordenadas del cursor

En `OpenLayers` disponemos de una gestión de eventos a través del objeto `OpenLayers.Map` que nos permite interactuar con el usuario de diversas maneras. Uno de los eventos proporcionado por el 'Map' es 'mousemove' que nos proporciona las coordenadas en pixels del cursor cuando este se mueve por el mapa. Para utilizar el evento debemos definir una función 'callback' que será llamada cada vez que se dispare el evento, esto es, cada vez que se mueva el cursor sobre el mapa. Además debemos 'registrar' el evento de forma que indiquemos: el evento que queremos gestionar, el objeto 'Map' y la función que hay que llamar cada vez que se dispare el evento.

```
map.events.register("mousemove", map, mouseMoveHandler);
```

2.4.11. Problemas en accesos locales

`OpenLayers` es una librería escrita en Javascript y tiene todas sus ventajas y sus limitaciones.

Por seguridad no se permite que los scripts accedan a los ficheros del ordenador cliente, el que está ejecutando el navegador. Sí que se permite acceder a ficheros en otros servidores.

Cuando se realiza una página HTML que accede a `OpenLayers` y muestra un mapa, siempre que los ficheros solicitados sean externos al ordenador cliente podrá ejecutarla directamente en el navegador y se mostrara los mapas solicitados.

Si ejecuta el HTML directamente en el navegador no podrá acceder a los ficheros de datos.

Para acceder a ficheros de datos debe ejecutar la página HTML a través de un servidor o bien que los ficheros de datos estén en un servidor, como se muestra en la figura II.20.

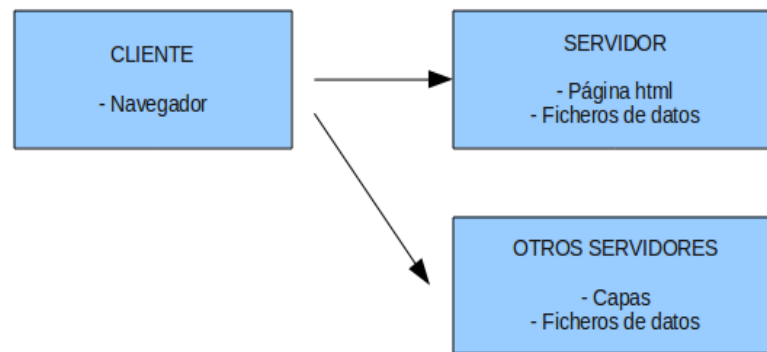


Figura II.20. Esquema de acceso a datos Openlayers

Fuente: <http://bicimap.es/wordpress/?p=749>

En este esquema se puede utilizar el 'localhost' como servidor, pero en el navegador debe ejecutar la página a través de él, esto es, con "http://localhost...".

2.5. VISOR CUENCA

2.5.1. Introducción

Visor Cuenca es un cliente Web GIS construido con clases de JavaScript, mapfish, y ext sin dependencia de servidores de mapas concretos. Este visor ofrece una interfaz de usuario más intuitiva que los anteriores.

Visor Cuenca es un conjunto de librerías escritas en varios lenguajes entre los que sobresalen JavaScript, HTML, CSS y también se hace de un lenguaje dinámico Java para la gestión de las capas los mismos que permiten tener un producto de calidad y que permite visualizar información espacial en la mayoría de los browsers modernos de la web.

Ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, Mapas comerciales (tipo Google Maps, Bing, Yahoo), distintos formatos vectoriales, mapas de OpenStreetMap, etc,

Los lenguajes impredecibles para dicho visor son los lenguajes HTML, CSS, JavaScript y Java para que las páginas web sean dinámicas con el objeto de mostrar la información en una interfaz atractiva y novedosa para el usuario.

Este visor hace uso de las clases básicas, controles, capas y eventos de OpenLayers.

2.5.2. Características

A continuación se pueden mencionar las siguientes características:

- Brinda facilidad de utilizar varios objetos o capas de diferentes servidores y los muestra en una sola aplicación.
- El acceso a muchos formatos de datos ráster: GIF, PNG, JPEG.
- Impresión.
- Permite la navegación de un mapa dentro una página web/wap.
- Visualiza hasta un punto mínimo y máximo del mapa, mediante la manipulación del zoom.
- Permite acceso a formatos de datos vectoriales shapefiles que están dentro de del archivos (.map) y lo trata con si fuesen marcas.
- Facilidad de uso.
- Facilidad al momento de llamar al servidor de mapas ya sea de tipo local o algún servidor externo como google map, yahoo map entre otros.

2.5.3. Tipos de capas que se puede cargar con el Visor Cuenca

En el Visor Cuenca existe el concepto de capa base, que es la capa principal de fondo activa, de la cual se toma el sistema de proyección y los niveles de zoom. Sobre ésta pueden añadirse más capas simultáneas, a modo de overlay, sean ráster o vectoriales. Este visor permite trabajar con un amplio número de orígenes de datos como servidores de mapa open source y además hace uso del estándar OGC por lo que proporciona mucha flexibilidad al usuario, en la tabla II.XII se presenta los tipos de capas o WMS que pueden ser utilizados por el visor.

Tabla II.XII. Tipos comunes de capas usados en Visor Cuenca

Capa	Descripción
Google	Ráster: EPSG 900913 Ofrece varias opciones como satélite, mapa carreteras, híbrido y relieve.
WMS	Ráster: EPSG variable. Permite seleccionar las capas del WMS.
OpenStreetMaps	Ráster: EPSG 900913.

2.5.4. Manejo de herramientas

Marcar y ampliar una zona rectangular del mapa; al dar un clic o elegir una zona en el mapa y aumenta el zoom del mapa, como se muestra en la figura II.21.



Figura II.21. Ampliar una zona
Fuente: Autores

Ampliar y centrar el mapa con un clic; al dar un clic sobre el mapa aumenta el zoom de Mapa, como se muestra en la figura II.22.



Figura II.22. Zoom Out
Fuente: Autores

Reducir y centrar el mapa con un clic; al dar un clic sobre el mapa disminuye el zoom de mapa, como se muestra en la figura II.23.



Figura II.23. Zoom In
Fuente: Autores

Mover el mapa arrastrando el cursor; al dar un clic sobre el mapa se puede desplazar hacia la izquierda, derecha, arriba o abajo, como se muestra en la figura II.24.



Figura II.24. Paneo
Fuente: Autores

Permite agregar una escala grafica al mapa, como se muestra en la figura II.25.



Figura II.25. Escala

Fuente: Autores

En la figura II.26 se muestra el icono por medio del cual al dar clic el mapa puede ser impreso.

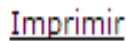


Figura II.26. Imprimir

Fuente: Autores

Permite visualizar en el mapa las coordenadas correspondientes a la ubicación del mouse, como se muestra en la figura II.27.

-80.43091, -1.89754

Figura II.27. Posición del Mouse

Fuente: Autores

Permite regresar al inicio es decir si esta en zoom out o zoom in regresa al inicio tal como se cargó al inicio el mapa, como se muestra en la figura II.28.



Figura II.28. Volver al inicio

Fuente: Autores

La figura II.29 representa el mapa de referencia, que consiste en tener un mapa que contiene la extensión total del mapa base en tamaño reducido, el cual representa por medio de un rectángulo la zona que se visualiza actualmente en el mapa base.



Figura II.29. Mapa de referencia

Fuente: Autores

2.5.5. Lenguajes de programación y clases usadas

JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.

Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador... aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Uso

JavaScript en Visor Cuenca, permite crear efectos especiales y ejecutar instrucciones como respuesta a las acciones del usuario, esto hace que la página sea interactiva.

Css

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación

de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>".

Uso

Las hojas de estilos son utilizadas en la presentación de la plantilla del visor, esto permite ahorrar tiempo al momento de definir el aspecto de visualización de una etiqueta.

HTML

HTML, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también sirve para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

Uso

JavaScript permite la construcción de la interfaz de usuario del visor por la cual se dará la interacción con el usuario.

Mapfish

El proyecto Mapfish es el más reciente en este campo, y ofrece al desarrollador de software un framework con el que crear complejas aplicaciones web GIS.

Mapfish soporta varios lenguajes de programación y combina varios componentes, todos ellos de software libre. Entre estos últimos se incluyen adaptadores de bases de datos, como SQL, Hibernate o la potente librería JavaScript Ext JS, para definir la interfaz. Mapfish puede incluso utilizar OpenLayers como componente de renderización de mapas.

Mapfish está formado por varios componentes, mientras que OpenLayers es parte integral del entorno. OpenLayers, a su vez, incluye sus propios archivos JavaScript y componentes Geo Ext (potentes widgets desarrollados con la librería JavaScript Ext JS). Podemos añadir nuestros propios widgets a Geo Ext en forma de extensiones de usuario. Mapfish está programado en HTML y JavaScript; incluye su propia API, y ha sido diseñado para varios lenguajes de programación.

GeoExt

GeoExt es una biblioteca JavaScript que proporciona una base para la creación de aplicaciones ricas de mapas web. Combina los cartografía en la red de bibliotecas OpenLayers con Extjs ", un cross-browser biblioteca de JavaScript para crear aplicaciones ricas de Internet." GeoExt proporciona un conjunto de widgets personalizables y manejo de datos de apoyo que facilita la creación de aplicaciones para la visualización, edición y estilo de datos geoespaciales.

- Construido con Ext JS, cross-browser marco de aplicaciones ricas de Internet.
- Integrado OpenLayers cliente mapeo.
- Interactividad completa usuario dinámica y animación sin necesidad de Flash o Silverlight.
- Bien diseñado y componente extensible modelo.
- Amplia variedad de widgets genéricos (rejillas, tablas, árboles, diseños, cuadros combinados, barras de herramientas).
- Flexibles y basados en estándares de mapeo reproductores como leyendas, pop-ups, y selectores de escala.
- Controles de impresión en PDF Plataforma de servicio, tales como diseños de

página, resolución y leyendas.

- Personalizable selección de capas.

Ext Js

Ext JS (pronunciado como "ekst") es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM. Fue desarrollada por Sencha.

Originalmente construida como una extensión de la biblioteca YUI por Jack Slocum, en la actualidad puede usarse como extensión para la biblioteca jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente.

Funcionalidades

Dispone de un conjunto de componentes (widgets) para incluir dentro de una aplicación web, como:

- Cuadros y áreas de texto.
- Campos para fechas.
- Campos numéricos.
- Combos.
- Radiobuttons y checkboxes.
- Editor HTML.
- Elementos de datos (con modos de sólo lectura, datos ordenables, columnas que se pueden bloquear y arrastrar, etc.).
- Árbol de datos.
- Pestañas.
- Barra de herramientas.
- Menús al estilo de Windows.
- Paneles divisibles en secciones.
- Sliders.
- Gráficos.

Varios de estos componentes están capacitados para comunicarse con el servidor usando AJAX. También contiene numerosas funcionalidades que permiten añadir interactividad a las páginasHTML, como:

- Cuadros de diálogo.
- *quicktips* para mostrar mensajes de validación e información sobre campos individuales.

Uso

Geoext y Ext JS permiten la construcción del árbol que es el que contiene las carpetas y subcarpetas con las capas que se encuentran descritas dentro del archivo .map.

AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza medianteXMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dados que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Java

Java es un lenguaje de programación de alto nivel orientado a objetos, desarrollado

por James Gosling en 1995. El lenguaje en sí mismo toma mucha de su sintaxis de C, Cobol y Visual Basic, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. La memoria es gestionada mediante un recolector de basura.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

Jsp

JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La Especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la Especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries) externas e incluso personalizadas.

Uso

Java y Jsp permiten la gestión de mapas.

CAPÍTULO III

3. CONFIGURACIÓN Y DESARROLLO DEL PROTOTIPO

Este capítulo contendrá la implementación de un prototipo con las herramientas CartoWeb, OpenLayers, y el Visor Cuenca, ya que en el capítulo siguiente será de vital importancia para el análisis estadístico.

3.1. Instalación y uso de CartoWeb

CartoWeb depende de varios programas para su propio funcionamiento. Algunos son necesarios, otros son opcionales, dependiendo de lo que se quiere hacer.

- Software indispensable:
 - Un servidor Web, como Apache
 - PHP > = 5.0.3
 - MapServer PHP / MapScript (de MapServer > = 4.4)

Instalación

Uno de los requisitos previos para la instalación de CartoWeb, es que primero se debe instalar fgs-MapServer (servidor de mapas) en la siguiente ruta /opt.

Luego descargar CartoWeb desde <http://www.cartoweb.org/downloads.html>, descomprimirlo y situarlo en el siguiente directorio: **/opt/fgs/apps**.

Ahora escriba lo siguiente para proceder con la instalación de CartoWeb, como se muestra en la figura III.30.

```
[root@ide cartoweb3]# php cw3setup.php --install --base-url http://localhost:8080/cartoweb3/htdocs --profile development
installing
Creating directories
Setting permissions
Copying .ini.dist files into .ini (if not existing)
Copying/linking resources into htdocs
Copied <files>.in into <files> for projects: all
Launching makemaps script for projects: all
Launching po2mo script for projects: all
Installation finished...
```

Figura III. 30. Comando instalación CartoWeb
Fuente: Los Autores

Si no ocurrió ningún error, usted debería ser capaz de acceder a la página de CartoWeb escribiendo <http://localhost/cartoweb3/htdocs/>, en su navegador, como se muestra en la figura III.31.

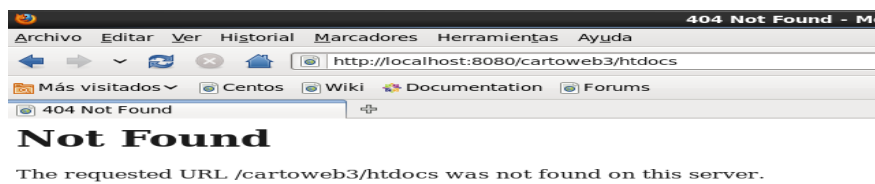


Figura III. 31. Error del servidor
Fuente: Los Autores

Error: EL servidor web no encuentra en su directorio por defecto la página de inicio del CartoWeb, es por esto que antes se debe modificar el archivo `httpd.conf` que se encuentra en el siguiente directorio `/opt/fgs/www/conf`, para que busque la ruta donde se encuentra la aplicación en un directorio externo, como se muestra en la figura III.32.

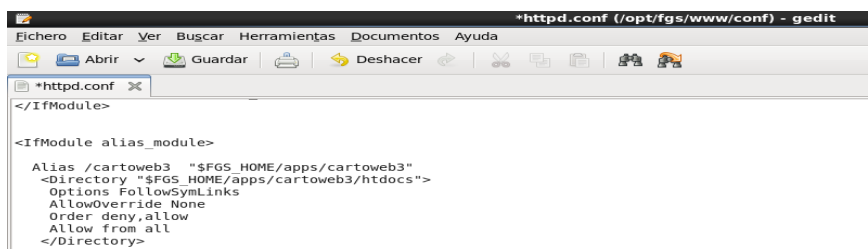


Figura III. 32. Modificación de archivo `httpd.conf`
Fuente: Los Autores

Luego ingresar nuevamente la url (<http://localhost:8080/cartoweb3/htdocs/>), así se comprueba que la instalación fue correcta, como se muestra en la figura III.33.

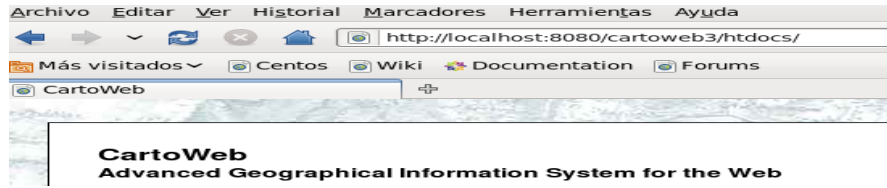


Figura III. 33. Instalación Correcta
Fuente: Los Autores

Creación de nuevo proyecto

Para crear un nuevo proyecto siga los siguientes pasos que se detallan a continuación:

Al momento de finalizar la instalación de CartoWeb, este conlleva en su interior una carpeta llamada `sample_project` que trae consigo archivos básicos para no empezar desde cero la aplicación.

A este paquete se le cambia el nombre por el nombre de su proyecto, para este caso prototipo en el directorio `cartoweb/projects/`. Esta carpeta será el propietario de un árbol similar a la carpeta raíz de CartoWeb,

Por lo tanto se debe incluir en el proyecto nuevo sólo los archivos que difieren de la configuración por defecto. Todos los elementos que se configuran en el proyecto prevalecerán sobre los configurados por defecto en la carpeta raíz, como se muestra en la figura III.34.

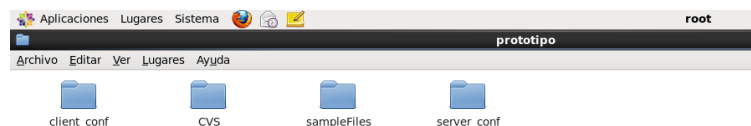


Figura III. 34. Archivos iniciales Prototipo
Fuente: Los Autores

En la carpeta `server_conf` hay dos archivos a los que se les debe cambiar el nombre (`sampleProject.map` y `sampleProject.ini`) con el nombre del nuevo proyecto.

Edite el `client.ini.in` de este nuevo proyecto que se encuentra en `/project/prototipo/client_conf` y modificar el valor del parámetro `mapid` con el nombre del archivo `.map` de su proyecto prototipo, esto se lo hace con el fin de que el cliente sepa el nombre del archivo de asignaciones que se encuentra en el servidor:

```
mapId = prototipo
```

En una línea de comandos (DOS, shell), inicie el script de configuración

cw3setup.php, como se muestra en la figura III.35.

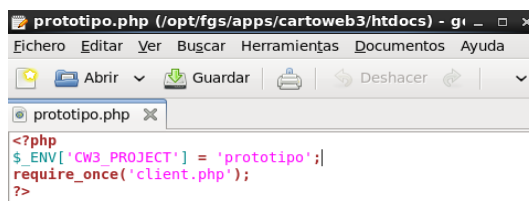
```
[root@ide cartoweb3]# php cw3setup.php --install --base-url http://localhost:8080/cartoweb3/htdocs --profile development --project prototipo
installing
Creating directories
Setting permissions
Copying .ini.dist files into .ini (if not existing)
Copying/linking resources into htdocs
Copied <files>.in into <files> for projects: prototipo
Launching makemaps script for projects: prototipo
Launching po2mo script for projects: prototipo
Installation finished...
```

Figura III. 35. Script de instalación prototipo

Fuente: Los Autores

Para facilitar el desarrollo, se puede crear un archivo de acceso directo para acceder al proyecto directamente. Ir a **/htdocs/** y copie el archivo demoCW3.php.

Cambie el nombre del archivo por el nombre del proyecto, en este caso prototipo. Editar el archivo y modificar la variable `$_ENV['CW3_PROJECT'] = 'prototipo'`. Para que el proyecto se ponga en marcha automáticamente cuando se abra la página `http://localhost:8080/cartoweb3/htdocs/prototipo.php`, como se muestra en la figura III.36.



```
prototipo.php (/opt/fgs/apps/cartoweb3/htdocs) - g...
Fichero Editar Ver Buscar Herramientas Documentos Ayuda
Abrir Guardar Deshacer
prototipo.php
<?php
$_ENV['CW3_PROJECT'] = 'prototipo';
require_once('client.php');
?>
```

Figura III. 36. Modificación archivo prototipo.php

Fuente: Los Autores

Luego ejecutar el siguiente comando para reconstruir el proyecto con los cambios efectuados, como se muestra en la figura III.37.

```
[root@ide cartoweb3]# php cw3setup.php --install --base-url http://localhost:8080/cartoweb3/htdocs --profile development --project prototipo
```

Figura III. 37. Ejecución de comando para la reconstrucción del proyecto prototipo

Fuente: Los Autores

El proyecto se encuentra ahora disponible directamente con la dirección URL `http://localhost:8080/cartoweb3/htdocs/prototipo.php`, como se muestra en la figura III.38.

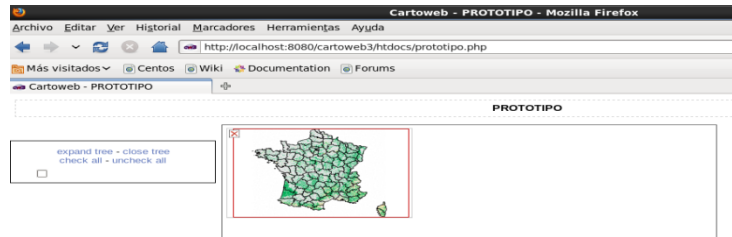


Figura III. 38. Proyecto prototipo inicial
Fuente: Los Autores

Como se ve en la figura III.10, hay 2 directorios por defecto una del lado del cliente (client_conf) y otra del lado del servidor (server_conf), los mismos se encargaran de solicitar información y del lado del servidor de responder a dichas solicitudes, además se debe tener en cuenta la parte de la presentación, esto se lo hace por medio de la creación de la carpeta templates que contendrá la plantilla cartoclient.tpl, la cual llamara el resto de plantillas que visualizara el cliente, como se muestra en la figura III.39.

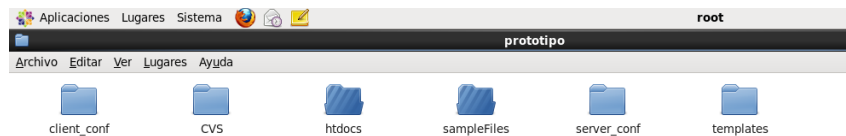


Figura III. 39. Directorio client_conf y server_conf
Fuente: Los Autores

También se crea el directorio htdocs que contendrá 2 subcarpetas más una con la hoja de estilo y el gfx que tendrá los iconos respectivamente, como se muestra en la figura III.40.

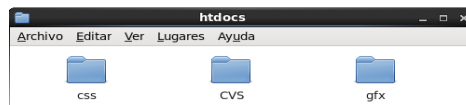


Figura III. 40. Directorio htdocs
Fuente: Los Autores

Como puede ver después de ingresar a su proyecto, este se encuentra vacío. Entonces, vamos a cargar los datos espaciales en el mismo.

Ubicar el mapa de referencia, el cual estará dentro de la siguiente ubicación **/server_conf/prototipo/images**, como se muestra en la figura III.41.



Figura III. 41. Mapa de referencia
Fuente: Los Autores

En el archivo prototipo.map llamar al mapa del Ecuador que va a servir como referencia, como se muestra en la figura III.42.

```
FORMATOPTION "INTERLACE=OFF"  
FORMATOPTION "QUALITY=88"  
END  
  
REFERENCE  
IMAGE images/ref_ecuador.png  
EXTENT -732143.488300 9445216.350100 1147851.641703 10189398.353102  
STATUS ON  
COLOR -1 -1 -1  
OUTLINECOLOR 200 0 0  
SIZE 120 90  
END  
  
SCALEBAR  
POSTLABELCACHE TRUE  
STYLE 0  
UNITS Kilometers  
SIZE 400 3  
TRANSPARENT TRUE  
COLOR 77 77 88
```

Figura III. 42. Prototipo.map - mapa de referencia
Fuente: Los Autores

Ubicar los datos disponibles, de forma predeterminada en el directorio data, que es el directorio que contiene la información espacial, por lo general archivos .shp, como se muestra en la figura III.43.

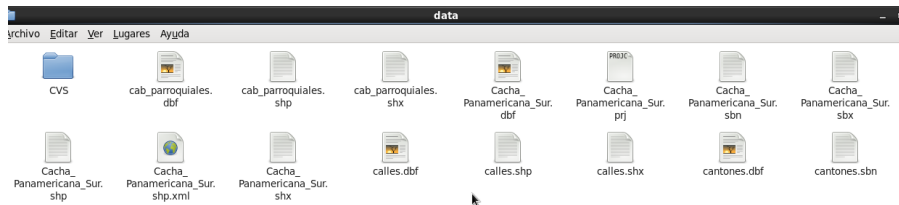


Figura III. 43. Información almacenada en directorio data
Fuente: Los Autores

Modificar el parámetro extend que está dentro del archivo prototipo.map, con el fin de que la información sea georeferenciada correctamente en el navegador, como se muestra en la figura III.44.

```
prototipo.map X  
MAP  
NAME "prototipo"  
EXTENT -732143.488300 9445216.350100 1147851.641703 10189398.353102  
IMAGETYPE PNG
```

Figura III. 44. Referencia de la capa base Ecuador
Fuente: Los Autores

Como se mencionó con anterioridad este visor tiene como característica principal la modularidad, lo que permite establecer estructuralmente la aplicación. Es por esto que se debe crear dos carpetas 1.coreplugins y 2.plugins, como se muestra en la figura III.45.

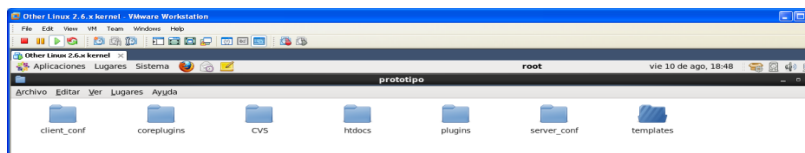


Figura III. 45. Directorio Coreplugins y plugins
Fuente: Los Autores

Ahora se define las capas que van a ser visualizadas en el navegador, esto se lo hace mediante parámetros escritos en el archivo de asignaciones (prototipo.map) , como se muestra en la figura III.46.

```
##capa mapa_Ecuador##  
  
LAYER  
  NAME "ecuador"  
  TYPE POLYGON  
  STATUS ON  
  DATA "mxcantones"  
  TRANSPARENCY 20 #Transparencia para que capas inferiores puedan visualizarse  
  TEMPLATE "ttt"  
  METADATA  
    "exported values" "id attribute string"  
    "id_attribute_string" "DPA_CANTON|string"  
    "query_returned_attributes" "DPA_DESPRO"  
  END  
  CLASS  
    NAME Ecuador  
    STYLE  
      SIZE 80  
      COLOR 255 0 0  
    END  
    LABEL  
      TYPE TRUETYPE  
      FONT "Vera"  
      SIZE 7  
      COLOR 0 0 0  
      OUTLINECOLOR 255 255 255  
      POSITION lc #a posicion inferior centrada  
      BUFFER 2 #cobertura de cada etiqueta en pixeles, para q no haya superposiciones entre las etiquetas  
    END  
  END  
END
```

Figura III. 46. Capas en el archivo prototipo.map
Fuente: Los Autores

Luego modifique el archivo layers.ini (server_conf/prototipo/) y agregue lo siguiente, como se muestra en la figura III.47.

```
layers.ini (sept/psuaps/cartasweb3/projects/prototipo/server_conf/prototipo) - gedit  
autoClassLegend = true #genera la leyenda automaticamente  
  
: Creacion de grupo de capas en la leyenda  
Layers.root.className = LayerGroup  
Layers.root.children = informacion, vias, laguna  
  
: capa de grupo  
Layers.informacion.className = LayerGroup  
!llama a todas las capas  
Layers.informacion.children = ecuador, Chimborazo, Riobamba, Penipe, Guamate, Cumanda, Colta, ciudades_ubicacion  
Layers.informacion.Label = Cantones  
  
Layers.vias.className = LayerGroup  
Layers.vias.Label = Vias  
Layers.vias.children = Linea_ferrea, Cubjijies_Quimiag, Cuatro_Esquinas_Sta_Rosa_de_Chupipigpio,  
Cacha_Panamericana_Sur  
  
: capa individual  
Layers.laguna.className = Layer  
Layers.laguna.Label = Lagunas  
Layers.laguna.asLayer = lagunas
```

Figura III. 47. Creación de leyenda en el archivo layers.ini
Fuente: Los Autores

El resultado será este, como se muestra en la figura III.48.

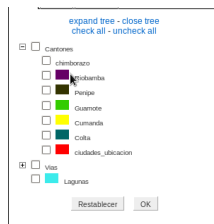


Figura III. 48. Resultado leyenda
Fuente: Los Autores

Modifique el archivo prototipo.ini, para que aparezca el mapa del Ecuador automáticamente al ingresar a la aplicación, como se muestra en la figura III.49.

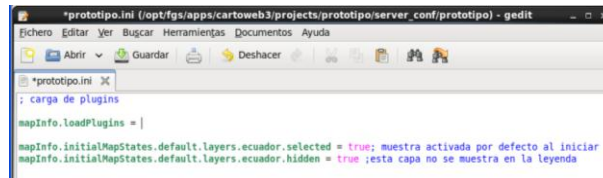


Figura III. 49. Prototipo.ini
Fuente: Los Autores

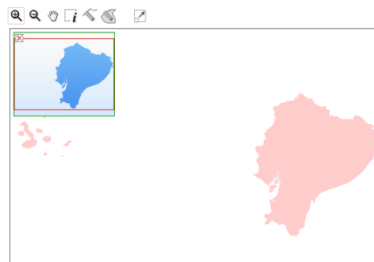


Figura III. 50. Resultado mapa del Ecuador
Fuente: Los Autores

Dentro de la carpeta coreplugins creamos lo siguiente, la carpeta layers que contendrá 2 subdirectorios, tales como htdocs que tendrá la hoja de estilo que va a utilizar la leyenda y templates que contendrá el archivo layers.tpl, como se muestra en la figura III.51.

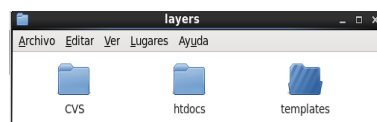
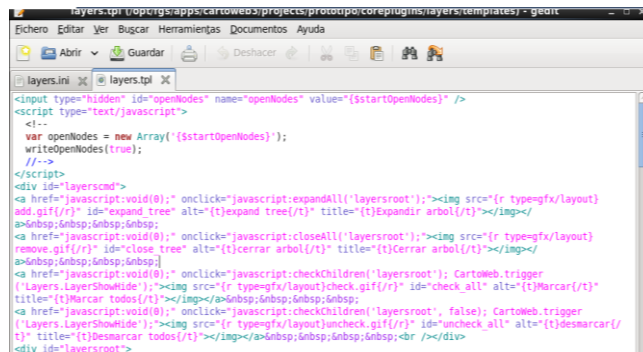


Figura III. 51. Directorio htdocs y templates
Fuente: Los Autores

Aquí cambiamos las letras por iconos que tendrá la leyenda, como se muestra en la figura III.52.



```
<input type="hidden" id="openNodes" name="openNodes" value="{startOpenNodes}" />
<script type="text/javascript">
<!--
var openNodes = new Array( '{startOpenNodes}' );
writeOpenNodes(true);
//-->
</script>
<div id="layerscmd">
<a href="javascript:void(0);" onclick="javascript:expandAll('layersroot');"></img></a>
<a href="javascript:void(0);" onclick="javascript:closeAll('layersroot');"></img></a>
<a href="javascript:void(0);" onclick="javascript:checkChildren('layersroot'); CartoWeb.trigger('{Layers.LayerShowHide}');"></img></a>
<a href="javascript:void(0);" onclick="javascript:checkChildren('layersroot', false); CartoWeb.trigger('{Layers.LayerShowHide}');"></img></a>
</div>
<div id="Layersroot">
```

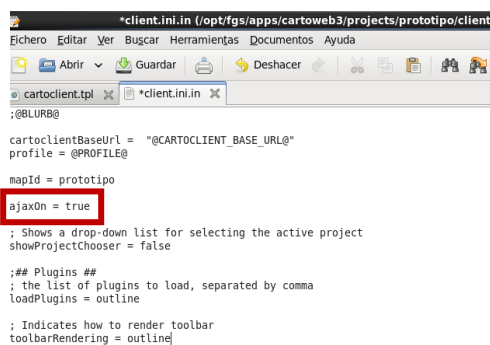
Figura III. 52. Archivo layers.tpl
Fuente: Los Autores

Resultado de leyenda, como se muestra en la figura III.53.



Figura III. 53. Leyenda
Fuente: Los Autores

En client.ini.in activar ajax mediante true, como se muestra en la figura III.54.



```
@BLURB@
cartoClientBaseUrl = "@CARTOCLIENT_BASE_URL@"
profile = @PROFILE@
mapId = prototipo
ajaxOn = true
; Shows a drop-down list for selecting the active project
showProjectChooser = false
;## Plugins ##
; the list of plugins to load, separated by comma
loadPlugins = outline
; Indicates how to render toolbar
toolbarRendering = outline
```

Figura III. 54. Activación de Ajax
Fuente: Los Autores

Llamada al archivo cartoclient_ajaxHeader.tpl en el archivo cartoclient.tpl

```
{include file="cartoclient_ajaxHeader.tpl"}
```


No se debe olvidar cargar el mismo. La activación de complementos normal se lleva a cabo mediante el establecimiento del parámetro *loadPlugins* en *client_conf/client.ini* para *plugins CartoClient* y en *server_conf/<mapId>/<mapId>.ini* para *CartoServer*. Por ejemplo:

```
loadPlugins = outline, exportPdf
```

Luego de la personalización se obtendrá esto, como se muestra en la figura III.58:

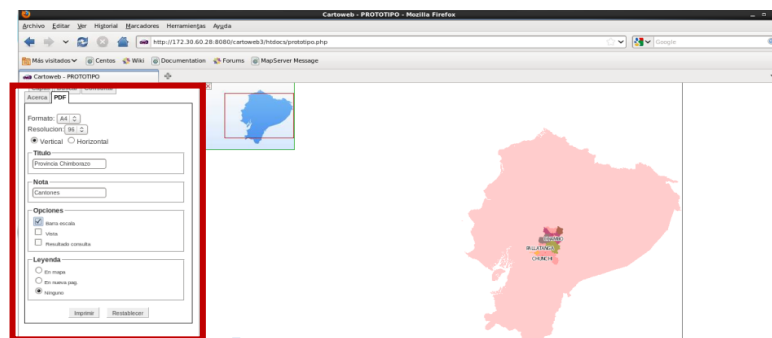


Figura III. 58. Exportación Pdf
Fuente: Los Autores

Después de lanzar el script de instalación del prototipo y la actualización de la aplicación (restablecer), usted debería ser capaz de probar la nueva funcionalidad, como se muestra en la figura III.59.

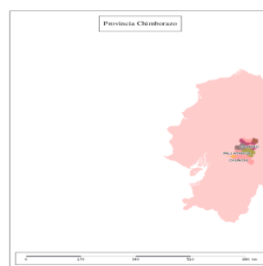


Figura III. 59. Archivo Pdf con el mapa
Fuente: Los Autores

Resultado Final

Prototipo finalizado, como se muestra en la figura III.60.

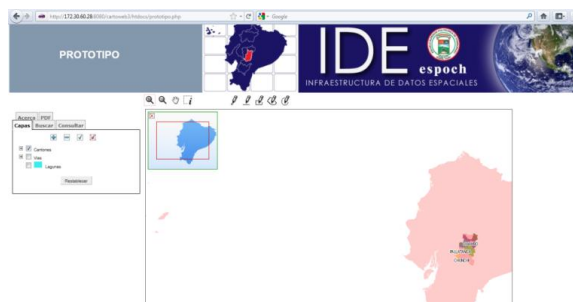


Figura III. 60. Prototipo CartoWeb
Fuente: Los Autores

Todos los módulos serán llamados a través de una plantilla web principal que los contiene en este caso `cartoclient.tpl`, que a su vez está contenida en el directorio `templates`, situado en `/prototipo/`, esta plantilla puede ser usada con otros archivos de configuración como CSS y JavaScript que van a permitir personalizar la misma.

Recordar luego de realizar algún cambio es necesario limpiar – clean.

```
[root@ide cartoweb3]# php cw3setup.php --clean
```

Luego poner en marcha el script de configuración para que el nuevo recurso que acaba de crear se copie o vincule a la aplicación,

```
[root@ide cartoweb3]# php cw3setup.php --install --base-url http://localhost:8080/cartoweb3/htdocs --profile development --project prototipo
```

3.2. Instalación y uso de OpenLayers

Requisitos

Los requerimientos para el funcionamiento de OpenLayers son los siguientes

- Tomcat.
- MapServer 4.x o superior (Windows) - `fgsmapsverbasic5.6.3-fgs_9.5` (Linux).
- JavaScript.

Instalación

Su instalación es sencilla y a la vez obedece a ciertos pasos que se recomienda seguirlos para un mejor desempeño:

Descargar el paquete OpenLayers 2.12 desde www.OpenLayers.org, www.maptools.org o <http://openlayers.org/download/>

Descomprimir el paquete y ubicarlo en el siguiente path del proyecto a desarrollarse **/opt/fgs/www/htdocs/prototipo/**, como se muestra en la figura III.61.



Figura III. 61. API OpenLayers-2-10
Fuente: Los Autores

Uno de los errores más comunes cuando trabajamos con OpenLayers es el path de las librerías, ya que muchos ejemplos de la web los toman de diversos servidores, la respuesta a este es direccionarlo correctamente donde se encuentran las librerías con las que vamos a trabajar.

Se agrega una referencia a la API de OpenLayers. Esta referencia no tiene que ser necesariamente la última versión ofrecida por la página sino que se puede descargar y usar en el servidor propio. Después de todo, la API es simplemente un archivo .js (versión unida) o un conjunto de ellos (versión "multiarchivo"), como se muestra en la figura III.62.

```
<link rel="stylesheet" href="/prototipo/OpenLayers-2.10/theme/default/style.css" type="text/css">
```

Figura III. 62. Referencia a la API de OpenLayers
Fuente: Los Autores

Ingrese la url (<http://localhost:8080/prototipo/OpenLayers-2.10/>), para comprobar si tiene acceso a la librería, como se muestra en la figura III.63.

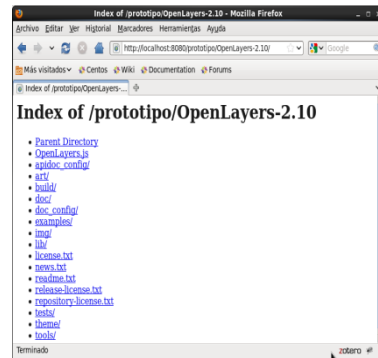


Figura III. 63. Url para comprobar acceso a OpenLayers
Fuente: Los Autores

Creación de nuevo proyecto

Para crear un nuevo proyecto siga los siguientes pasos que se detallan a continuación:
Cree un archivo prototipo.map en el cual se definirá las capas que van a ser visualizadas en el navegador, como se muestra en la figura III.64.



Figura III. 64. Archivo prototipo.map
Fuente: Los Autores

Es importante modificar el parámetro extend que está dentro del archivo mapfile, con el fin de que la información sea georeferenciada correctamente en el navegador, como se muestra en la figura III.65.

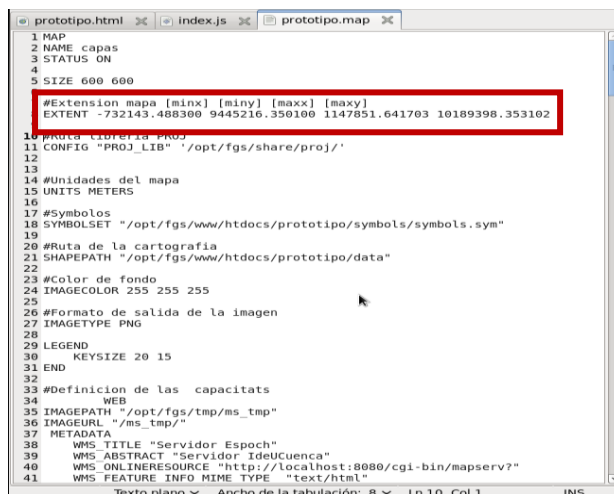


Figura III. 65. Extensión del mapa en prototipo.map
Fuente: Los Autores

```
prototipo.map X
83 LAYER
84 NAME "DICE"
85 SIZEUNITS PIXELS
86 TYPE POLYGON
87 UNITS METERS
88 STATUS ON
89 DATA DICE
90 DUMP TRUE
91 METADATA
92   "wms_srs" "epsg:4326 EPSG:900913"
93   "wms_name" "DICE"
94 END
95 TOLERANCE 9
96 TOLERANCEUNITS PIXELS
97 TRANSPARENCY 100
98 SYMBOLSCALE 25000
99 CLASS
100 NAME "Chimborazo"
101 STYLE
102 SYMBOL 'solid'
103 SIZE 10
104 MINSIZE 10
105 MAXSIZE 10
106 COLOR 0 255 0
107 OUTLINECOLOR 32 32 32
108 END
109 LABEL
110 ANTIALIAS TRUE
111 POSITION AUTO
112 TYPE TRUETYPE
113 FONT arial-bold
114 MINSIZE 6
115 MAXSIZE 10
116 SIZE 6
117 BUFFER 2
118 COLOR 0 255 0
119 OUTLINECOLOR 255 0 0
120 END
121 END
122 END
123
124
125
126 LAYER
127 NAME "Alausi"
128 DATA Alausi
129 STATUS on
```

Figura III. 66. Capas del mapa en prototipo.map
Fuente: Los Autores

Ubicar los datos disponibles, de forma predeterminada en el directorio data, que es el directorio que contiene la información espacial, por lo general archivos .shp, como se muestra en la figura III.67.

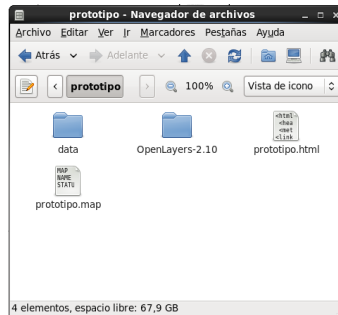


Figura III. 67. Directorio de información geoespacial
Fuente: Los Autores

Cree un archivo prototipo.html el cual es simplemente una página con un div en donde se insertara el mapa, como se muestra en la figura III.68.

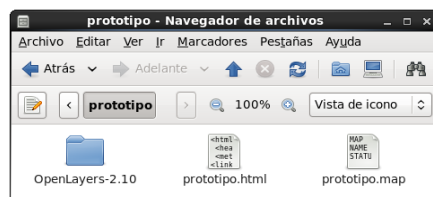
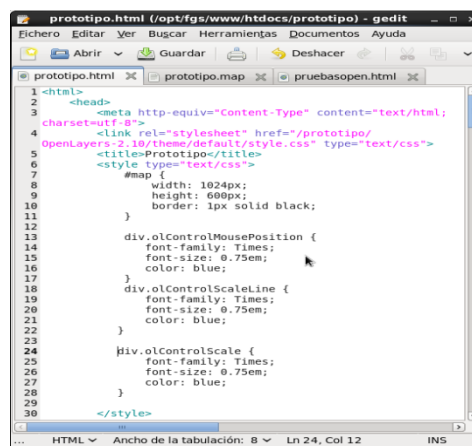


Figura III. 68. Directorio de información geoespacial
Fuente: Los Autores

Se aplican estilos al código. Las clases que comienzan con "ol" son estilos aplicados a objetos de OpenLayers y sirven para ilustrar que es posible personalizar su apariencia. Sin embargo, la parte más importante de los estilos es definir el tamaño del mapa y un borde.

El tamaño definido es necesario por dos razones: la primera es que el mapa en si no maneja dimensiones de su representación por lo que sin un tamaño no se verá y la segunda es para poder trabajar con resolución automática si se desea.

El borde no es necesario pero lo sugiero debido a que es más fácil para los usuarios ver un borde que defina lo que es un mapa y lo que no lo es, sobre todo cuando el fondo de la página y del mapa es el mismo color, como se muestra en la figura III.69.



```
1 <html>
2 <head>
3   <meta http-equiv="Content-Type" content="text/html;
4     charset=utf-8">
5   <link rel="stylesheet" href="/prototipo/
6     OpenLayers:2.10/theme/default/style.css" type="text/css">
7   <title>Prototipo</title>
8   <style type="text/css">
9     #map {
10      width: 1024px;
11      height: 600px;
12      border: 1px solid black;
13    }
14    div.olControlMousePosition {
15      font-family: Times;
16      font-size: 0.75em;
17      color: blue;
18    }
19    div.olControlScaleLine {
20      font-family: Times;
21      font-size: 0.75em;
22      color: blue;
23    }
24    div.olControlScale {
25      font-family: Times;
26      font-size: 0.75em;
27      color: blue;
28    }
29  </style>
30
```

Figura III. 69. Estilos aplicados a los objetos de OpenLayers
Fuente: Los Autores

Código JavaScript

onLoad

Es importante primero mencionar que el código se utiliza en el evento onload debido a que el JavaScript hace referencia a objetos del HTML (div del mapa y otros posibles elementos), por lo que si se utilizara antes de terminar de cargar correctamente la página, sucederían errores y el mapa no se cargaría, como se muestra en la figura III.70.

```
function init()
{
  map = new OpenLayers.Map( 'map', { maxExtent: new OpenLayers.Bounds(-732143.488308, 9445216.350188, 1147851.641703, 10189398.353102),
    maxResolution: 'auto',
    units: 'm',
    Projection: "EPSG:4326"});
}
```

Figura III. 70. Función init()
Fuente: Los Autores

Luego, si bien `onload="init()"` dentro del `body` es la manera más usada, como se muestra en la figura III.71.

```
<body onload="init()">
```

Figura III. 71. Llamar a la función `init()`
Fuente: Los Autores

Init

`Init` es la función que define la inicialización de todos los objetos del mapa (capas, controles, propiedades, colores, estilos, handlers, etc.).

Crear el objeto Map

Este objeto es el objeto principal y es el que contiene las capas y los controles a mostrarse. Las opciones son una serie de parámetros iniciales que pueden aplicarse. Si se desea trabajar con otra proyección que no es la predeterminada es importante cambiar una serie de parámetros para que OpenLayers lo muestre correctamente, como se muestra en la figura III.72.

```
map = new OpenLayers.Map( 'map', { maxExtent: new OpenLayers.Bounds(-732143.488300, 9445216.350100, 1147051.641703, 10189398.353102),  
maxResolution: 'auto',  
units: "m",  
Projection: "EPSG:4326"});
```

Figura III. 72. Objeto Map
Fuente: Los Autores

Controles

Los controles son objetos que controlan cierta interacción con el usuario, ya sea mostrándole la posición en la que se encuentra el mouse `MousePosition`, entre muchos otros, como se muestra en la figura III.73.

```
map.addControl(new OpenLayers.Control.ScaleLine());  
map.addControl(new OpenLayers.Control.MousePosition());  
map.addControl(new OpenLayers.Control.MouseToolbar());  
map.addControl(new OpenLayers.Control.Navigation());  
map.addControl(new OpenLayers.Control.LayerSwitcher());  
map.addControl(new OpenLayers.Control.EditingToolbar(vlayer));  
map.addControl(new OpenLayers.Control.ArgParser());  
map.addControl(new OpenLayers.Control.OverviewMap());
```

Figura III. 73. Controles
Fuente: Los Autores

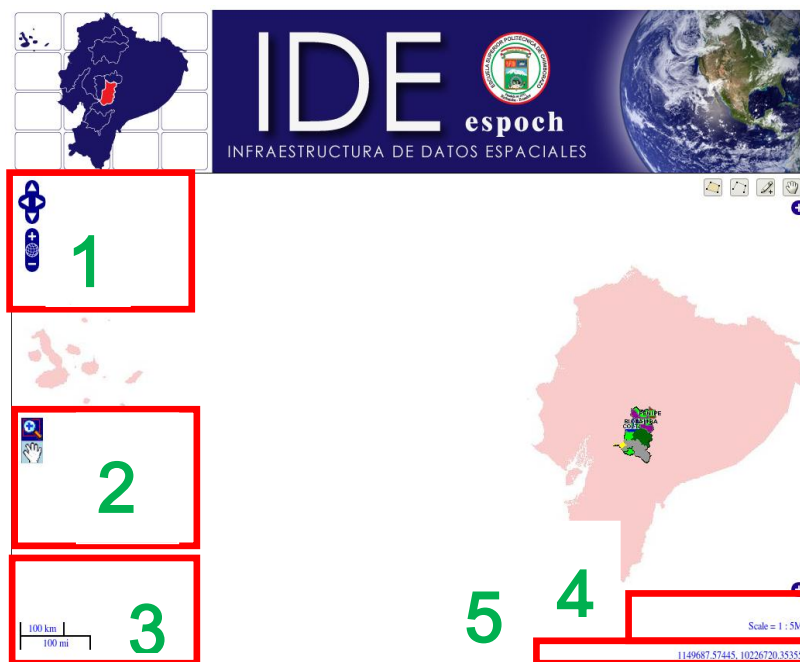


Figura III. 74. Controles
Fuente: Los Autores

1. Pan Zoom Bar
2. Mouse Toolbar
3. Scale Line
4. Scale
5. Mouse Position

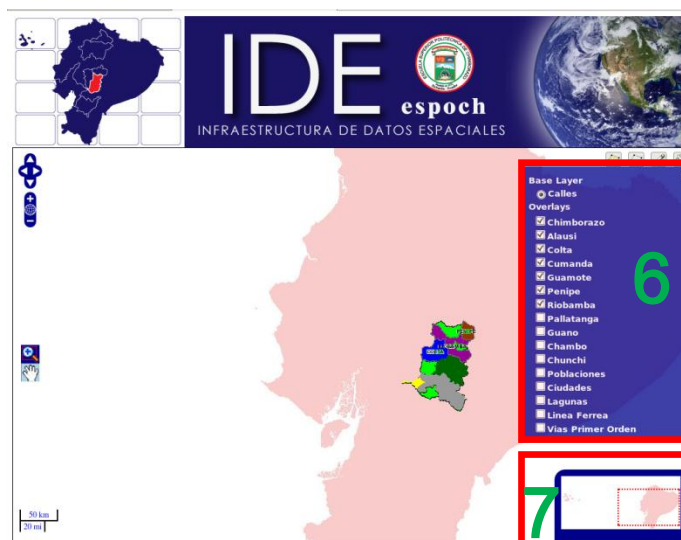


Figura III. 75. Controles
Fuente: Los Autores

6. Layer Switcher
7. Overview Map

Capas

Las capas son en sí la información geográfica que se va a presentar al usuario y por lo tanto la parte más importante del código, entonces se debe escribir en el archivo prototipo.html parámetros como el nombre de la capa, la ruta donde se encuentra alojado el archivo .map que es el que contiene los parámetros que definen la capa, como se muestra en la figura III.76.

```
prototipo.html X prototipo.map X pruebasopen.html X
42 layer = new OpenLayers.Layer.WMS( "Calles", "http://localhost:8080/cgi-bin/mapserv?map=/opt/fgs/www/htdocs/prototipo/prototipo.map",
(layers: "nacantonez" );
43
44 layer1 = new OpenLayers.Layer.WMS( "Chimborazo",
45 "http://localhost:8080/cgi-bin/mapserv?map=/opt/fgs/www/htdocs/prototipo/prototipo.map", {layers: "DICE", visibility: true, transparent:
true, format: "image/png"}, {isBaseLayer: false, visibility: false});
46
47 layer2 = new OpenLayers.Layer.WMS( "Alausi",
48 "http://localhost:8080/cgi-bin/mapserv?map=/opt/fgs/www/htdocs/prototipo/prototipo.map", {layers: "Alausi", visibility: true, transparent:
true, format: "image/png"}, {isBaseLayer: false, visibility: false});
49
50 layer3 = new OpenLayers.Layer.WMS( "Colta",
51 "http://localhost:8080/cgi-bin/mapserv?map=/opt/fgs/www/htdocs/prototipo/prototipo.map", {layers: "Colta", visibility: true, transparent:
true, format: "image/png"}, {isBaseLayer: false, visibility: false});
52
53 layer4 = new OpenLayers.Layer.WMS( "Cumanda",
54 "http://localhost:8080/cgi-bin/mapserv?map=/opt/fgs/www/htdocs/prototipo/prototipo.map", {layers: "Cumanda", visibility: true, transparent:
true, format: "image/png"}, {isBaseLayer: false, visibility: false});
55
56
57
58
```

Figura III. 76. Capas
Fuente: Los Autores

Posicionando el mapa

Inicialmente asigne al mapa en qué posición y zoom debe empezar, como se muestra en la figura III.77.

```
prototipo.html X index.js X
104
165 var lat=-1.850
166 var lon=-79.0246;
167 var zoom=19;
168 map.setCenter(new OpenLayers.LonLat(lon, lat), zoom);
169
```

Figura III. 77. Capas
Fuente: Los Autores

Resultado final

Ingrese en el navegador la siguiente dirección <http://ide.esPOCH.edu.ec:8080/prototipo/prototipo.html>, como se muestra en la figura III.78.

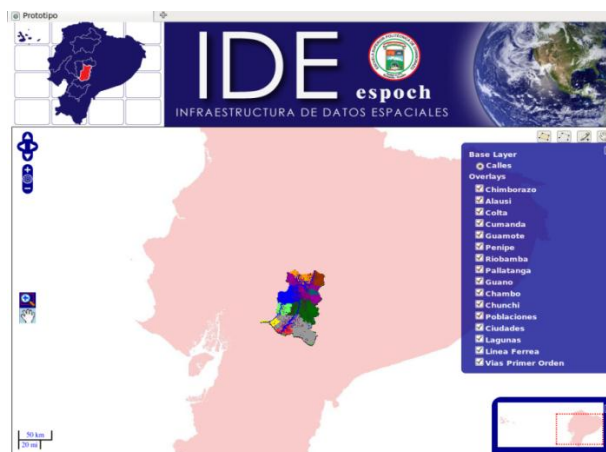


Figura III. 78. Prototipo OpenLayers

Fuente: Los Autores

3.3. Instalación y uso del Visor Cuenca

Requisitos

Visor Cuenca depende de varios programas para su propio funcionamiento. Algunos son necesarios, otros son opcionales, dependiendo de lo que se quiere hacer.

- JDK.
- Tomcat.
- Mapserver.

Instalación

Su instalación es sencilla y a la vez obedece a ciertos pasos que se recomienda seguirlos para un mejor desempeño:

La tabla III.XIII indica los puertos en los que se instalará las aplicaciones

Tabla III.XIII. Puertos de las aplicaciones

				Path
Apache	80	http://ide.esepoch.edu.ec	Páginas Web Ide.ucuenca.edu.ec , etc	/var/www/
Tomcat	8085	http://ide.esepoch.edu.ec	Aplicaciones IDE	/usr/local/apache-tomcat-6.0.35"/webapps/
Fgs	8080	http://ide.esepoch.edu.ec	Mapserver-Linux	/opt/fgs/

Elaborado por: Los Autores

Copiar “ideucuenca” en **/usr/local/apache-tomcat-6.0.35/webapps/**, como se muestra en la figura III.79.

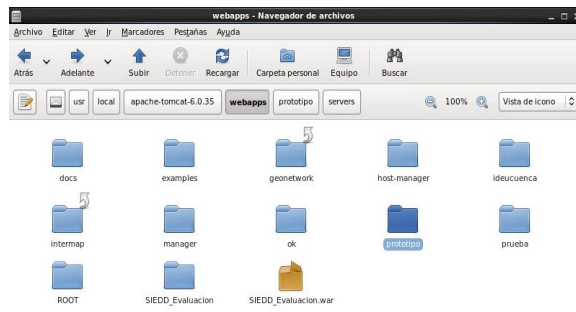


Figura III. 79. Instalación Visor
Fuente: Los Autores

Registrar la clave google en: <http://code.google.com/apis/maps/>

Agregar la Key al archivo “/prototipo/gclient/index.jsp”



Figura III. 80. Archivo para agregar clave
Fuente: Los Autores

Y ponga la siguiente clave en dicho archivo, como se muestra en la figura III.81.

```
<!-- KEY Google localhost -->  
<script src="http://maps.google.com/maps?  
file=api&v=2&sensor=false&key=A1zaSyDJsQPZCD57sph7rAxu5wifA7_jHwAZ5VY*" type="text/  
javascript"></script>
```

Figura III. 81. Clave de google
Fuente: Los Autores

Estructura de directorios



Figura III. 82. Directorios aplicación web

Fuente: Los Autores

Directorio raíz **“gclient”**: Se encuentran todos los JSP (Java Server Pages) necesarios para el funcionamiento de la aplicación.

Directorio **“gclient/css”**: Se encuentran todos los archivos CSS (Cascading Style Sheets) para dar estilo a la aplicación.

Directorio **“gclient/efects”** y **“gclient/image”**. Contienen imágenes de ventanas (maximizar, minimizar, etc.) utilizadas por la aplicación.

Directorio **“share/css”**. Directorio no utilizado actualmente, su uso está previsto para albergar estilos CSS en el caso que se quiera crear visores temáticos.

Directorio **“share/img”**. Directorio no utilizado actualmente, su uso está previsto para albergar imágenes en el caso que se quiera crear visores temáticos.

Directorio **“share/javascript”**. Contiene las librerías Javascript utilizadas por la aplicación

- **“share/javascript/images”**. Imágenes de las herramientas de zoom.
- **“share/javascript/js”**. Librerías propias de la aplicación basadas en OpenLayers.
- **“share/javascript/OL”**. Librerías de OpenLayers
- **“share/javascript/pro4js”**. Librerías de coordenadas.
- **“share/javascript/prototype”**. Librería requerida por OpenLayers
- **“share/javascript/theme”**. Temas y estilos CSS de OpenLayers
- **“share/javascript/arbol”**. Contiene la estructura del árbol para el panel de servidores del visor de mapas.

Directorio raíz **“Servers”**: Contiene archivos JS con las capas que han sido publicadas por los usuarios. El nombre de estos archivos empieza por

“servidores_editor_aideucuenca.js” y son leídos de forma automática por las aplicaciones **Gclient**.

También contiene archivos referentes a los servidores de mapas usados como capa de referencia.

Archivos principales para el funcionamiento del viso

Archivo “**gclient/index.jsp**”. Se encarga de arrancar la aplicación y carga los archivos:

- “**http://maps.google.com/maps?**”. API de Google Maps. Permite visualizar capas de Google Maps
- “**/share/javascript/OL/lib/OpenLayers.js**”. Archivo inicial de OpenLayers
- “**http://www.openstreetmap.org/openlayers/OpenStreetMap.js**”. Permite o visualizar capa de OpenStreetMaps.
- “**/share/javascript/prototype/prototype-1.6.0.3.js**”. Librería requerida por OpenLayers

Librerías para definir coordenadas del mapa:

- “**/share/javascript/proj4js/proj4js.js**”, “**/share/javascript/proj4js/projCode/merc.js**”, “**/share/javascript/proj4js/projCode/merc.js**”, “**/share/javascript/proj4js/projCode/utm.js**”, “**/share/javascript/proj4js/defs/EPSG900913.js**” y “**/share/javascript/coord.js**”
- “**/servers/servers_google.js**” .Carga los servidores definidos por el administrador.
- “**/share/javascript/main.js**”. Contiene funciones para activar y desactivar las capas de referencia.
- “**/share/javascript/dhtmlwindow/dhtmlwindow.js**”. Librería de efectos de ventanas emergentes.
- “**/share/javascript/index.js**”. Contiene funciones para inicializar el mapa y cargar todas las capas.
- “**/share/javascript/busquedas.js**”. Contiene funciones para buscar un lugar y direcciones en el mapa.
- Archivo “**gclient/configuration_client.jsp**”. Carga variables y rutas definidas por el administrador, así como los archivos de lenguaje.
- Archivo “**gclient/share/javascript/index.js**”. Archivo principal dónde se crea el objeto Map de OpenLayers y dónde se cargan las capas de Google Maps y OpenStreetMaps como capas de referencia. También se cargan las capas procedentes del Editor y las definidas por el administrador.

Uso del visor

Crear el ideespoch.map, como se muestra en la figura III.84.



Figura III. 83. Path ideespoch.map
Fuente: Los Autores

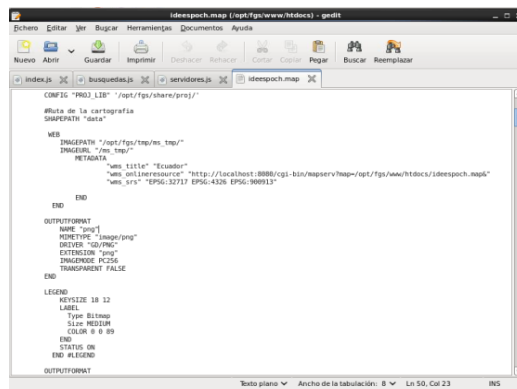


Figura III. 84. Creacion ideespoch.map
Fuente: Los Autores

En el directorio data se almacenan las capas tipo shape que serán cargados en el visor, como se muestra en la figura III.85.



Figura III. 85. Directorio archivo shapes
Fuente: Los Autores

Cargar en el archivo **"/prototipo/servers/servers_google.js"** los servidores, capas y crear una variable con la dirección del servidor, como se muestra en la figura III.87.



Figura III. 86. Path de servers_google.js
Fuente: Los Autores

La figura III.91 muestra la ubicación del archivo index.js, en el cual se agrega una referencia a la API de OpenLayers. Después de todo, la API es simplemente un archivo .js y a todos los archivos que se utiliza para la visualización de los datos espaciales.



Figura III. 91. Path de index.js
Fuente: Los Autores

En la figura III.93 se muestra el archivo index.jsp que es donde se van a describir las librerías necesarias para que las capas sean presentadas al usuario.



Figura III. 92. Path de index.jsp
Fuente: Los Autores

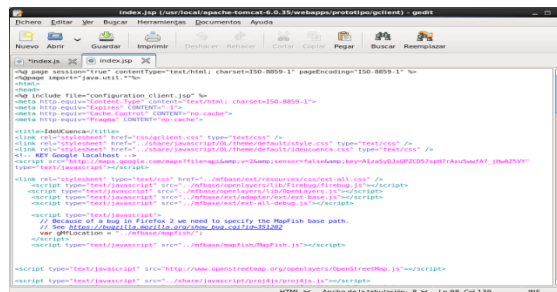


Figura III. 93. Referencia a las librerías utilizadas
Fuente: Los Autores

Establecer las proyecciones con las cuales trabajará el Visor Cuenca es importante que la cartografía este en la misma proyección y sistemas de coordenadas. Para ello muchas veces es necesario reprojectar las capas de información antes de integrarla.

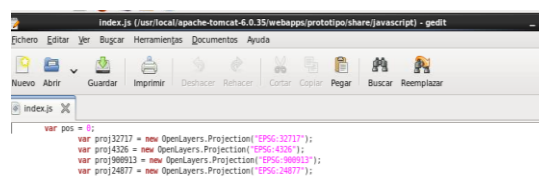


Figura III. 94. Definición de la proyección
Fuente: Los Autores

Este objeto define la extensión del mapa el cual contiene las capas y los controles a mostrarse, como se muestra en la figura III.95.

```
var coordMax = new OpenLayers.Bounds(-9580761.6242, -544824.2220, -8349908.8628, 156706.3181);
```

Figura III. 95. Definición del Bound

Fuente: Los Autores

Permite acceder a las distintas vistas de capas que proporciona google maps, como se muestra en la figura III.96.

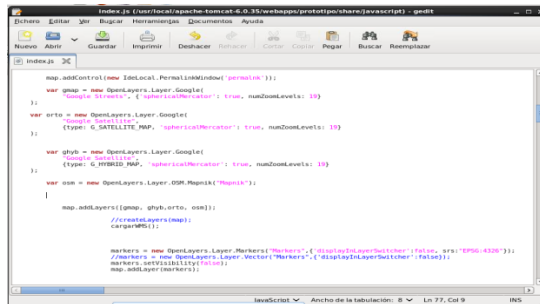


Figura III. 96. Cargar mapas de google

Fuente: Los Autores

Permite añadir controles del mapa para interactuar con el mapa, como se muestra en la figura III.97.

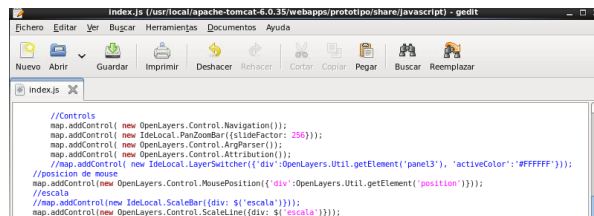


Figura III. 97. Controles del mapa

Fuente: Los Autores

Permite añadir referencia del mapa, como se muestra en la figura III.98.

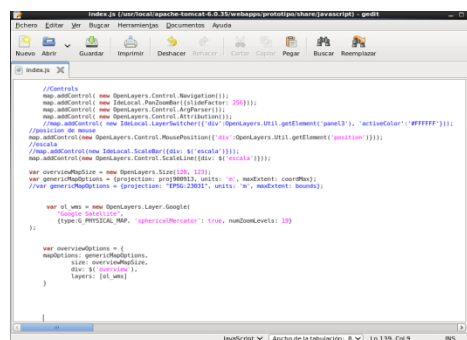


Figura III. 98. Referencia del mapa

Fuente: Los Autores



Figura III. 108. Path de aideucuenca.map
Fuente: Los Autores

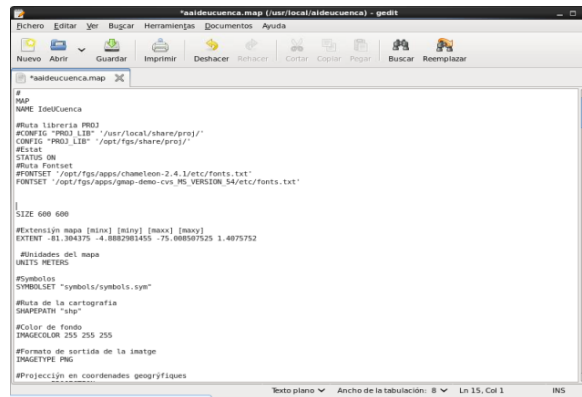


Figura III. 109. Configuración de aideucuenca.map
Fuente: Los Autores

En la figura III.110 se indica la ubicación del archivo configuración_editor.jsp que se encuentra dentro de **/usr/local/apache-tomcat-0.35/webapp/prototipo/edición/configuracion_editor.jsp**, se debe abrir el archivo y se debe cambiar las rutas de donde se encuentra la aplicación instalada.

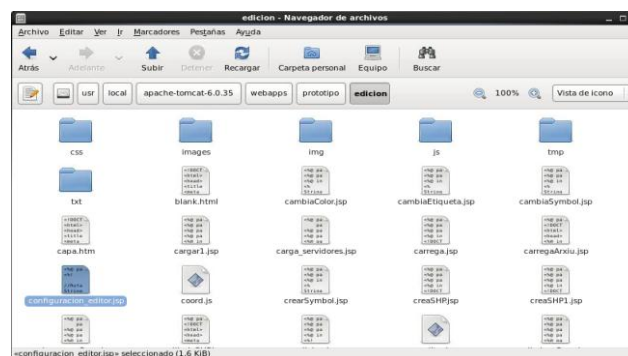
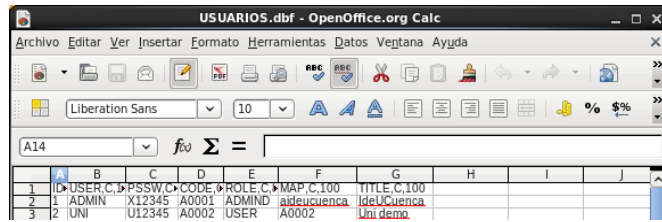


Figura III. 110. Path configuracion_editor.jsp
Fuente: Los Autores

En el archivo usuarios.dbf se encuentran todos los usuarios quienes serán autorizados para gestionar las capas, como se muestra en la figura III.111.



The screenshot shows a spreadsheet window titled 'USUARIOS.dbf - OpenOffice.org Calc'. The spreadsheet has columns labeled A through J and rows 1 through 3. The data is as follows:

	A	B	C	D	E	F	G	H	I	J
1	ID	USER	C	PSW	CODE	ROLE	C	MAP	C	TITLE
2	1	ADMIN	X12345	A0001	ADMIN	ideUcuenca	ideUcuenca			
3	2	UNI	UI2345	A0002	USER	A0002	Uni demo			

Figura III. 111. Archivo Usuarios.dbf

Fuente: Los Autores

Estructura de directorios

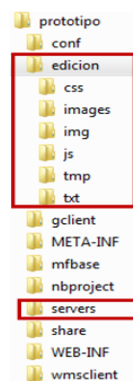


Figura III. 112. Directorios aplicación web

Fuente: Los Autores

Directorio raíz “**Edición**”: Se encuentran todos los JSP (Java Server Pages) y JS (JavaScripts) necesarios para el funcionamiento de la aplicación.

Directorio “**Edicion/css**”: Se encuentran todos los archivos CSS (Cascading Style Sheets) para dar estilo a la aplicación.

Directorios “**Edicion/images**” y “**Edicion/img**”. Contiene imágenes y utilizadas en la aplicación.

Directorio “**Edicion/JS**”. Contiene archivos JS secundarios, que contiene efectos utilizados por la aplicación (Ejemplo: Paleta de Colores web).

Directorio “**Edicion/TMP**”. Directorio utilizado para escribir los archivos ZIP de descarga de capas.

Directorio “**Edicion/TXT**”. Directorio utilizado para escribir archivos txt que contiene información asociadas a los proyectos de Minnesota Mapserver.

Directorio raíz “**Servers**”: Contiene archivos JS con las capas que han sido publicadas por los usuarios. El nombre de estos archivos empieza por “**servidores_editor_xxx.js**” y son leídos de forma **automática** por las aplicaciones **Gclient**. También contiene archivos referentes a los servidores de mapas usados como capa de o Directorios aplicación WEB IdeUcuenca o Aplicaciones IdeUcuenca referencia.

Directorio raíz “**aidecuencia**”: Se encuentran todos los MAP (Mapfiles) de los proyectos de Minnesota Mapserver. Los templates HTML para mostrar información así como la tabla “Usuarios.dbf” usada por la aplicación web del editor.

Directorio “**aidecuencia/SHP**”: Dónde se escribe y guardan los archivos SHP creados por los usuarios.

Directorio “**aidecuencia/Symbols**”: Simbología para representar a los mapas.

Archivos principales para el funcionamiento del editor

Archivo “**Edición/edicion.jsp**”: Archivo inicial, una vez autenticado el usuario, que carga todos los componentes de la aplicación.

Librerías de Java incluidas dentro de “**idecuencia/WEB-INF**”, Contiene funciones para escribir y leer archivos SHP y MAP.

Archivo “**configuracion_editor.jsp**”. Carga las variables y rutas definidas por el administrador.

Archivos “**coords.js**”, “**funciones.js**” y “**editor.js**”. Contienen funciones Javascript referentes a coordenadas y al funcionamiento de la aplicación.

Archivos “**wz_jsgraphics.js**” y “**wz_tooltip.js**”: Contienen funciones para el dibujo de geometrías y etiquetas.

Archivo “**Edición/editor.js**”: Contiene todas las funciones que gestionan las diferentes acciones de la aplicación. Las principales funciones son:

- Función **ogcmap_writeln**: Escribe la ventana de mapa y barra de herramientas en el navegador web
- Función **ogcMap**: Inicializa la función del objeto mapa y de sus capas.
- Función **ogcmap_Layer**: Añade capas al mapa de forma dinámica.

Código y uso del editor

La figura III.113 presenta el archivo edicion.jsp que contiene la validación de las credenciales ingresadas permitiendo la autenticación de usuarios.

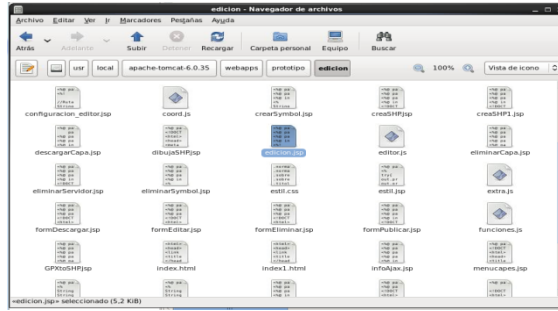


Figura III. 113. Path de edicion.jsp

Fuente: Los Autores

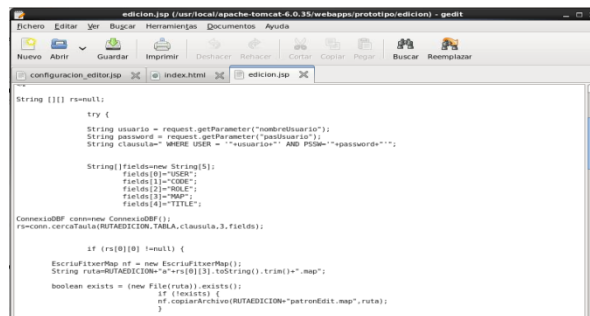


Figura III. 114. Autenticación de usuario

Fuente: Los Autores

En las figuras III.115, III.116, III.117., se muestra las funciones que se utilizan para la creación, eliminación y modificación de un shp respectivamente.

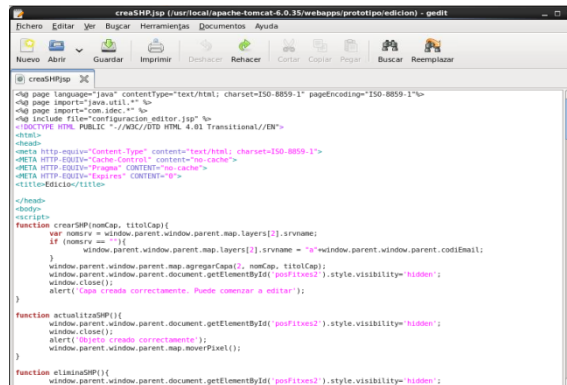
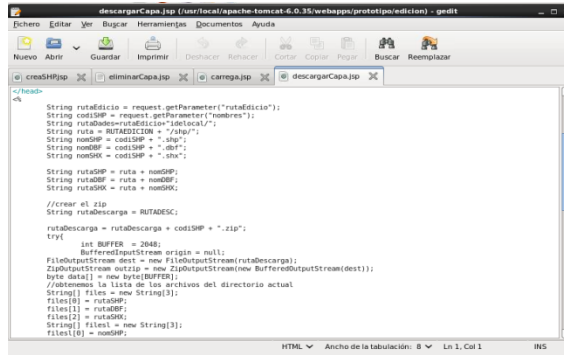


Figura III. 115. Creación de un shape

Fuente: Los Autores



```
String rutaEdicio = request.getParameter("rutaEdicio");
String codISP = request.getParameter("nombres");
String rutaDescarga=rutaEdicio+"descarga.";
String ruta = RUTAEDICION + ".shp";
String nomDF = codISP + ".shp";
String nomDB = codISP + ".dbf";
String nomDT = codISP + ".dxf";

String rutaGSP = ruta + nomGSP;
String rutaDBF = ruta + nomDBF;
String rutaDSC = ruta + nomDSC;

//crear el zip
String rutaDescarga = RUTADESC;
rutaDescarga = rutaDescarga + codISP + ".zip";
try {
    int BUFFER = 2048;
    BufferedOutputStream origina = null;
    FileOutputStream dest = new FileOutputStream(rutaDescarga);
    ZipOutputStream outzip = new ZipOutputStream(new BufferedOutputStream(dest));
    byte data[] = new byte[BUFFER];
    //obtenemos la lista de los archivos del directorio actual
    String[] files = new String[3];
    files[0] = rutaGSP;
    files[1] = rutaDBF;
    files[2] = rutaDSC;
    String[] files1 = new String[3];
    files1[0] = nomDF;
    files1[1] = nomDBF;
    files1[2] = nomDSC;
}
```

Figura III. 119. Descargar un shape
Fuente: Los Autores

En la siguiente figura se presenta la interfaz que va a permitir la gestion de capas.



Figura III. 120. Editor de objetos
Fuente: Los Autores

CAPÍTULO IV

4. ANÁLISIS ESTADÍSTICO Y COMPROBACIÓN DE LA HIPÓTESIS

En la actualidad existe una variabilidad de visores web open source, que ofrecen diferentes marcos que los desarrolladores pueden asumir a la hora de realizar su trabajo.

Es por esto que se desea proponer un visor web open source, que brinde las mejores prestaciones al momento de proporcionar información geográfica, para lo cual es necesario realizar un análisis basado en parámetros que reflejaran los puntos más relevantes de cada uno de ellos permitiendo la comparación entre los mismos.

4.1. Descripción de prototipo

Se desarrolló un prototipo con cada visor, el cual puede ser accedido por el usuario a través de un navegador, desplegando así la información geográfica requerida. En general el prototipo podrá prestar una serie de servicios dependiendo de su funcionalidad, estos pueden ofrecer diferentes servicios, tales como servicio de mapas Web, el cual se encarga de la visualización de las capas, servicio de features, este permite consultar información en relación a un objeto visualizado en el mapa, y el servicio de cobertura, que permite la localización de un punto en base a coordenadas, además permite el uso de herramientas de navegación como zooms, paneos, mapa de referencia, escala, barra de escalas, coordenadas, autenticación de usuarios para la gestión de capas.

4.2. Definición de los parámetros de comparación

En la tabla IV.XIV se definen los parámetros que colaboraran en el establecimiento de la comparación entre CartoWeb, OpenLayers y el Visor Cuenca, mismos que han sido establecidos según el criterio de los autores de la tesis.

Tabla IV.XIV. Parámetros de comparación

N°	PARÁMETRO	CONCEPTO
1	Compatibilidad	Compresión correcta entre la aplicación web o visor web con los navegadores más populares actualmente.
2	Soporte de Lenguaje	Capacidad de admitir un lenguaje de programación.
3	Instalación	Es la facilidad con la cual se llevará a cabo las acciones y disposiciones previas para que el visor web pueda funcionar correctamente.
4	Usabilidad	Un visor web con usabilidad es aquél que muestra todo de una forma clara y sencilla de entender por el usuario.
5	Integración de Servicios de mapas web	Capacidad de cargar o integrar capas desde servidores externos.
6	Gestión de capas	Permite el ingreso, modificación y eliminación de capas en el visor.
7	Exportación de archivos	Formato de archivos admitidos para exportar.
8	Soporte de servicios Open Geospatial Consortium.	Capacidad de disposición de servicios OGC que puede brindar cada visor.
9	Tipo de formatos de datos aceptado	Se refiere a los tipos de datos compatibles con los formatos de datos que soporta el visor.
10	Tiempo de ejecución	Lapso de tiempo transcurrido desde el momento en que el usuario hace la petición hasta que recibe la respuesta.
11	Plataformas Soportadas	Sistema operativo sobre el cual funciona el visor.

Elaborado por: Los Autores

4.3. Definición de los indicadores

En la tabla IV.XV se menciona los indicadores para cada parámetro ya establecido, los mismos que serán analizados con el fin de entregar información específica que servirá como estadística.

Tabla IV.XV. Indicadores

N°	PARÁMETRO	INDICADORES
1	Compatibilidad	Interfaz adaptable en el navegador.
2	Soporte de Lenguaje	Lenguaje de programación soportado.
3	Instalación	Facilidad de instalación.
4	Usabilidad	Calidad estética.
		Comprensibilidad de texto o información.
		Soporte de herramientas de navegación.
5	Integración de WMS	Integración de capas.
6	Gestión de capas	Ingreso.
		Modificación.
		Eliminación.
7	Exportación de archivos	Archivos admitidos para exportar.
8	Soporte de servicios OGC.	WMS
		WFS
		WCS
9	Tipo de formatos de datos aceptado	Vectorial
		Ráster
10	Tiempo de ejecución	Tiempo de respuesta.
11	Plataformas soportadas	Compatibilidad con Sistemas Operativos.

Elaborado por: Los Autores

4.4. Criterios de evaluación

La siguiente tabla muestra los valores cuantitativos, cualitativos que serán otorgados a cada parámetro.

Tabla IV.XVI. Criterios de evaluación General

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	Nulo	Mínimo	Medio	Medio Alto	Máximo
	Muy difícil	Difícil	Medio	Fácil	Muy Fácil
	No	-	-	-	Si

Elaborado por: Los Autores

Para la valorización porcentual se hizo uso de la siguiente fórmula:

$$Porc_c = \frac{pc}{pm} \cdot 100\%$$

$$Porc_o = \frac{po}{pm} \cdot 100\%$$

$$Porc_v = \frac{pv}{pm} \cdot 100\%$$

Pc = Puntaje acumulado de CartoWeb en el parámetro.

Po = Puntaje acumulado de OpenLayers en el parámetro.

Pv = Puntaje acumulado de visor cuenca en el parámetro.

Pm = Puntaje máximo sobre el que se califica el parámetro.

Porc_c = Porcentaje de aceptación CartoWeb.

Porc_o = Porcentaje de aceptación OpenLayers.

Porc_v = Porcentaje de aceptación Visor Cuenca.

4.5. Análisis de los parámetros de comparación

Este análisis se basará en la información recopilada mediante la investigación bibliográfica que se realizó, así como la observación que se llevará a cabo en el prototipo final que ha sido desarrollado en cada visor por parte de los autores de la tesis.

4.5.1. Compatibilidad

Se ha escogido los navegadores web más populares como Internet Explorer, Firefox Mozilla y Google Chrome, según un estudio sobre el ranking de navegadores más utilizados en los países de habla hispana en el 2012, en los cuales se observará la capacidad que tiene el visor web para funcionar de forma satisfactoria sin provocar irregularidades que afecten la visualización normal de la información espacial que se muestra con el visor.

Tabla IV.XVII. Criterios de evaluación para el parámetro Compatibilidad

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XVIII. Parámetro Compatibilidad

HERRAMIENTAS INDICADOR	CARTOWEB			OPENLAYERS			VISOR CUENCA		
INTERFAZ ADAPTABLE	INTERNET EXPLORER < 7	MOZILLA	GOOGLE CHROME	INTERNET EXPLORER < 7	MOZILLA	GOOGLE CHROME	INTERNET EXPLORER < 7	MOZILLA	GOOGLE CHROME
CRITERIO EVALUACIÓN DE	0	4	4	4	4	4	4	0	4
TOTAL	2,66			4			2,66		
PORCENTAJE	66,50%			100%			66,50%		

Elaborado por: Los Autores

En la figura IV. 121 se representa los porcentajes obtenidos en cada visor de forma gráfica.

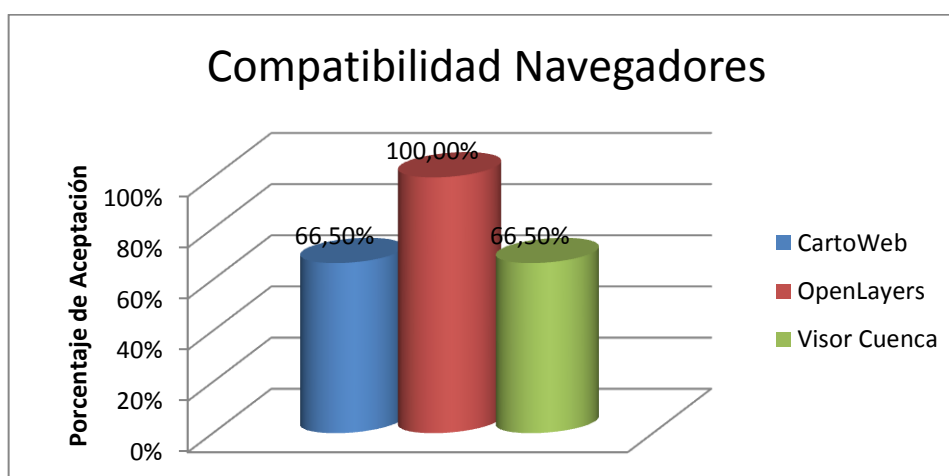


Figura IV. 121. Parámetro Compatibilidad Navegadores
Fuente: Los Autores

Interpretación

De acuerdo con el resultado obtenido en la tabla IV.XXI en cada uno de los visores web se deduce lo siguiente:

CartoWeb, no puede mostrar las capas de manera completa en navegadores Internet Explorer con versiones anteriores a 7.0, ya que solo muestra la mitad de la capa en el

marco asignado, en cambio es mostrado de forma completa en Mozilla y Google Chrome, trayendo consigo una aceptación del 66,50%.

OpenLayers, reflejan el 100% de cumplimiento en este parámetro porque las capas son visualizadas de forma completa en los navegadores establecidos.

Visor Cuenca, tiene una aceptación del 66,50%, ya que no puede mostrar las capas de manera completa en Mozilla, pero en Internet Explorer y google Chrome se visualiza sin ninguna dificultad.

4.5.2. Soporte de lenguaje

Es el lenguaje de programación sobre el cual está desarrollado el visor para proporcionar sus funcionalidades. Los lenguajes de programación son JavaScript, HTML, Java y Php de los cuales se obtuvo los resultados que se muestran en la tabla IV.XX.

Tabla IV.XIX. Criterios de evaluación para el parámetro Soporte de Lenguaje

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	Nulo	Mínimo	Medio	Medio Alto	Máximo

Tabla IV.XX. Parámetro Soporte de Lenguaje

HERRAMIENTAS INDICADOR	CARTOWEB				OPENLAYERS				VISOR CUENCA			
	JAVASCRIPT	HTML	JAVA	PHP	JAVASCRIPT	HTML	JAVA	PHP	JAVASCRIPT	HTML	JAVA	PHP
LENGUAJE	JAVASCRIPT	HTML	JAVA	PHP	JAVASCRIPT	HTML	JAVA	PHP	JAVASCRIPT	HTML	JAVA	PHP
CRITERIO DE EVALUACIÓN	4	4	0	4	4	1	0	0	4	4	4	0
TOTAL	3				1,25				3			
PORCENTAJE	75%				31,25%				75%			

Elaborado por: Los Autores

En la figura IV.122 se representa los porcentajes obtenidos en cada visor de manera gráfica.

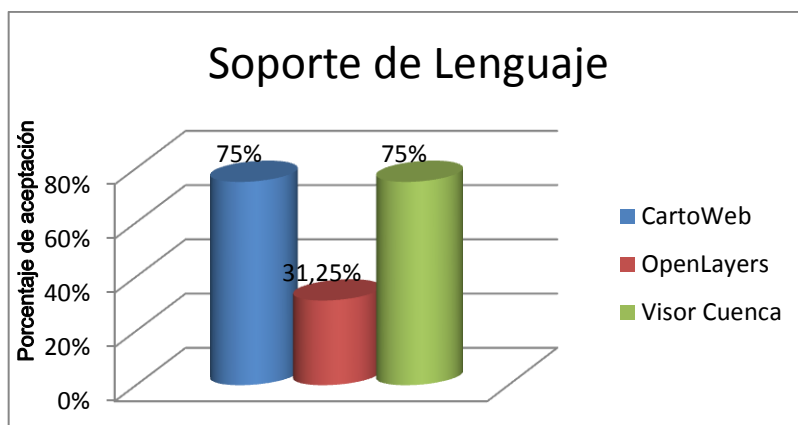


Figura IV.122. Parámetro Soporte de Lenguaje
Fuente: Los Autores

Interpretación

La tabla IV.XX indica los resultados arrojados en porcentajes en cada visor permitiendo deducir lo siguiente:

CartoWeb, necesita de un dominio del lenguaje JAVASCRIPT y PHP, a la vez se requiere tener un conocimiento estable de HTML para hacer uso de las funcionalidades del visor, conllevando a un cumplimiento del 75%.

OpenLayers, requiere un dominio de JAVASCRIPT y HTML para la utilización del visor en este caso tiene un equivalente del 31,25% ya que no permite realizar diversas funcionalidades como exportar a pdf, imprimir, etc.

Visor Cuenca, es indispensable tener un conocimiento de JAVASCRIPT, HTML y además del lenguaje JAVA para la utilización de las distintas funcionalidades del visor, por ende el porcentaje de aceptación es del 75%.

4.5.3. Instalación

La tabla IV.XXII contiene la calificación que obtuvo cada visor en valores cuantitativos, basándose en la facilidad de colocar, arreglar o disponer determinados elementos que lleven a la ejecución de los mismos.

Tabla IV.XXI. Criterios de evaluación para el parámetro Instalación

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	Muy difícil	Difícil	Medio	Fácil	Muy Fácil

Tabla IV.XXII. Parámetro Instalación

HERRAMIENTAS INDICADOR	CARTOWEB	OPENLAYERS	VISOR CUENCA
CRITERIO DE EVALUACIÓN	2	3	3
TOTAL	2	3	3
PORCENTAJE	50%	75%	75%

Elaborado por: Los Autores

En la figura IV.123 se representa los porcentajes obtenidos en cada visor de manera gráfica.

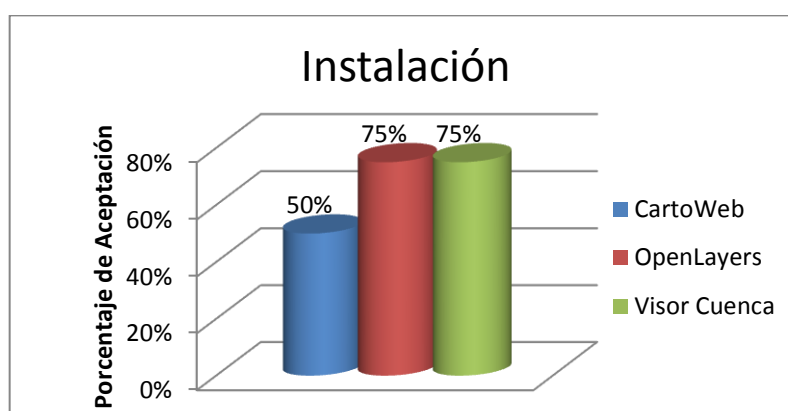


Figura IV. 123. Parámetro Instalación
Fuente: Los Autores

Interpretación

Por medio de la apreciación que se tiene en la tabla IV.XXII se deduce lo siguiente:

CartoWeb, tiene una forma de instalación más compleja con relación a los otros visores, ya que la instalación se la hace mediante comandos, además se debe tener instalado como requisitos previos Php, MapServer y el Servidor Web Apache, trayendo consigo una aceptación del 50%.

OpenLayers, tiene una aceptación del 75% ya que requiere descargar y vincular las librerías al archivo index.html para el funcionamiento del visor, además se debe de tener instalado como requisito previo MapServer y el Servidor Web Apache.

Visor Cuenca, refleja el 75% de cumplimiento ya que requiere la descarga y vinculación de la librería al archivo index.html, además se debe de tener instalado como requisito previo Jdk, MapServer y el Servidor Tomcat.

4.5.4. Usabilidad

La tabla IV.XXIV contiene la calificación obtenida por cada visor en valores cuantitativos, basándose en la facilidad que tienen los usuarios para interactuar con el visor de forma más fácil, cómoda e intuitiva posible, para lo cual se analizaron los resultados que arrojaron las encuestas (**Ver Anexo II**).

El indicador herramientas de navegación contiene las siguientes opciones a evaluarse como son: zoom, paneo, mapa de referencia, escala, impresión, coordenadas y volver al inicio. En caso de que el visor no cuente con una de las opciones de las herramientas de navegación se le dará una valoración de cero.

Tabla IV.XXIII. Criterios de evaluación para el parámetro Usabilidad

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	0 - 3 personas	4 - 8 personas	9 - 13 personas	14 - 18 personas	>=19 personas

Tabla IV.XXIV. Parámetro Usabilidad

HERRAMIENTAS INDICADOR	CARTOWEB			OPENLAYERS			VISOR CUENCA		
	USABILIDAD	Calidad Estética	Comprensibilidad	Herramientas de navegación	Calidad Estética	Comprensibilidad	Herramientas de navegación	Calidad Estética	Comprensibilidad
CRITERIO EVALUACIÓN DE	4	4	0	0	4	0	4	4	4
TOTAL	2,66			1,33			4		
PORCENTAJE	66,50%			33,25%			100%		

Elaborado por: Los Autores

En la figura IV.124 se representa los porcentajes obtenidos en cada visor de manera gráfica.

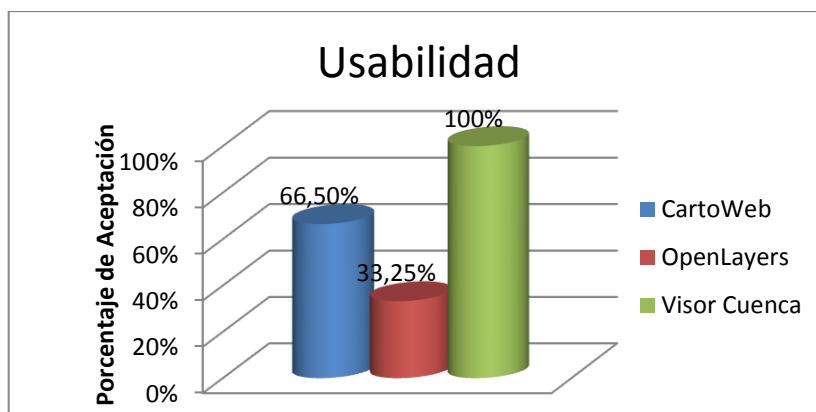


Figura IV. 124. Parámetro Usabilidad
Fuente: Los Autores

Interpretación

La tabla IV.XXIV presenta el resultado obtenido por cada visor conllevando a concluir con lo siguiente:

CartoWeb, cuenta con una interfaz de texto comprensible para el usuario y una estructura ordenada de los componentes como iconos, títulos, botones, espacio de texto e imágenes, sin embargo no cuenta con todas las herramientas que le permiten al usuario la manipulación del mapa, es por esta razón tiene una aceptación del 66,50%.

OpenLayers, su equivalencia es del 33,25%, ya que tiene una interfaz de texto comprensible, pero tiene como punto desfavorable no contar con una calidad estética buena y además no dispone de todas las herramientas de navegación.

Visor Cuenca, refleja el 100% de cumplimiento, ya que proporciona todas las opciones mencionadas dando lugar a la interacción total del usuario con el mapa.

4.5.5. Integración de WMS

En este punto se observará si el visor web permite la integración del servidor de mapa google maps, este es un servicio que es accedido a través del internet, el cual proporciona o exhibe capas que pueden servir como capas base al visor, dando lugar a que las demás capas que se encuentran en el servidor local se superpongan sobre este.

Tabla IV.XXV. Criterios de evaluación para el parámetro Integración de WMS

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XXVI. Parámetro Integración de WMS

HERRAMIENTAS	CARTOWEB	OPENLAYERS	VISOR CUENCA
INDICADOR			
CRITERIO DE EVALUACIÓN	0	4	4
TOTAL	0	4	4
PORCENTAJE	0%	100%	100%

Elaborado por: Los Autores

En la figura IV.125 se representa de forma gráfica los resultados obtenidos en cada visor.

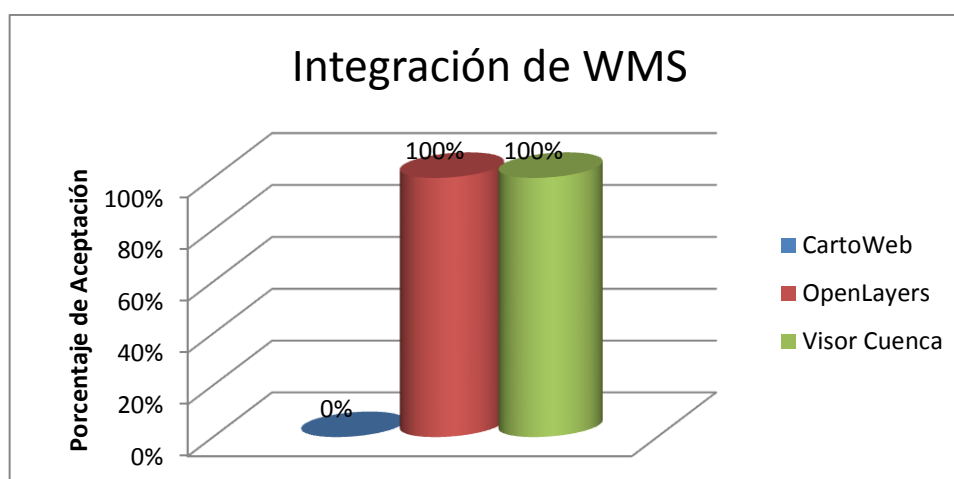


Figura IV. 125. Parámetro de Integración de WMS

Fuente: Los Autores

Interpretación

La tabla IV.XXVI presenta el puntaje obtenido en cada visor conllevando a concluir con lo siguiente:

CartoWeb, no ofrece la integración de capas desde google maps, es por esto que su nivel de aceptación es del 0%.

OpenLayers y el Visor Cuenca, si proporcionan este servicio permitiendo así que nuestra aplicación sea mejor, ya que brinda imágenes en distintas vistas, lo que equivale al 100%.

4.5.6. Gestión de capas

La gestión consiste en la creación de capas que luego serán cargadas por el administrador en el archivo .map para que puedan ser visualizadas, además de la modificación o eliminación de capas.

Tabla IV.XXVII. Criterios de evaluación para el parámetro Gestión de capas

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XXVIII. Parámetro Gestión de capas

HERRAMIENTAS INDICADOR	CARTOWEB			OPENLAYERS			VISOR CUENCA		
	INGRESO	MODIFICACIÓN	ELIMINACIÓN	INGRESO	MODIFICACIÓN	ELIMINACIÓN	INGRESO	MODIFICACIÓN	ELIMINACIÓN
GESTIÓN DE CAPAS									
CRITERIO DE EVALUACIÓN	0	0	0	0	0	0	4	4	4
TOTAL	0			0			4		
PORCENTAJE	0%			0%			100%		

Elaborado por: Los Autores

En la figura IV.126 se aprecia los porcentajes obtenidos de manera gráfica por cada visor.

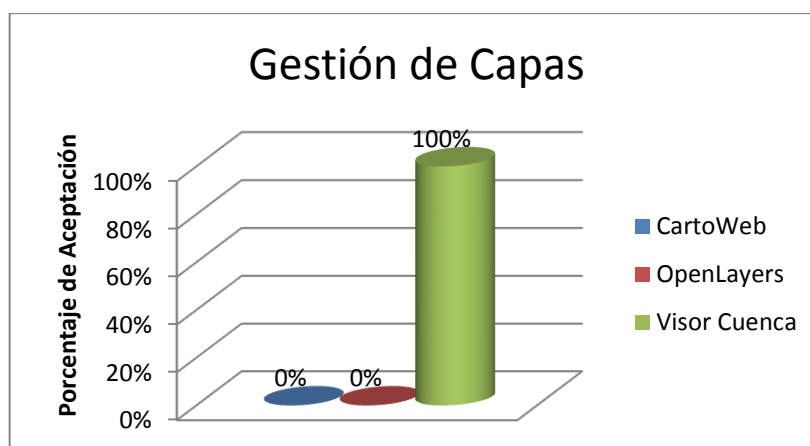


Figura IV.126. Parámetro de Gestión de Capas
Fuente: Los Autores

Interpretación

La tabla IV.XXVIII presenta los resultados obtenidos en cada visor conllevando a concluir con lo siguiente:

CartoWeb, no posibilita la creación, modificación, ni eliminación de capas constituyéndose esto en una desventaja, es por tal motivo que trae consigo una aceptación del 0%.

OpenLayers, no proporciona este parámetro para el usuario, es por eso que el porcentaje de aceptación es del 0%.

Visor Cuenca, si cuenta con la capacidad de crear, modificar, y eliminar capas.

Para realizar la gestión de capas el usuario previamente debe estar autorizado, es por eso que el nivel de aceptación que tiene este visor en este punto es del 100%.

4.5.7. Exportación de archivos

En la tabla IV.XXX se contempla el resultado obtenido en cuanto a la capacidad que tiene el visor de exportar un archivo en un formato específico, tales como pdf, csv o shp.

Tabla IV.XXIX. Criterios de evaluación para el parámetro Exportación de archivos

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XXX. Parámetro Exportación de archivos

HERRAMIENTAS INDICADOR	CARTOWEB			OPENLAYERS			VISOR CUENCA		
	PDF	CSV	SHP	PDF	CSV	SHP	PDF	CSV	SHP
TIPO DE ARCHIVOS									
CRITERIO DE EVALUACIÓN	4	4	0	0	0	0	4	0	4
TOTAL	2,66			0			2,66		
PORCENTAJE	66,50%			0%			66,50%		

Elaborado por: Los Autores

En la figura IV.127 se aprecia los porcentajes obtenidos en cada visor de manera gráfica.



Figura IV. 127. Parámetro de Exportación de Archivos

Fuente: Los Autores

Interpretación

De acuerdo con la tabla IV.XXX se llega a la conclusión que:

CartoWeb, permite transferir datos a un archivo pdf o csv, pero tiene como punto desfavorable que no permite la exportación en formatos shp, conllevando a que este tenga una aceptación del 66,50%.

OpenLayers, no permite la exportación de archivos en ningún formato, es por tal motivo que el nivel de aceptación que este tiene es del 0%.

Visor Cuenca, tiene una aceptación del 66,50%, ya que proporciona la exportación de datos a archivos pdf, además de generar la exportación de archivos del tipo shapefile (.shp).

4.5.8. Soporte de servicios

La tabla IV.XXXII muestra los resultados obtenidos en cada visor en cuanto a la capacidad de soporte de los 3 servicios establecidos en el OGC, tales como WMS que es el servicio encargado de mostrar mapas en imágenes, WFS es un servicio que va a permitir la búsqueda de objetos en el mapa, y WCS que permitirá el ingreso de coordenadas geográficas como datos para localizar objetos dentro de una cobertura.

Tabla IV.XXXI. Criterios de evaluación para el parámetro Soporte de Servicios

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XXXII. Parámetro Soporte de Servicios

HERRAMIENTAS INDICADOR	CARTOWEB			OPENLAYERS			VISOR CUENCA		
	WMS	WFS	WCS	WMS	WFS	WCS	WMS	WFS	WCS
SERVICIOS									
CRITERIO DE EVALUACIÓN	4	4	4	4	0	0	4	4	0
TOTAL	4			1,33			2,66		
PORCENTAJE	100%			33,25%			66,50%		

Elaborado por: Los Autores

En la figura IV.128 se aprecia los porcentajes obtenidos en cada visor de manera gráfica.

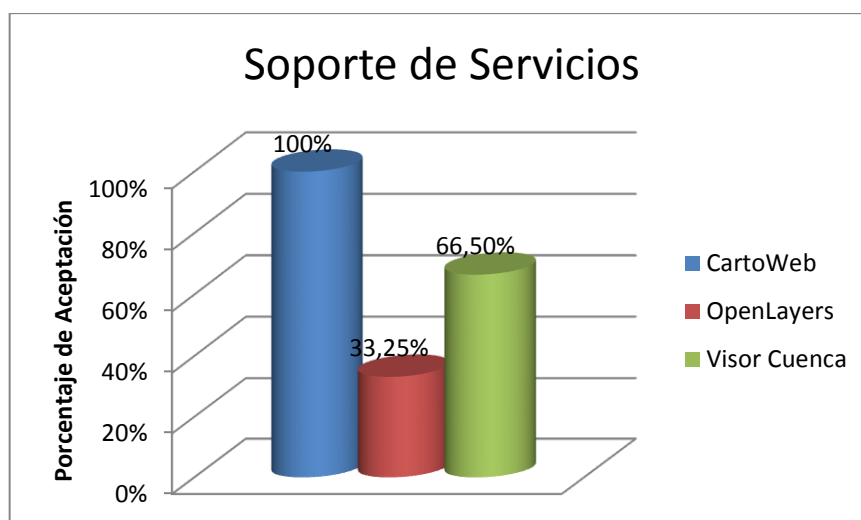


Figura IV. 128. Parámetro de Soporte de Servicios
Fuente: Los Autores

Interpretación

Se concluye con lo siguiente en base a los resultados que se presentan en la tabla IV.XXXII.

CartoWeb, refleja el 100% de cumplimiento en el ofrecimiento de servicios para el usuario, proporcionándole a este la visualización de información en el mapa por medio del WMS, así como también muestra la ubicación de un objeto en el mapa a través de WFS, además localiza un objeto a través de coordenadas usando WCS.

OpenLayers, solo proporciona el servicio WMS, que es la visualización de mapas en el ordenador, con un nivel de aceptación del 33,25%.

Visor Cuenca, este dispone de 2 prestaciones para el usuario, el servicio WMS, el cual proporciona la visualización de capas geográficas, así como también ofrece el servicio WFS, el cual permite realizar búsquedas de un objeto, dando lugar a la posición del objeto en el mapa, equivalente al 66,50% de los servicios que puede ofrecer este visor.

4.5.9. Tipos de formatos de datos aceptados

La tabla IV.XXXIV contiene el resultado en valores cuantitativos en cuanto a la aceptación del tipo de información de entrada que el visor va a comprender. En este parámetro se evalúa los dos tipos de formatos que funcionan en un sistema de información geográfica.

Tabla IV.XXXIII. Criterios de evaluación para el parámetro Tipo de Formatos Aceptados

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XXXIV. Parámetro Tipo de Formatos Aceptados

HERRAMIENTAS INDICADOR	CARTOWEB		OPENLAYERS		VISOR CUENCA	
	VECTORIAL	RÁSTER	VECTORIAL	RÁSTER	VECTORIAL	RÁSTER
FORMATOS DE DATOS						
CRITERIO DE EVALUACIÓN	4	4	4	4	4	4
TOTAL	4		4		4	
PORCENTAJE	100%		100%		100%	

Elaborado por: Los Autores

En la figura IV.129 se aprecia los porcentajes obtenidos en cada visor de manera gráfica.

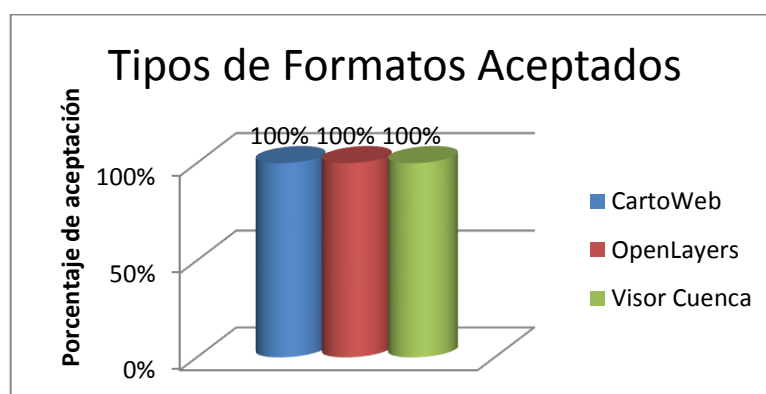


Figura IV. 129. Parámetro de Formatos Aceptados
Fuente: Los Autores

Interpretación

De acuerdo con los resultados que se presentan en la tabla IV.XXXIV, CartoWeb, OpenLayers y el Visor Cuenca poseen la capacidad para interpretar los 2 tipos de datos que son utilizados en un sistema que brinda información geográfica, cumpliendo los 3 visores con el 100% de aceptación.

4.5.10. Tiempo de ejecución

La tabla IV.XXXVI contiene el resultado en valores cuantitativos en cuanto al tiempo que transcurre desde el momento que el usuario hace alguna petición hasta cuando el visualiza la misma. Para esta prueba se utilizó un ancho de banda de 5374 Kbps.

Tabla IV.XXXV. Criterios de evaluación para el Parámetro Tiempo de Ejecución

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	≥ 10segundos	9.9 -7.5 segundos	7.4- 5.0 segundos	4.9-2.5- segundos	2.4-1-0 segundos

Tabla IV.XXXVI. Parámetro Tiempo de Ejecución

HERRAMIENTAS	CARTOWEB	OPENLAYERS	VISOR CUENCA
INDICADOR			
CRITERIO DE EVALUACIÓN	3	1	1
TOTAL	3	1	1
PORCENTAJE	75%	25%	25%

Elaborado por: Los Autores

En la figura IV.130 se representa de manera gráfica el porcentaje obtenido en cada visor.

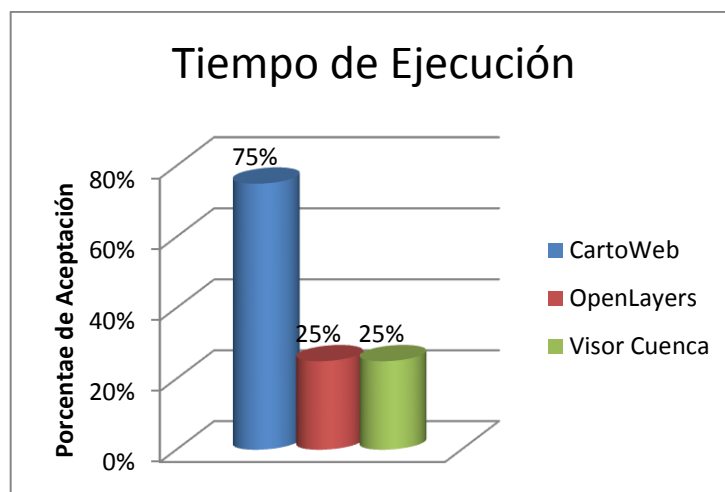


Figura IV. 130. Parámetro de Tiempo de Ejecución
Fuente: Los Autores

Interpretación

La tabla IV.XXXVI presenta los resultados que se obtuvieron en este parámetro, por lo que se concluye con lo siguiente:

CartoWeb, se demora menos que el Visor Cuenca y OpenLayers al momento de cargar todos los componentes, reflejando el 75% de aceptación.

Visor Cuenca y OpenLayers, tienen una aceptación del 25%, ya que se demora más en cargar los componentes de todo el sitio.

4.5.11. Plataformas soportadas

La tabla IV.XXXVIII contiene el resultado en valores cuantitativos en cuanto a la capacidad que tiene el visor para poder funcionar adecuadamente bajo los sistemas operativos como Windows, gnu/Linux y Unix.

Tabla IV.XXXVII. Criterios de evaluación para el parámetro Plataformas Soportadas

CUANTITATIVA	0	1	2	3	4
CUALITATIVA	No	-	-	-	Si

Tabla IV.XXXVIII. Parámetro Plataformas Soportadas

HERRAMIENTAS INDICADOR	CARTOWEB			OPENLAYERS			VISOR CUENCA		
	WINDOWS	GNU/LINUX	UNIX	WINDOWS	GNU/LINUX	UNIX	WINDOWS	GNU/LINUX	UNIX
PLATAFORMAS SOPORTADAS									
CRITERIO DE EVALUACIÓN	4	4	4	4	4	4	4	4	4
TOTAL	4			4			4		
PORCENTAJE	100%			100%			100%		

Elaborado por: Los Autores

En la figura IV.131 se representa de manera gráfica el porcentaje obtenido en cada visor.

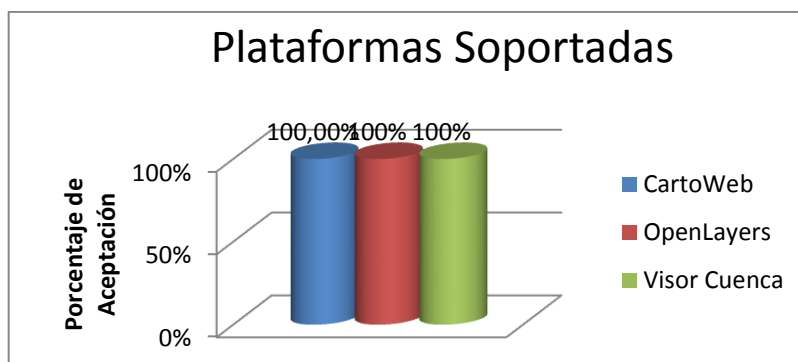


Figura IV. 131. Parámetro de Plataformas Soportadas

Fuente: Los Autores

Interpretación

La tabla IV.XXXVIII refleja la calificación en porcentajes que obtuvo cada visor, lo cual refleja que los 3 visores tienen la capacidad de ser ejecutados sobre los 3 sistemas operativos dando una equivalencia de aceptación del 100%.

4.6. Resumen general

Luego del análisis estadístico que se realizó a cada parámetro se procede a establecer un resumen general como se muestra en la tabla IV.XXXIX cuyo objetivo es recopilar los datos de las tablas individuales de cada parámetro con excepción del parámetro Usabilidad.

Tabla IV.XXXIX. Comparativa General - Prototipo

PARÁMETRO	CARTOWEB	OPENLAYERS	VISOR CUENCA
Compatibilidad	2,66	4	2,66
Similitud	3	1,25	3
Instalación	2	3	3
Integración de WMS.	0	4	4
Gestión de capas	0	0	4
Exportación de archivos	2,66	0	2,66
Soporte de servicios	4	1,33	2,66
Tipo de formatos de datos aceptado	4	4	4
Tiempo de ejecución.	3	1	1
Plataformas Soportadas	4	4	4
TOTAL	25,32	22,58	30,98

Elaborado por: Los Autores

La tabla IV.XL muestra la suma total de los parámetros evaluados en el prototipo y el resultado obtenido a través de las encuestas.

Tabla IV.XL. Suma Total del Prototipo y Encuesta

PARÁMETRO	CARTOWEB	OPENLAYERS	VISOR CUENCA
Suma total del Prototipo	25,32	22,58	30,98
Suma total de la encuesta	2,66	1,33	4

Elaborado por: Los Autores

En base a los resultados arrojados en la tabla IV.XL se procede a evaluar cada total sobre el 50% para obtener el resultado final del análisis estadístico que nos permitirá determinar el visor más adecuado.

Tabla IV.XLI. Comparativa General

PARÁMETRO	CARTOWEB	OPENLAYERS	VISOR CUENCA
Porcentaje total del Prototipo	31,65%	28,23%	38,73%
Porcentaje total de la encuesta	33,25%	16,63%	50%
PORCENTAJE TOTAL	64,90%	44,86%	88,73%

Elaborado por: Los Autores

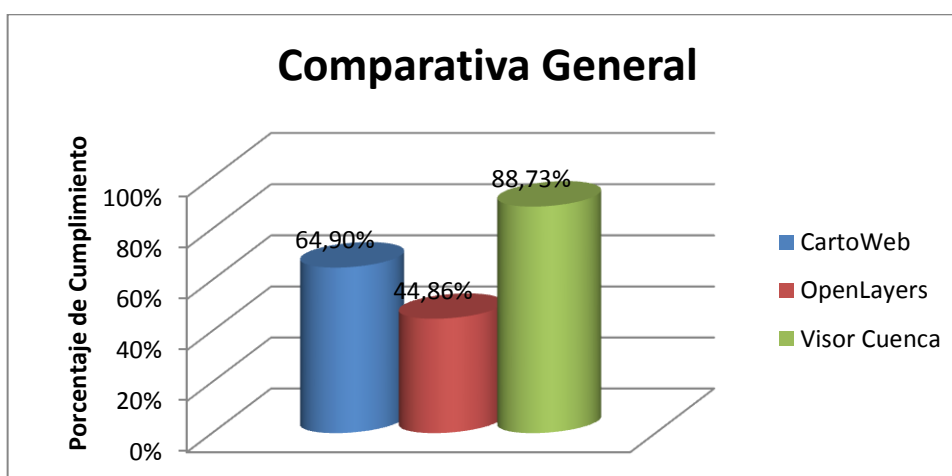


Figura IV. 132. Resumen de la Comparativa General entre CartoWeb, OpenLayers y Visor Cuenca

Fuente: Los Autores

Interpretación

Observando la figura IV.132 se puede determinar que el Visor Cuenca obtuvo el 88,73% superando de forma significativa a CartoWeb y OpenLayers, ya que CartoWeb alcanzó el 64,90% y OpenLayers consiguió el 44,86%, esto significa que para visualizar la información espacial en una infraestructura de datos espaciales se utilizará el Visor Cuenca.

4.7. Comprobación de la hipótesis

4.7.1. Hipótesis

El análisis estadístico de visores web permitirá establecer el más adecuado para mostrar información espacial en una infraestructura de datos espaciales.

4.7.2. Tipo de hipótesis

La Hipótesis de esta investigación es de tipo Causa - Efecto.

Causa: El análisis estadístico de visores web.

Efecto: Establece el más adecuado para mostrar información espacial en una infraestructura de datos espaciales.

4.7.3. Población y Muestra

La población son los estudiantes de octavo, noveno y décimo semestre que están en capacidad de evaluar la funcionalidad de los visores web en este caso el total de la misma es 119.

Para la comprobación de la hipótesis se realizará 23 encuestas de acuerdo al valor calculado de la muestra (**Ver Anexo III**).

4.7.4. Determinación de las variables

- **Variable Independiente:** Visores web.
- **Variable Dependiente:** Visualización de información espacial.

4.7.5. Operacionalización Conceptual

Tabla IV.XLII. Operacionalización Conceptual

VARIABLE	TIPO	CONCEPTO
Visores web	Variable Independiente	Un visor es una herramienta que permite su utilización en la construcción de una aplicación web que permitirá mostrar información geográfica a través de servicios.
Visualización de información espacial en la web.	Variable Dependiente	Despliegue de información sobre objetos o entidades abstraídos del espacio geográfico.

Elaborado por: Los Autores

4.7.6. Operacionalización Metodológica

Tabla IV.XLIII. Operacionalización Metodológica

HIPOTESIS	VARIABLES	PARAMETROS	TÉCNICAS	FUENTES DE VERIFICACIÓN
<p>El análisis estadístico de visores web permitirá establecer el más adecuado para mostrar información espacial en una infraestructura de datos espaciales.</p>	Visores Web	Herramienta de Visualización	Revisión de documentos	Internet Manuales
	Visualización de información espacial.	Compatibilidad	Observación	Prototipo
		Similitud	Observación	Prototipo
		Instalación	Observación	Prototipo
		Usabilidad	Observación	Encuesta
		Integración de WMS	Pruebas Observación	Prototipo
		Gestión de capas	Observación	Prototipo

		Exportación de archivos.	Observación	Prototipo
		Soporte de servicios OGC.	Observación	Prototipo
		Tipo de formatos de datos aceptado.	Recopilación de Información. Observación.	Prototipo
		Tiempo de ejecución.	Observación	Prototipo GTmetrix
		Plataformas soportadas	Recopilación de Información. Observación	Prototipo

Elaborado por: Los Autores

4.7.7. Demostración de la Hipótesis

Para demostrar la hipótesis de la presente investigación se utilizó como estadístico la técnica “Chi Cuadrado” el cual fue aplicado al parámetro Usabilidad (**Ver Anexo III**).

Gráfico

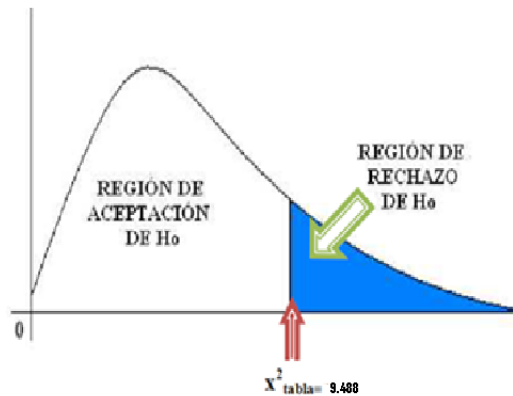


Figura IV. 133. Demostración de la Hipótesis
Fuente: Los Autores

Decisión

Del cálculo estadístico que se realizó en el Anexo III se definen los siguientes resultados:

x^2 **calculado**: Es el valor calculado en base a las encuestas realizadas.

x^2 **tabla**: Es el valor definido por la tabla de la técnica de “Chi Cuadrado”.

$$x^2_{\text{calculado}} = 34.96$$

$$x^2_{\text{tabla}} = 9.488$$

Entonces: $x^2_{\text{calculado}} > x^2_{\text{tabla}}$

Entonces se concluye de acuerdo a la figura IV.133 que el análisis estadístico de visores web permitió seleccionar el más adecuado para mostrar información espacial en una infraestructura de datos espaciales”, siendo seleccionado el Visor web de la Universidad de Cuenca como el más adecuado.

CAPÍTULO V

5. IMPLEMENTACIÓN DEL GEOPORTAL

El presente capítulo, contiene el desarrollo del geoportal para la ESPOCH, con el visor web seleccionado en el capítulo IV utilizando la metodología Programación Extrema (XP).

5.1. Metodología Programación Extrema (XP)

El escoger adecuadamente la metodología a seguir durante el desarrollo de software determina el éxito o no del proyecto, ya que este se encarga de establecer el conjunto de políticas, reglas, procedimientos que especifica la secuencia de pasos a seguir para tener el producto o terminación del mismo. Para esto se hizo uso de la metodología ágil XP que consta de 4 fases que son las siguientes:

- Planificación
- Diseño
- Desarrollo
- Prueba

5.1.1. Planificación

Esta es la primera etapa del proyecto en donde interviene la interacción del cliente y los desarrolladores para descubrir los requerimientos del sistema. Además de establecer las historias de usuarios, número de iteraciones, velocidad del proyecto.

5.1.1.1. Visión del sistema

El geoportal permitirá que los usuarios obtengan conocimientos y usen la información geográfica según sean sus necesidades.

5.1.1.2. Descripción del sistema

Es una aplicación web que tiene como objetivo la difusión de datos de la provincia de Chimborazo.

Desarrollada bajo herramientas libres como fgs-mapserver, Apache Tomcat, Joomla, el visor Cuenca y GeoNetwork.

Es posible visualizar las zonas deseadas a distintas escalas mediante la herramienta de zoom de alejamiento o de aproximación. Proporciona el servicio google maps en sus diferentes formatos como Mapa, Satélite, híbrido, OSM, Blanco y superpuestos a esta imagen, los datos vectoriales que correspondan.

Asimismo, la aplicación ofrece entre otras, herramientas de visualización (desplazamiento por la imagen, coordenadas, barra de escala), búsqueda e impresión.

Para su ejecución basta con un navegador estándar de internet.

5.1.1.2.1. Requerimientos

Requerimientos funcionales

En la tabla V.XLIV se aprecia una descripción de los requisitos que el sistema deberá cumplir.

Tabla V.XLIV. Requerimientos Funcionales

ID	DESCRIPCIÓN
Req. 1	Instalar herramientas de software.
Req. 2	Georeferenciar los shapes.
Req. 3	Crear archivo mapfile.
Req. 4	Crear leyenda que contendrá las capas de la provincia de Chimborazo.
Req. 5	Publicar información general de la IDE.
Req. 6	Integrar el visor de mapas.
Req. 7	Integrar el editor de mapas.
Req. 8	Integrar el catálogo de datos.
Req. 9	Visualizar galería de fotos.

Elaborado por: Los Autores

Requerimientos no funcionales

Especifican propiedades del sistema como restricciones de ambiente y desarrollo, dependencias de plataformas, mantenibilidad, confiabilidad, rendimiento, escalabilidad, interfaces, y usabilidad.

5.1.1.3. Historias de usuario

Las historias de usuario tienen el mismo fin que los casos de uso con la diferencia de que éstos son escritos por el propio usuario, quien conoce de los requerimientos de la aplicación, estas historias no tiene muchos detalles, son concretas.

Son pequeños textos en los que el cliente describe una actividad que realizará la aplicación, la redacción de los mismos se realiza bajo la terminología del cliente de forma, clara y sencilla.

Puntos Estimados: 1 punto = 1 día.

1ra Historia: Instalación de herramientas de software.

Tabla V.XLV. Tarjeta Historia de usuario 1

Historia de Usuario	
Numero: 01	
Nombre: Instalar herramientas de software.	
Iteración Asignada: 1	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos Estimados: 20	Puntos reales: 15
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: Permite a los usuarios ingresar a la pantalla de inicio del geoportal. Para la construcción del mismo se utilizará las herramientas Joomla y MySQL, para la gestión de contenido del geoportal para cumplir con el requerimiento. Este requisito conlleva la instalación previa de herramientas como el servidor MapServer, Apache, Jdk, Tomcat. Las salidas van dirigidas a los usuarios sin ninguna restricción.	

Elaborado por: Los Autores

2da Historia: Georeferenciar los shapes.

Tabla V.XLVI. Tarjeta Historia de usuario 2

Historia de Usuario	
Numero: 02	
Nombre: Georeferenciar los shapes para que puedan ser visualizadas o difundidas como información espacial en la WEB.	
Iteración Asignada: 2	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos Estimados: 10	Puntos Reales: 15
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: El geoportal permitirá el acceso a información espacial en la WEB. Por medio de la dirección http://ide.esPOCH.edu.ec , luego dar clic en la opción Visor de Mapas del menú Recursos que redirigirá la aplicación al visor, las herramientas utilizadas serán Tomcat, FGS mapserver, y el visor Cuenca para cumplir con el requerimiento.	

Elaborado por: Los Autores

3ra Historia: Crear archivo mapfile.

Tabla V.XLVII. Tarjeta Historia de usuario 3

Historia de Usuario	
Numero: 03	
Nombre: Crear archivo mapfile.	
Iteración Asignada: 2	
Prioridad de Negocio: Alta	Riesgo de desarrollo: Alta
Puntos Estimados: 3	Puntos reales: 3
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: Este archivo contiene la descripción de las capas y como serán procesadas por el servidor de mapas MapServer para luego ser visualizadas en el visor dependiendo de los parámetros ingresados en el mismo, todo esto ingresando la dirección http://ide.espoch.edu.ec , en donde luego debe dar clic en visor de mapas en el menú Recursos, que redirigirá la aplicación al visor.	

Elaborado por: Los Autores

4ta Historia: Crear leyenda que contendrá las capas de la provincia de Chimborazo.

Tabla V.XLVIII. Tarjeta Historia de usuario 4

Historia de Usuario	
Numero: 04	
Nombre: Crear leyenda que contendrá las capas de la provincia de Chimborazo.	
Iteración Asignada: 3	
Prioridad de Negocio: Alta	Riesgo de desarrollo: -
Puntos Estimados: 7	Puntos reales: 7
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: Representa las capas contenidas en la vista, permitiendo controlar su estado de visualización.	

Elaborado por: Los Autores

5ta Historia: Publicar información general de la IDE

Tabla V.XLIX. Tarjeta Historia de usuario 5

Historia de Usuario	
Numero: 05	
Nombre: Publicar información general de la IDE.	
Iteración Asignada: 4	
Prioridad de Negocio: Alta	Riesgo de desarrollo: -
Puntos Estimados: 1	Puntos reales: 1
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: Muestra información general sobre la IDE en el geoportal.	

Elaborado por: Los Autores

6ta Historia: Integrar el visor de mapas.

Tabla V.L. Tarjeta Historia de usuario 6

Historia de Usuario	
Numero: 06	
Nombre: Integrar el visor de mapas.	
Iteración Asignada: 4	
Prioridad de Negocio: Alta	Riesgo de desarrollo: -
Puntos Estimados: 1	Puntos reales: 1
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: El visor de mapas será integrado dentro del geoportal.	

Elaborado por: Los Autores

7ma Historia: Integrar el editor de mapas en el geoportal.

Tabla V.LI. Tarjeta Historia de usuario 7

Historia de Usuario	
Numero: 07	
Nombre: Integrar el editor de mapas.	
Iteración Asignada: 4	
Prioridad de Negocio: Media	Riesgo de desarrollo: -
Puntos Estimados: 1	Puntos reales: 1
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: Integración del editor de mapas en el geoportal.	

Elaborado por: Los Autores

8va Historia: Integrar el catálogo de datos en el geoportal.

Tabla V.LII. Tarjeta Historia de usuario 8

Historia de Usuario	
Numero: 08	
Nombre: Integrar el catálogo de datos.	
Iteración Asignada: 4	
Prioridad de Negocio: Alta	Riesgo de desarrollo: -
Puntos Estimados: 5	Puntos reales: 5
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción: El geoportal permitirá la consulta de Metadatos sobre las capas publicadas en el visor, mostrando información tales como fecha de creación o publicación, sistema de referencia, autor del metadato, entre otros. Esto lo hace mediante el uso de Tomcat y geonetwork.	

Elaborado por: Los Autores

9na Historia: El geoportal permitirá visualizar fotos en la galería.

Tabla V.LIII. Tarjeta Historia de usuario 9

Historia de Usuario	
Numero: 09	
Nombre: El geoportal permitirá visualizar fotos en la galería.	
Iteración Asignada: 4	
Prioridad de Negocio: Media	Riesgo de desarrollo: -
Puntos Estimados: 4	Puntos reales: 4
Programador responsable: Damaris Chinlli, Lorena Morales.	
Descripción El geoportal permitirá visualizar una galería de fotos, por medio de la utilización de la herramienta bowslieshow.	

Elaborado por: Los Autores

5.1.1.4. Iteraciones

Durante cada iteración se cierran varias historias, lo que hace que toda iteración añada un valor tangible para el cliente.

Plan de iteraciones

El desarrollo del proyecto consta de 4 iteraciones sumando como total 9 historias de usuario.

La primera iteración contiene las siguientes historias de usuario:

- **Req. 1.** Instalar herramientas de software.

La segunda iteración contiene las siguientes historias de usuario:

- **Req. 2.** Georeferenciar los shapes.
- **Req. 3.** Crear archivo mapfile.

La tercera iteración contiene las siguientes historias de usuario:

- **Req. 4.** Crear leyenda que contendrá las capas de la provincia de Chimborazo.

La cuarta iteración contiene las siguientes historias de usuario:

- **Req. 5.** Publicar información general de la IDE.
- **Req. 6.** Integrar el visor de mapas.
- **Req. 7.** Integrar el editor de mapas.

- **Req. 8.** Integrar el catálogo de datos.
- **Req. 9.** Visualizar galería de fotos.

Plan de Entrega

El plan de entrega se estableció mediante un acuerdo que se tuvo entre el cliente y los programadores.

A continuación se indica cómo se estableció el plan de entrega:

La iteración #1 contiene el proceso para el levantamiento del geoportal comenzando desde la instalación del sistema operativo Centos 6, apache, joomla y mysql que va a dar paso a la administración de contenidos para el sitio.

La iteración #2 conlleva la instalación de las herramientas como fgs-mapserver que es el servidor de mapas donde van a estar alojados las capas georeferenciadas en el directorio data, además del archivo mapfile que va a contener los parámetros que definen las capas disponibles en el servicio, el estilo con que se representarán, su simbología, formato en que se generará la imagen, el sistema de referencia.

En la iteración #3 se creará la leyenda que contendrá las capas de la provincia de Chimborazo contenidas en la vista, permitiendo controlar su estado de visualización.

La iteración #4 contiene información relacionada con la IDE, además de la integración del visor de mapas, editor de mapas, catálogo de datos, este último va hacer uso de la herramienta geonetwork para la creación de metadatos, además el usuario podrá visualizar la galería de fotos por medio del módulo o componente bowlieshow añadido en el gestor de contenidos joomla.

Velocidad del Proyecto

La velocidad del proyecto está dada por la sumatoria de los tiempos que se estiman sin tener en cuenta ningún tipo de interrupción para desarrollar cada historia de usuario.

Tabla V.LIV. Velocidad del proyecto

Velocidad del Proyecto		
Tiempo de Calendario	5 días a la semana - lunes a viernes.	
Equipo XP	2 personas	
Fecha de inicio: 6 Julio 2012	Fecha de fin: 19 de Octubre 2012	
Tiempo estimado de ejecución de historia de usuario		
Iteraciones	Numero de historia de usuario	Días calendario
Iteración 1	1	15
Iteración 2	2	18
Iteración 3	1	7
Iteración 4	5	10
Tiempo Estimado Aproximado días		50 días
Tiempo Estimado Aproximado semanas		10 semanas

Elaborado por: Los Autores

En la figura IV.134 se muestra el diagrama de velocidad del proyecto.

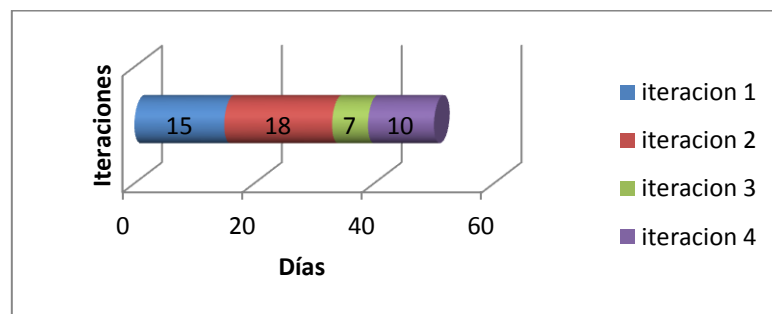


Figura IV. 134. Velocidad del proyecto

Fuente: Los Autores

5.1.2. Diseño

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. Se plasma el diseño en tarjetas CRC.

Tarjetas CRC

Ayudan al programador a identificar objetos involucrados en el sistema en conjunto entre todo el equipo basado en el desarrollo orientado a objetos. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. A continuación se muestran algunas tarjetas CRC de las clases más influyentes en nuestro proyecto.

La tabla V.LV indica la tarjeta Administrador donde muestra los atributos con los que cuenta como son: User, pass, y como métodos la creación, modificación y eliminación. Esta clase no depende de otra.

Tabla V.LV. Tarjeta CRC Administrador

Administrador	
User	
Pass	
Gestión	

Elaborado por: Los Autores

La siguiente tabla V.LVI contiene el objeto capa, donde muestra los atributos con los que cuenta como son: Id, Descripción, así mismo cuenta con métodos como crear, modificar y eliminar. Esta clase no depende de otra.

Tabla V.LVI. Tarjeta CRC Capa

Capa	
Id	
Descripción	
Referencia	
Gestión	

Elaborado por: Los Autores

5.1.2.1. Riesgos

Considerando que un posible riesgo se llegará producir en el futuro traería consigo cambios desfavorables que afectarían a la organización es por aquello que se debió tener en cuenta el análisis de riesgo.

A continuación se identifican los riesgos, determinación del impacto, determinación de la exposición del riesgo, prioridad del riesgo, y por último la hoja de gestión de riesgo en la que establece todos los valores de los parámetros que se especificaron con anterioridad, así como una posible solución a los mismos, todo esto con el fin de analizar y gestionar los posibles riesgos para así evitar que en el transcurso del desarrollo del geoportail se conviertan en un problema.

5.1.2.1.1. Identificación del riesgo

Para el análisis es necesario identificar cada uno de los riesgos, su categoría, consecuencias, probabilidad, impacto y exposición anexando una hoja de información de cada riesgo que contiene un plan de reducción, supervisión y gestión del riesgo.

En la tabla V.LVII se presenta la identificación de los posibles riesgos para la cual se utilizara la siguiente nomenclatura:

- RP: riesgos de proyecto
- RT: riesgos técnicos
- RN: riesgos de negocios

En la tabla que se indica a continuación se presenta la identificación de los riesgos potenciales que han sido definidos para el proyecto, en donde se describe un Id, la descripción del riesgo, la categoría y la consecuencia en caso de que llegará a producirse.

Tabla V.LVII. Identificación del Riesgo

ID	DESCRIPCIÓN DEL RIESGO	CATEGORÍA	CONSECUENCIA
R1	Los recursos no se asignan oportunamente.	RP	Retraso en el desarrollo de la aplicación.
R2	No cumplimiento con los tiempos establecidos para el desarrollo del proyecto.	RP	Retraso en la entrega de la aplicación.
R3	Error en la instalación y configuración de las herramientas softwares.	RT	Retraso en el funcionamiento del geoportal.
R4	Falta de experiencia del personal para el desarrollo de la aplicación.	RT	Retraso en el proyecto.

Elaborado por: Los Autores

5.1.2.1.2. Análisis de riesgos

En esta sección se describirán las tablas que servirán como referencia para determinar la valoración del riesgo, las probabilidades, y el impacto, de los riesgos indicados en la tabla V.LVII, permitiendo establecer la prioridad de los mismos.

Tabla V.LVIII. Valoración del Riesgo

RANGO DE PROBABILIDAD	DESCRIPCIÓN	VALOR
1% - 33%	Baja	1
34% - 67%	Media	2
68% - 99%	Alta	3

Elaborado por: Los Autores

La tabla V.LIX detalla la probabilidad de ocurrencia que se le asigna a cada posible riesgo.

Tabla V.LIX. Probabilidad

ID	PROBABILIDAD		
	%	VALOR	PROBABILIDAD
R1	40	2	Media
R2	35	2	Media
R3	40	2	Media
R4	35	2	Media

Elaborado por: Los Autores

Criterios de valoración del impacto

La tabla V.LX sirve como referencia para establecer el impacto del riesgo.

Tabla V.LX. Impacto del Riesgo

IMPACTO	RETRASO	IMPACTO TÉCNICO	COSTO	VALOR
Bajo	1 semana	Leve efecto en el desarrollo del proyecto.	< 1%	1
Moderado	2 semanas	Moderado efecto en el desarrollo del proyecto.	< 5%	2
Alto	1 mes	Severo efecto en el desarrollo del proyecto.	< 10%	3
Crítico	> 1 meses	Proyecto no puede ser culminado.	> 20%	4

Elaborado por: Los Autores

En la siguiente tabla se describe el valor del impacto asignado a cada riesgo.

Tabla V.LXI. Riesgo – Impacto

ID	IMPACTO	
	VALOR	IMPACTO
R1	2	Moderado
R2	3	Alto
R3	2	Moderado
R4	3	Alto

Elaborado por: Los Autores

Criterios de valoración de la exposición al riesgo

La tabla V.LXII sirve como referencia para establecer la exposición al riesgo, mediante un valor cuantitativo y representativo.

Tabla V.LXII. Exposición al riesgo

EXPOSICIÓN	VALOR	COLOR
Baja	1 o 2	Verde
Media	3 o 4	Amarillo
Alta	Mayor a 6	Rojo

Elaborado por: Los Autores

La tabla V.LXIII detalla la exposición al riesgo representado por un color dependiendo del valor, este valor se lo obtiene mediante la multiplicación del valor de impacto y de probabilidad.

Tabla V.LXIII. Impacto – Probabilidad

IMPACTO PROBAB	BAJO=1	MODERADO=2	ALTO=3	CRÍTICO=4
ALTA	3	6	9	12
MEDIA	2	4	6	8
BAJA	1	2	3	4

Elaborado por: Los Autores

La siguiente tabla fue construida basada en los valores obtenidos de las tablas anteriores.

Tabla V.LXIV. Resumen del Riesgo

ID	PROBABILIDAD			IMPACTO		EXPOSICIÓN AL RIESGO	
	%	Valor	Probabilidad	Valor	Impacto	Valor	Exposición
R1	40	2	Media	2	Moderado	4	Media
R2	35	2	Media	3	Alto	6	Alta
R3	40	2	Media	2	Moderado	4	Media
R4	35	2	Media	3	Alto	6	Alta

Elaborado por: Los Autores

A continuación se ordenará la tabla anterior de forma ascendente de acuerdo a la prioridad de riesgos e incluiremos el código de colores y la línea de corte para los riesgos de prioridad 1, porque son estos los que pueden convertirse en problemas.

Tabla V.LXV. Prioridades del Riesgo

IDENTIFICACIÓN	PRIORIDAD	EXPOSICIÓN
R2	1	6
R4	1	6
R1	2	4
R3	2	4

Elaborado por: Los Autores

5.1.2.1.3. Planeación y programación del riesgo

Tabla V.LXVI. Gestión del riesgo 1

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R1		FECHA: 23/7/2012	
Probabilidad: Media Valor: 2	Impacto: Moderado Valor: 2	Exposición: Media Valor: 4	Prioridad: 2
DESCRIPCIÓN: Los recursos no se asignan oportunamente.			
REFINAMIENTO: Causas: No existió una comunicación adecuada entre el usuario y los responsables del proyecto Consecuencias: Retraso en el desarrollo de la aplicación.			
REDUCCIÓN: Que exista una adecuada comunicación entre el cliente y el programador.			
SUPERVISIÓN: Que el ambiente de comunicación sea el más propicio entre el cliente y el responsable del proyecto.			
GESTIÓN: Reuniones para establecer acuerdos que agilicen el éxito del proyecto.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: Damaris Chinlli, Lorena Morales.			

Elaborado por: Los Autores

Tabla V.LXVII. Gestión del riesgo 2

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R2		FECHA: 23/7/2012	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 1
DESCRIPCIÓN: No cumplimiento con los tiempos establecidos para el desarrollo del proyecto.			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Falta de experiencia en el manejo de las herramientas. • Mala organización y asignación de tareas entre los desarrolladores. 			
Consecuencias:			
<ul style="list-style-type: none"> • Insatisfacción por no cumplimiento de tareas establecidas en el cronograma. • Inconformidad del usuario. 			
REDUCCIÓN:			
Establecer una relación más participativa entre los desarrolladores para que así haya un ambiente de dialogo y se determinen las tareas de manera adecuada.			
SUPERVISIÓN:			
Supervisar el cumplimiento de cada una de las actividades en los tiempos planificados.			
GESTIÓN:			
Reordenar o aumentar la asignación de tareas a cada desarrollador para que la aplicación sea entregada a tiempo			
ESTADO ACTUAL:			
Fase de reducción iniciada:		<input checked="" type="checkbox"/>	
Fase de supervisión iniciada:		<input type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Damaris Chinlli, Lorena Morales.			

Elaborado por: Los Autores

Tabla V.LXVIII. Gestión del riesgo 3

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R3		FECHA: 23/7/2012	
Probabilidad: Media Valor: 2	Impacto: Moderado Valor: 2	Exposición: Media Valor: 4	Prioridad: 2
DESCRIPCIÓN: Error en la instalación y configuración de las herramientas softwares.			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Poco conocimiento de los desarrolladores. 			
Consecuencias:			
<ul style="list-style-type: none"> • Retraso en el funcionamiento del geoportal. 			
REDUCCIÓN:			
Recopilar información y capacitarse sobre el tema.			
SUPERVISIÓN:			
Mantener un control sobre el cumplimiento de las actividades en el tiempo acordado.			
GESTIÓN:			
Capacitar a los desarrolladores.			
ESTADO ACTUAL:			
Fase de reducción iniciada:		<input checked="" type="checkbox"/>	
Fase de supervisión iniciada:		<input type="checkbox"/>	
Gestionando el riesgo:		<input type="checkbox"/>	
RESPONSABLES:			
Damaris Chinlli, Lorena Morales.			

Elaborado por: Los Autores

Tabla V.LXIX. Gestión del riesgo 4

HOJA DE GESTIÓN DEL RIESGO			
ID DEL RIESGO: R4		FECHA: 23/7/2012	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 1
DESCRIPCIÓN: Falta de experiencia del personal para el desarrollo de la aplicación.			
REFINAMIENTO: Causas: Falta de oportunidades. Consecuencias: Retraso en el proyecto.			
REDUCCIÓN: Analizar las necesidades de los desarrolladores.			
SUPERVISIÓN: Revisar que las tareas se cumplan y sean entregadas según los tiempos establecidos.			
GESTIÓN: Fomentar métodos de capacitación para los desarrolladores.			
ESTADO ACTUAL: Fase de reducción iniciada: <input checked="" type="checkbox"/> Fase de supervisión iniciada: <input type="checkbox"/> Gestionando el riesgo: <input type="checkbox"/>			
RESPONSABLES: Damaris Chinlli, Lorena Morales.			

Elaborado por: Los Autores

5.1.3. Desarrollo

5.1.3.1. Análisis del MapFile

Se crea un archivo MapFile (ideepoch.map), ya que es el archivo de configuración en donde se definen los parámetros necesarios tales como: el tipo de proyección, escala,

etiquetas y clases. Este archivo se encuentra situado en este directorio `opt/fgs/www/htdocs`.

5.1.3.2. Entrada de Dato

Luego de recopilar la información espacial en formato `.shp`, se hizo uso del software ArcGis para observar atributos, tipo, y asignar la correcta georeferenciación de las capas que serán publicadas sobre el mapa base de google maps.

Las capas de la provincia de Chimborazo están alojadas en el servidor de forma local, además se pueden visualizar capas provenientes de servidores externos como las capas que se encuentran en el servidor de la universidad de Cuenca.

5.1.3.3. Almacenamiento de la Información

La información espacial no está contenida en una base de datos común, esta información está incorporada dentro de archivos con formatos `.dbf`, los mismos que se encuentra en el servidor de mapas local, específicamente en el directorio `data` que está ubicado en `opt/fgs/www/htdocs`.

5.1.3.4. Implementación

A continuación se detallarán las actividades realizadas para la implementación de la solución, desde la instalación de las herramientas necesarias hasta la entrada de datos, y salida de datos para el levantamiento del geoportal.

Herramientas utilizadas

La elección de los componentes se realizó en base a investigación por parte de los autores, utilizándose como criterio de la elección, la gratuidad de las herramientas y que estas además hubieran ya sido probadas satisfactoriamente por otras personas, para lo cual fue necesaria una investigación en foros de sistemas de información geográfica.

A continuación se menciona las herramientas utilizadas:

- Centos 6.0.
- `fgs-mapserver_basic_5.6.3-fgs_9.5-linux-i386.bin`.

- Joomla_1.7.5-Spanish-Pack_Completo.tar.gz.
- jdk-6u17-linux-i586.bin.
- apache-tomcat-6.0.35.tar.gz.
- Visor Cuenca.
- GeoNetwork.

Configuración e instalación de herramientas

Para empezar se debe de tener conocimientos básicos para proceder con la instalación del sistema operativo Centos 6.0.

Luego de haber terminado con la instalación se realiza lo siguiente:

Instalación de MapServer- fgs-mapserver_basic_5.6.3-fgs_9.5-linux-i386.bin

Es importante la instalación de este paquete, ya que contiene el servidor de mapas MapServer así como las librerías geográficas que van a permitir que se proporcione el correcto despliegue o publicación de cartografía en el geoportal.

Para llevar a cabo el objetivo de instalar MapServer sobre Centos 6.0, se debe tomar en cuenta los siguientes pasos:

Descargar MapServer FGS y ejecutar el instalador, como se muestra en la figura V.135.

```
[root@ideepoch Escritorio]# sh ./fgs-mapserver_basic_5.6.3-fgs_9.5-linux-i386.bin
```

Figura V. 135. Ejecución de Instalador

Fuente: Los Autores

Al ejecutar MapServer por defecto tiene un path o ruta que utiliza como referencia para la instalación en este caso es **/opt/fgs**, como se muestra en la figura V.136.

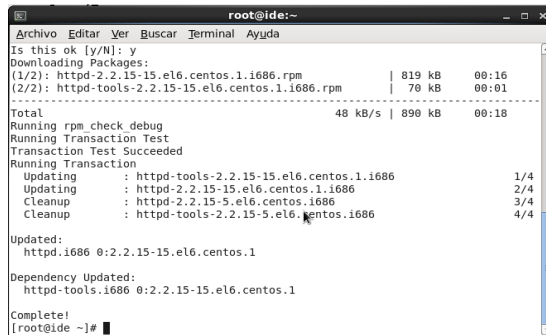
```
Please send any questions to <fgs@dmsolutions.ca>.  
To install the FGS suite, press ENTER.  
To cancel, press Control-C.  
Where do you want to install the FGS suite [/opt/fgs] ? /opt/fgs
```

Figura V. 136. Ubicación del directorio MapServer

Fuente: Los Autores

Instalación de Apache

Abra un terminal y ejecute el siguiente comando **yum install httpd**. Ingrese **y** para que proceda la descarga e instalación del paquete, como se muestra en la figura V.140.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
Is this ok [y/N]: y  
Downloading Packages:  
(1/2): httpd-2.2.15-15.el6.centos.1.i686.rpm | 819 kB 00:16  
(2/2): httpd-tools-2.2.15-15.el6.centos.1.i686.rpm | 70 kB 00:01  
-----  
Total | 48 kB/s | 890 kB 00:18  
Running rpm_check_debug  
Running Transaction Test  
Transaction Test Succeeded  
Running Transaction  
Updating : httpd-tools-2.2.15-15.el6.centos.1.i686 1/4  
Updating : httpd-2.2.15-15.el6.centos.1.i686 2/4  
Cleanup : httpd-2.2.15-5.el6.centos.i686 3/4  
Cleanup : httpd-tools-2.2.15-5.el6.centos.i686 4/4  
Updated:  
httpd.i686 0:2.2.15-15.el6.centos.1  
Dependency Updated:  
httpd-tools.i686 0:2.2.15-15.el6.centos.1  
Complete!  
[root@ide ~]#
```

Figura V. 140. Instalación de httpd
Fuente: Los Autores

Ingrese **http://localhost** para comprobar que se instaló correctamente, como se muestra en la figura V.141.

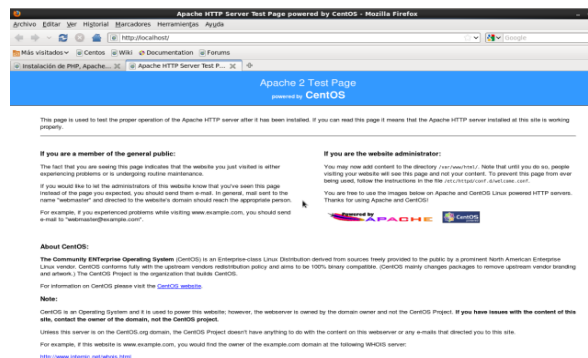
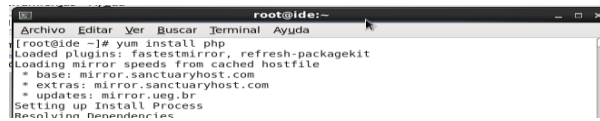


Figura V. 141. Prueba de httpd instalado
Fuente: Los Autores

Instalación de php

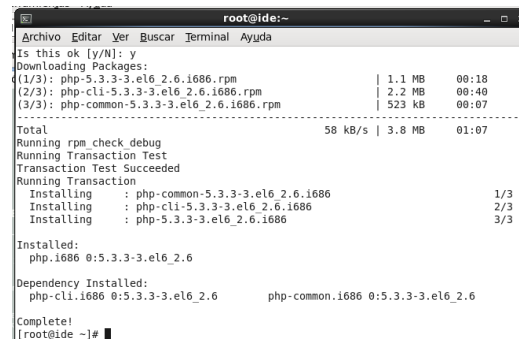
Abra un terminal y ejecute el siguiente comando **yum install php** como se muestra en la figura V.142.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# yum install php  
Loaded plugins: fastestmirror, refresh-packagekit  
Loading mirror speeds from cached hostfile  
* base: mirror.sanctuaryhost.com  
* extras: mirror.sanctuaryhost.com  
* updates: mirror.ueg.br  
Setting up Install Process  
Resolving Dependencies
```

Figura V. 142. Instalación de php
Fuente: Los Autores

Ingrese **y** para que proceda la descarga e instalación del paquete, como se muestra en la figura V.143.

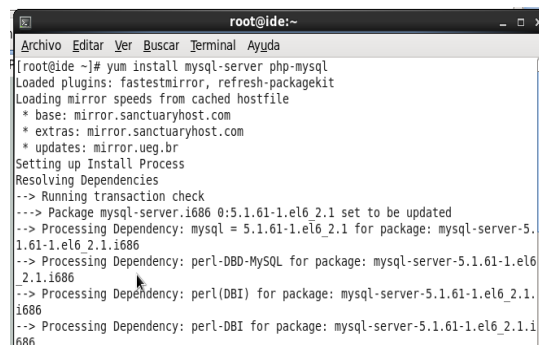


```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
Is this ok [y/N]: y  
Downloading Packages:  
(1/3): php-5.3.3-3.el6_2.6.i686.rpm | 1.1 MB 00:18  
(2/3): php-cli-5.3.3-3.el6_2.6.i686.rpm | 2.2 MB 00:40  
(3/3): php-common-5.3.3-3.el6_2.6.i686.rpm | 523 kB 00:07  
-----  
Total | 58 kB/s | 3.0 MB 01:07  
Running rpm check debug  
Running Transaction Test  
Transaction Test Succeeded  
Running Transaction  
Installing : php-common-5.3.3-3.el6_2.6.i686 1/3  
Installing : php-cli-5.3.3-3.el6_2.6.i686 2/3  
Installing : php-5.3.3-3.el6_2.6.i686 3/3  
Installed:  
php.i686 0:5.3.3-3.el6_2.6  
Dependency Installed:  
php-cli.i686 0:5.3.3-3.el6_2.6 php-common.i686 0:5.3.3-3.el6_2.6  
Complete!  
[root@ide ~]#
```

Figura V. 143. Descarga e instalación de php
Fuente: Los Autores

Instalación de Mysql

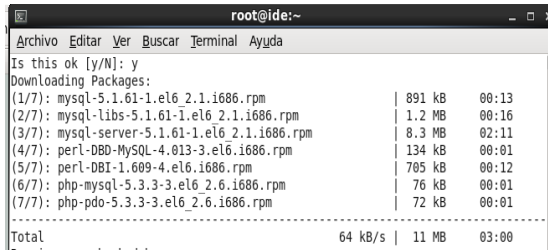
Abrir un terminal y ejecutar el siguiente comando **yum install mysql-server php-mysql**, como se muestra en la figura V.144.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# yum install mysql-server php-mysql  
Loaded plugins: fastestmirror, refresh-packagekit  
Loading mirror speeds from cached hostfile  
* base: mirror.sanctuaryhost.com  
* extras: mirror.sanctuaryhost.com  
* updates: mirror.ueg.br  
Setting up Install Process  
Resolving Dependencies  
--> Running transaction check  
--> Package mysql-server.i686 0:5.1.61-1.el6_2.1 set to be updated  
--> Processing Dependency: mysql = 5.1.61-1.el6_2.1 for package: mysql-server-5.1.61-1.el6_2.1.i686  
--> Processing Dependency: perl-DBD-MySQL for package: mysql-server-5.1.61-1.el6_2.1.i686  
--> Processing Dependency: perl(DBI) for package: mysql-server-5.1.61-1.el6_2.1.i686  
--> Processing Dependency: perl-DBI for package: mysql-server-5.1.61-1.el6_2.1.i686
```

Figura V. 144. Instalación de mysql
Fuente: Los Autores

Ingrese y para que proceda la descarga e instalación del paquete, como se muestra en la figura V.145.

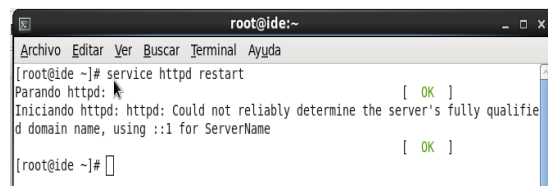


```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
Is this ok [y/N]: y  
Downloading Packages:  
(1/7): mysql-5.1.61-1.el6_2.1.i686.rpm | 891 kB 00:13  
(2/7): mysql-libs-5.1.61-1.el6_2.1.i686.rpm | 1.2 MB 00:16  
(3/7): mysql-server-5.1.61-1.el6_2.1.i686.rpm | 8.3 MB 02:11  
(4/7): perl-DBD-MYSQL-4.013-3.el6.i686.rpm | 134 kB 00:01  
(5/7): perl-DBI-1.609-4.el6.i686.rpm | 705 kB 00:12  
(6/7): php-mysql-5.3.3-3.el6_2.6.i686.rpm | 76 kB 00:01  
(7/7): php-pdo-5.3.3-3.el6_2.6.i686.rpm | 72 kB 00:01  
-----  
Total | 64 kB/s | 11 MB 03:00
```

Figura V. 145. Descarga e instalación de mysql

Fuente: Los Autores

Reiniciar el servidor Apache con el siguiente comando **service httpd restart**, como se muestra en la figura V.146.

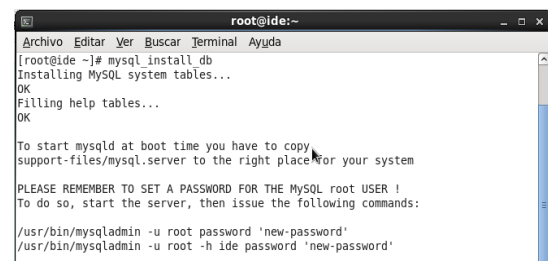


```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# service httpd restart  
Parando httpd: [ OK ]  
Iniciando httpd: httpd: Could not reliably determine the server's fully qualified domain name, using ::1 for ServerName [ OK ]  
[root@ide ~]#
```

Figura V. 146. Reiniciando apache

Fuente: Los Autores

Ejecutar el siguiente comando **mysql_install db** para instalar la base de datos, como se muestra en la figura V.147.

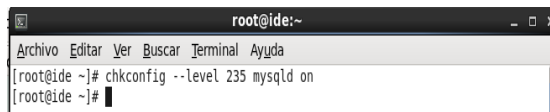


```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# mysql_install db  
Installing MySQL system tables...  
OK  
Filling help tables...  
OK  
To start mysqld at boot time you have to copy support-files/mysql.server to the right place for your system  
PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !  
To do so, start the server, then issue the following commands:  
  
/usr/bin/mysqladmin -u root password 'new-password'  
/usr/bin/mysqladmin -u root -h ide password 'new-password'
```

Figura V. 147. Instalacion de la db

Fuente: Los Autores

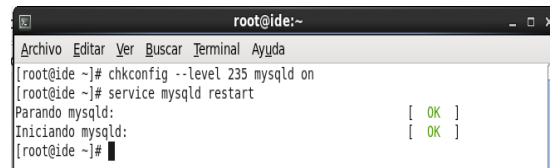
Ejecute el siguiente comando **chkconfig --levels 235 mysqld on**, como se muestra en la figura V.148 para que el servicio se levante automáticamente al momento de reiniciar la máquina.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# chkconfig --level 235 mysqld on  
[root@ide ~]#
```

Figura V. 148. Levantamiento de servicio
Fuente: Los Autores

Reiniciar el servidor mysql con el siguiente comando **service mysqld restart**, como se muestra en la figura V.149.

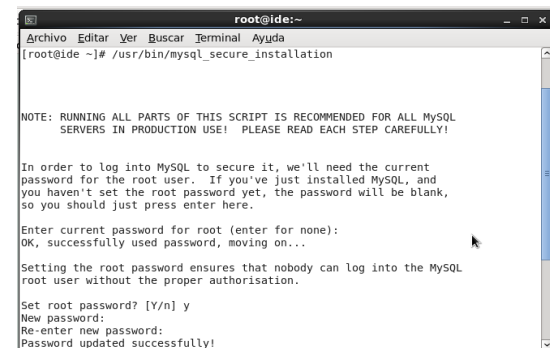


```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# chkconfig --level 235 mysqld on  
[root@ide ~]# service mysqld restart  
Parando mysqld: [ OK ]  
Iniciando mysqld: [ OK ]  
[root@ide ~]#
```

Figura V. 149. Reiniciando mysql
Fuente: Los Autores

Configuración de mysql

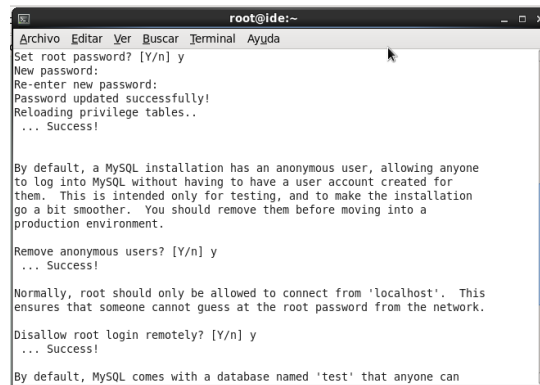
Ejecutar el siguiente comando **/usr/bin/mysql_secure_installation**, como se muestra en la figura V.150 para configurar la base de datos mysql.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# /usr/bin/mysql_secure_installation  
  
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!  
  
In order to log into MySQL to secure it, we'll need the current  
password for the root user. If you've just installed MySQL, and  
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.  
  
Enter current password for root (enter for none):  
OK, successfully used password, moving on...  
  
Setting the root password ensures that nobody can log into the MySQL  
root user without the proper authorisation.  
  
Set root password? [Y/n] y  
New password:  
Re-enter new password:  
Password updated successfully!
```

Figura V. 150. Configuración de mysql
Fuente: Los Autores

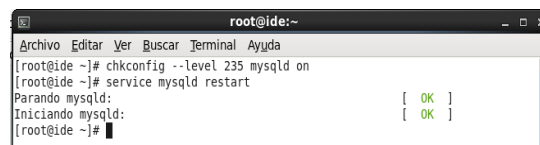
Ingresar el **password** para acceder a la base de datos, como se muestra en la figura V.151.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
Set root password? [Y/n] y  
New password:  
Re-enter new password:  
Password updated successfully!  
Reloading privilege tables..  
... Success!  
  
By default, a MySQL installation has an anonymous user, allowing anyone  
to log into MySQL without having to have a user account created for  
them. This is intended only for testing, and to make the installation  
go a bit smoother. You should remove them before moving into a  
production environment.  
  
Remove anonymous users? [Y/n] y  
... Success!  
  
Normally, root should only be allowed to connect from 'localhost'. This  
ensures that someone cannot guess at the root password from the network.  
  
Disallow root login remotely? [Y/n] y  
... Success!  
  
By default, MySQL comes with a database named 'test' that anyone can
```

Figura V. 151. Ingreso de password para la db
Fuente: Los Autores

Reinicie el servidor mysql con el siguiente comando **service mysqld restart**, como se muestra en la figura V.152.



```
root@ide:~  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ide ~]# chkconfig --level 235 mysqld on  
[root@ide ~]# service mysqld restart  
Parando mysqld: [ OK ]  
Iniciando mysqld: [ OK ]  
[root@ide ~]# █
```

Figura V. 152. Reiniciando mysqld
Fuente: Los Autores

Instalación de Joomla

Ubicar el paquete Joomla en el directorio de publicación y luego darle permiso mediante **chmod 777**, como se muestra en la figura V.153.



```
root@ideepoch:/var/www/html  
Archivo Editar Ver Buscar Terminal Ayuda  
[root@ideepoch html]# chmod 777 Joomla_1.7.5-Spanish-Pack Completo
```

Figura V. 153. Ejecución de joomla
Fuente: Los Autores

Seleccione el idioma y dar en siguiente, como se muestra en la figura V.154.

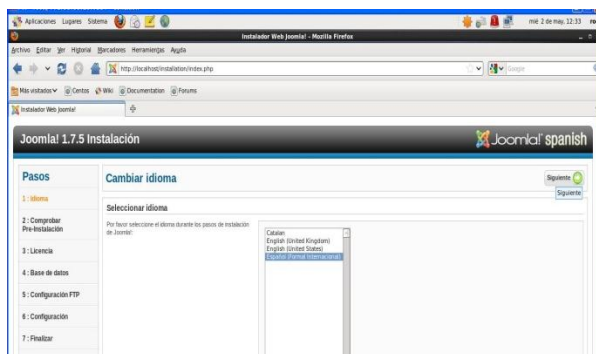


Figura V. 154. Selección de idioma
Fuente: Los Autores

Dar siguiente si está de acuerdo con los términos de licencia, como se muestra en la figura V.155.



Figura V. 155. Contrato de licencia
Fuente: Los Autores

Configuración de conexión con la base de datos MySQL, como se muestra en la figura V.156.

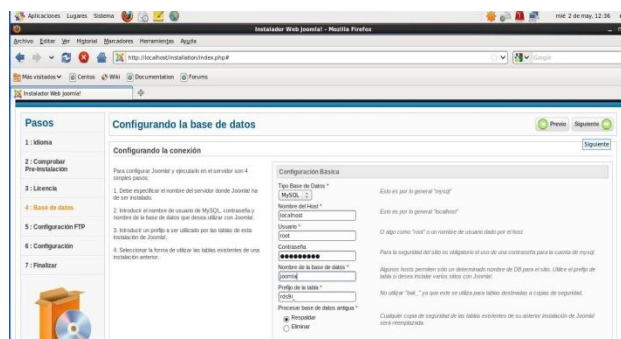


Figura V. 156. Configuración de base de datos
Fuente: Los Autores

Rellene los parámetros de configuración del sitio web, como se muestra en la figura V.157.

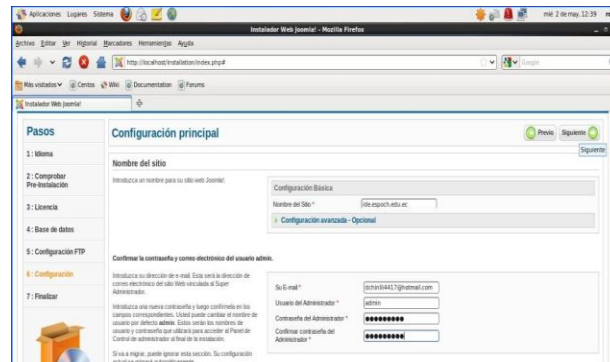


Figura V. 157. Configuración del sitio web
Fuente: Los Autores

Instalación finalizada, no olvide borrar el directorio de instalación, como se muestra en la figura V.158.

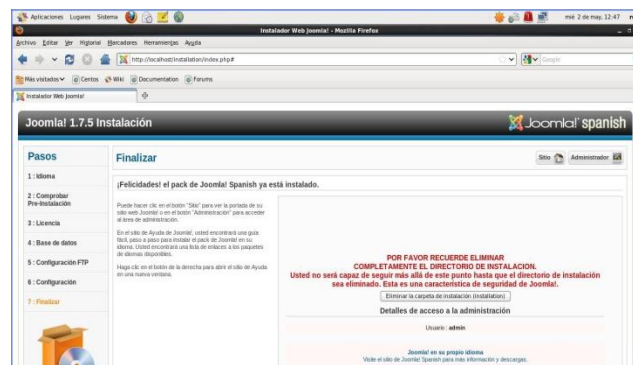


Figura V. 158. Instalación finalizada
Fuente: Los Autores

Ingresar al sitio administrativo de Joomla para desarrollar el geoportal, mediante su usuario y contraseña, como se muestra en la figura V.159.

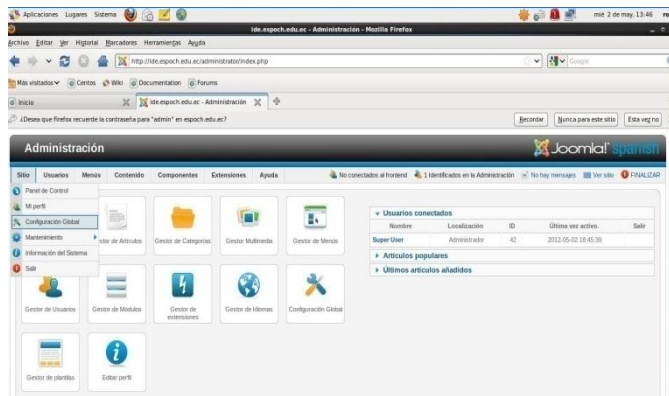


Figura V. 159. Backend de Joomla!
Fuente: Los Autores

Instalación jdk

Este paquete es requerido previamente antes de la instalación de tomcat, ya que va a permitir que la aplicación hecha en java sea entendida por el servidor.

De permiso al instalador con **chmod 777** y luego ejecutar, como se muestra en la figura V.160.

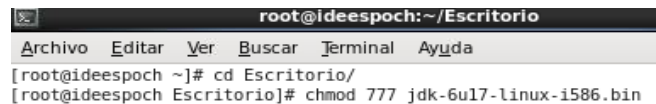


Figura V. 160. Permisos al instalador
Fuente: Los Autores

Presione **yes** si está de acuerdo con los términos de licencia, como se muestra en la figura V.161.

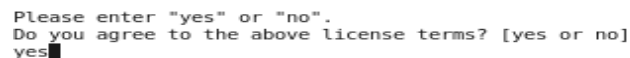


Figura V. 161. Aceptación de licencia
Fuente: Los Autores

Así transcurrirá el proceso de instalación que culminará cuando se muestre esta página que indica la Finalización del mismo, como se muestra en la figura V.162.



Figura V. 162. Comprobación de instalación correcta
Fuente: Los Autores

Instalación de Apache Tomcat

Para alojar el visor Cuenca y para su funcionamiento es necesario utilizar un servidor web en este caso se escogió Apache Tomcat .

Instalacion de tomcat

Copie el instalador en la siguiente direccion **/usr/local/**, abrir un terminal y ejecutar el siguiente comando **gedit /etc/profile**, como se muestra en la figura V.163.

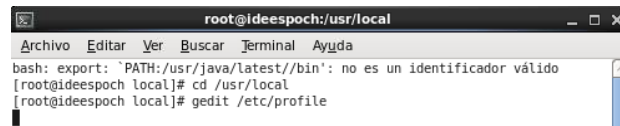


Figura V. 163. Ingreso al archivo profile
Fuente: Los Autores

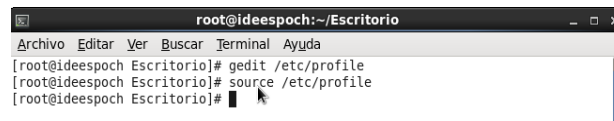
Ingrese lo siguiente en el archivo **profile**, como se muestra en la figura V.164.

```
export JAVA_HOME="/usr/latest"  
export PATH=$PATH:$JAVA_HOME/bin  
CATALINA_HOME="/usr/java/latest"  
Export CATALINA HOME
```

```
unset i  
unset pathmunge  
export JAVA_HOME="/usr/java/latest/"  
export PATH=$PATH:$JAVA_HOME/bin  
  
CATALINA_HOME="/usr/local/apache-tomcat-6.0.35"  
export CATALINA_HOME
```

Figura V. 164. Configuración del archivo profile
Fuente: Los Autores

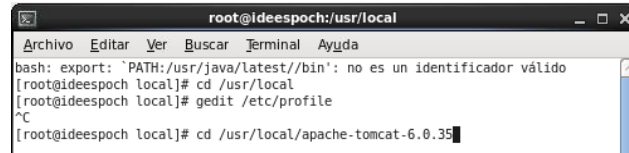
Actualice con el siguiente comando **source /etc/profile**, como se muestra en la figura V.165.



```
root@ideepoch:~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideepoch Escritorio]# gedit /etc/profile
[root@ideepoch Escritorio]# source /etc/profile
[root@ideepoch Escritorio]#
```

Figura V. 165. Actualización archivo profile
Fuente: Los Autores

Para ejecutar el servidor ingrese a la siguiente dirección **cd /usr/local/apache-tomcat - 6.0.35**, como se muestra en la figura V.166.



```
root@ideepoch:/usr/local
Archivo Editar Ver Buscar Terminal Ayuda
bash: export: `PATH:/usr/java/latest/bin': no es un identificador válido
[root@ideepoch local]# cd /usr/local
[root@ideepoch local]# gedit /etc/profile
^C
[root@ideepoch local]# cd /usr/local/apache-tomcat-6.0.35
```

Figura V. 166. Ingreso de ruta para ejecutar servidor
Fuente: Los Autores

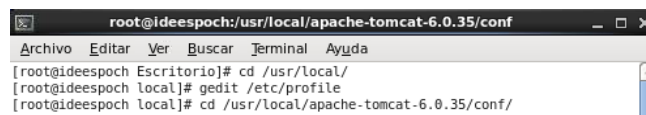
Ejecute el servidor **./catalina.sh run**, como se muestra en la figura V.167.



```
root@ideepoch:/usr/local/apache-tomcat-6.0.35
Archivo Editar Ver Buscar Terminal Ayuda
bash: export: `PATH:/usr/java/latest/bin': no es un identificador válido
[root@ideepoch local]# cd /usr/local
[root@ideepoch local]# gedit /etc/profile
^C
[root@ideepoch local]# cd /usr/local/apache-tomcat-6.0.35
[root@ideepoch apache-tomcat-6.0.35]# ./catalina.sh run
```

Figura V. 167. Ejecución del servidor
Fuente: Los Autores

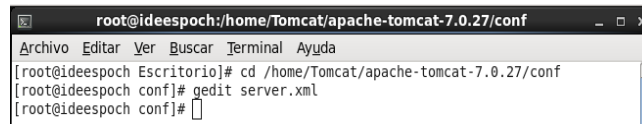
Para configurar el puerto del servidor ingrese a la siguiente dirección **cd /usr/local/apache-tomcat-6.0.35/conf/**, como se muestra en la figura V.168.



```
root@ideepoch:/usr/local/apache-tomcat-6.0.35/conf
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideepoch Escritorio]# cd /usr/local/
[root@ideepoch local]# gedit /etc/profile
[root@ideepoch local]# cd /usr/local/apache-tomcat-6.0.35/conf/
```

Figura V. 168. Ingreso de ruta para configurar el puerto
Fuente: Los Autores

Ingrese el siguiente comando **gedit server.xml**, como se muestra en la figura V.169.



```
root@ideespoch:/home/Tomcat/apache-tomcat-7.0.27/conf
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideespoch Escritorio]# cd /home/Tomcat/apache-tomcat-7.0.27/conf
[root@ideespoch conf]# gedit server.xml
[root@ideespoch conf]#
```

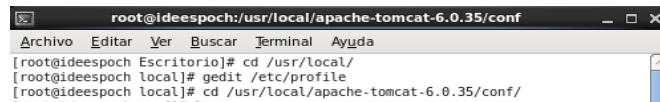
Figura V. 169. Configurar puerto
Fuente: Los Autores

Cambie el puerto que viene por defecto 80 por **8085**, como se muestra en la figura V.170.

```
<Connector port="8085" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443" />
```

Figura V. 170. Cambio de puerto
Fuente: Los Autores

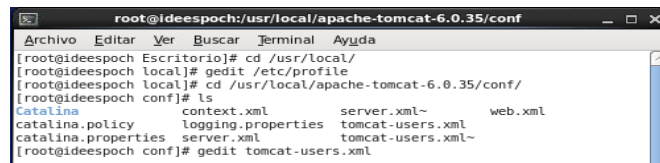
Para configurar usuario y contraseña ingrese a la siguiente dirección **cd /usr/local/apache-tomcat-6-0.35/conf/**, como se muestra en la figura V.171.



```
root@ideespoch:/usr/local/apache-tomcat-6.0.35/conf
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideespoch Escritorio]# cd /usr/local/
[root@ideespoch local]# gedit /etc/profile
[root@ideespoch local]# cd /usr/local/apache-tomcat-6.0.35/conf/
[root@ideespoch conf]#
```

Figura V. 171. Ingreso de ruta para configurar usuario y contraseña
Fuente: Los Autores

Ingrese el siguiente comando **gedit tomcat-users.xml**, como se muestra en la figura V.172.



```
root@ideespoch:/usr/local/apache-tomcat-6.0.35/conf
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideespoch Escritorio]# cd /usr/local/
[root@ideespoch local]# gedit /etc/profile
[root@ideespoch local]# cd /usr/local/apache-tomcat-6.0.35/conf/
[root@ideespoch conf]# ls
Catalina      context.xml      server.xml~      web.xml
catalina.policy  logging.properties  tomcat-users.xml
catalina.properties  server.xml      tomcat-users.xml~
[root@ideespoch conf]# gedit tomcat-users.xml
```

Figura V. 172. Configuración de credenciales
Fuente: Los Autores

Ingrese el usuario y password, como se muestra en la figura V.173.

```
<!--
<role rolename="tomcat"/>
<role rolename="role1"/>
<user username="tomcat" password="tomcat" roles="tomcat"/>
<user username="both" password="tomcat" roles="tomcat,role1"/>
<user username="role1" password="tomcat" roles="role1"/>
-->

<user username="admin" password="ideespoch" roles="manager-gui,admin-gui"/>
</tomcat-users>
```

Figura V. 173. Ingreso de credenciales
Fuente: Los Autores

Ingrese sus credenciales para ingresar a la administración del servidor tomcat, como se muestra en la figura V.174.

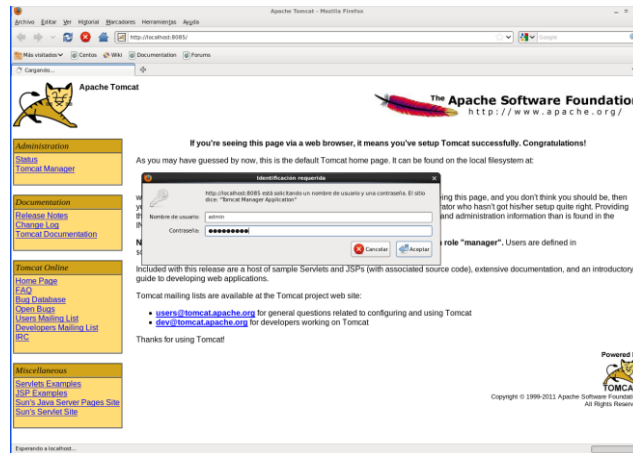


Figura V. 174. Autenticación
Fuente: Los Autores

Tomcat como servicio

Ingrese el siguiente comando **cd /etc/init.d** para configurar al servidor tomcat como servicio.

Crear un script en **/etc/init.d** para automatizar tomcat, como se muestra en la figura V.175.

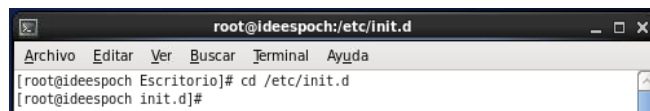


Figura V. 175. Ingreso al directorio para crear un script
Fuente: Los Autores

Ingrese el comando **gedit tomcat** para editar el script **init.d**, como se muestra en la figura V.176.

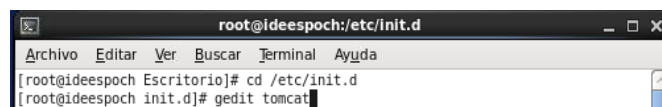
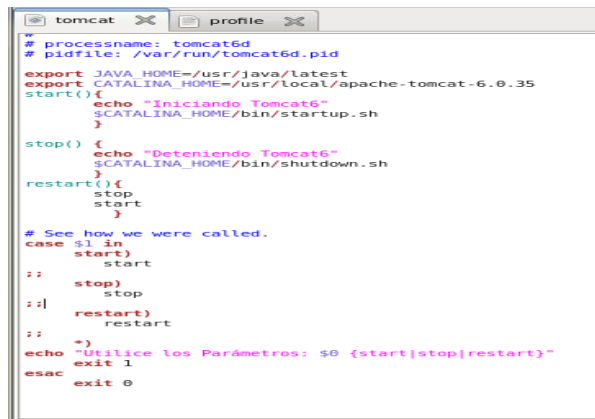


Figura V. 176. Comando para editar el archivo tomcat
Fuente: Los Autores

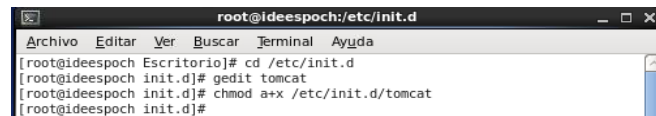
Ingrese el siguiente código en el archivo tomcat, como se muestra en la figura V.177.



```
tomcat  X  profile  X
# processname: tomcat6d
# pidfile: /var/run/tomcat6d.pid
export JAVA_HOME=/usr/java/latest
export CATALINA_HOME=/usr/local/apache-tomcat-6.0.35
start(){
    echo "Iniciando Tomcat6"
    SCATALINA_HOME/bin/startup.sh
}
stop(){
    echo "Deteniendo Tomcat6"
    SCATALINA_HOME/bin/shutdown.sh
}
restart(){
    stop
    start
}
# See how we were called.
case $1 in
    start)
        start
    ;;
    stop)
        stop
    ;;
    restart)
        restart
    ;;
    *)
        echo "Utilice los Parámetros: $0 {start|stop|restart}"
        exit 1
esac
exit 0
```

Figura V. 177. Ingreso de código
Fuente: Los Autores

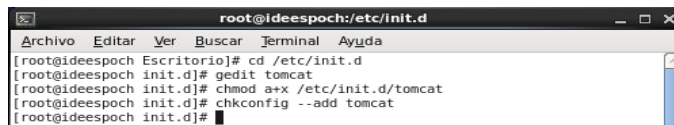
Cambiar los permisos con el comando **chmod a+x /etc/init.d/tomcat**, para que se pueda ejecutar código, como se muestra en la figura V.178.



```
root@ideepoch:/etc/init.d
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[root@ideepoch Escritorio]# cd /etc/init.d
[root@ideepoch init.d]# gedit tomcat
[root@ideepoch init.d]# chmod a+x /etc/init.d/tomcat
[root@ideepoch init.d]#
```

Figura V. 178. Permisos para ejecutar el código
Fuente: Los Autores

Agregar este script a servicios del sistema con el comando **chkconfig --add tomcat**, como se muestra en la figura V.179.

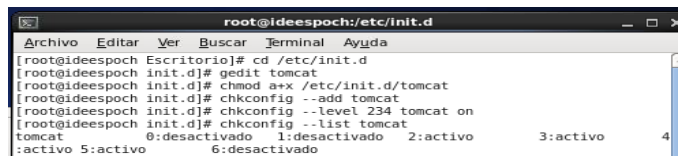


```
root@ideepoch:/etc/init.d
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[root@ideepoch Escritorio]# cd /etc/init.d
[root@ideepoch init.d]# gedit tomcat
[root@ideepoch init.d]# chmod a+x /etc/init.d/tomcat
[root@ideepoch init.d]# chkconfig --add tomcat
[root@ideepoch init.d]# █
```

Figura V. 179. Agregar script a servicios de sistema
Fuente: Los Autores

Verificar modificaciones (este script utiliza Niveles 2, 3 y 4), como se muestra en la figura V.180.

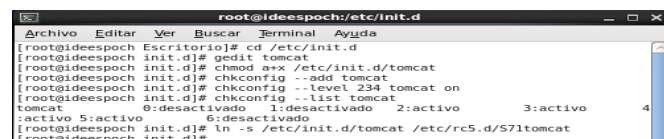
chkconfig --level 234 tomcat on
chkconfig --list tomcat



```
root@ideespoch:/etc/init.d
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideespoch Escritorio]# cd /etc/init.d
[root@ideespoch init.d]# gedit tomcat
[root@ideespoch init.d]# chmod a+x /etc/init.d/tomcat
[root@ideespoch init.d]# chkconfig --add tomcat
[root@ideespoch init.d]# chkconfig --level 234 tomcat on
[root@ideespoch init.d]# chkconfig --list tomcat
tomcat
0:desactivado 1:desactivado 2:activo 3:activo 4:
:activo 5:activo 6:desactivado
```

Figura V. 180. Verificar modificaciones
Fuente: Los Autores

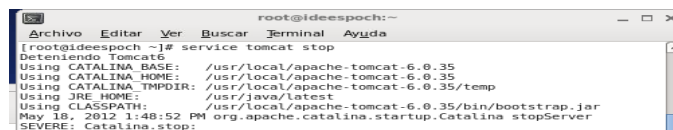
Agregue este script como un link simbolico para que se inicie automaticamente con el comando `ln -s/etc/init.d/tomcat/etc/rc5.d/S71tomcat`, como se muestra en la figura V.181.



```
root@ideespoch:/etc/init.d
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideespoch Escritorio]# cd /etc/init.d
[root@ideespoch init.d]# gedit tomcat
[root@ideespoch init.d]# chmod a+x /etc/init.d/tomcat
[root@ideespoch init.d]# chkconfig --add tomcat
[root@ideespoch init.d]# chkconfig --level 234 tomcat on
[root@ideespoch init.d]# chkconfig --list tomcat
tomcat
0:desactivado 1:desactivado 2:activo 3:activo 4:
:activo 5:activo 6:desactivado
[root@ideespoch init.d]# ln -s /etc/init.d/tomcat /etc/rc5.d/S71tomcat
```

Figura V. 181. Link simbolico
Fuente: Los Autores

Probar el script con `service tomcat stop`, como se muestra en la figura V.182.



```
root@ideespoch:~
Archivo Editar Ver Buscar Terminal Ayuda
[root@ideespoch ~]# service tomcat stop
Deteniendo Tomcat6
Using CATALINA_BASE:   /usr/local/apache-tomcat-6.0.35
Using CATALINA_HOME:   /usr/local/apache-tomcat-6.0.35
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-6.0.35/temp
Using JRE_HOME:         /usr/java/latest
Using CLASSPATH:        /usr/local/apache-tomcat-6.0.35/bin/bootstrap.jar
May 18, 2012 1:48:52 PM org.apache.catalina.startup.Catalina stopServer
SEVERE: Catalina.stop:
```

Figura V. 182. Parar servicio tomcat
Fuente: Los Autores

Probar el script con `service tomcat start`, como se muestra en la figura V.183.



```
root@ideespoch:~
Archivo Editar Ver Buscar Terminal Ayuda
:182)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:391)
at java.net.Socket.connect(Socket.java:579)
at java.net.Socket.connect(Socket.java:528)
at java.net.Socket.<init>(Socket.java:425)
at java.net.Socket.<init>(Socket.java:288)
at org.apache.catalina.startup.Catalina.stopServer(Catalina.java:422)
at sun.reflect.NativeMethodAccessorImpl.invoke(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.
java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAcces
sorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:601)
at org.apache.catalina.startup.Bootstrap.stopServer(Bootstrap.java:338)
at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:416)

[root@ideespoch ~]# service tomcat start
Iniciando Tomcat6
Using CATALINA_BASE:   /usr/local/apache-tomcat-6.0.35
Using CATALINA_HOME:   /usr/local/apache-tomcat-6.0.35
Using CATALINA_TMPDIR: /usr/local/apache-tomcat-6.0.35/temp
Using JRE_HOME:         /usr/java/latest
Using CLASSPATH:        /usr/local/apache-tomcat-6.0.35/bin/bootstrap.jar
[root@ideespoch ~]# █
```

Figura V. 183. Iniciar servicio tomcat
Fuente: Los Autores

Agregar el puerto 8085 al cortafuego del servidor, como se muestra en la figura V.184.



Figura V. 184. Agregar puerto 8085
Fuente: Los Autores

Instalación de Visor Cuenca

El visor Cuenca consta de 2 partes, la primera que contiene el visualizador (ideucuenca) y la segunda que consta del editor (aideucuenca) de capas.

La primera parte ideucuenca se debe situar dentro del directorio **usr/local/apache-tomcat/webapps**, ya que es el directorio de publicación en donde va a estar alojada la aplicación, como se muestra en la figura V.185.

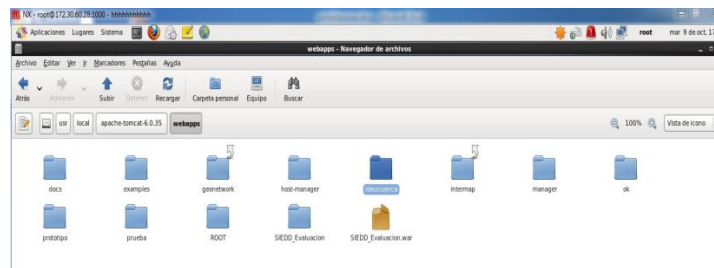


Figura V. 185. Directorio de publicación Visor de Cuenca
Fuente: Los Autores

Configurar el archivo `servers_google.js`, que esta ubicado dentro de `usr/local/apache_tomcaat/ideucuenca/servers`, como se muestra en la figura V.186.



Figura V. 186. Ubicación `servers_google.js`
Fuente: Los Autores

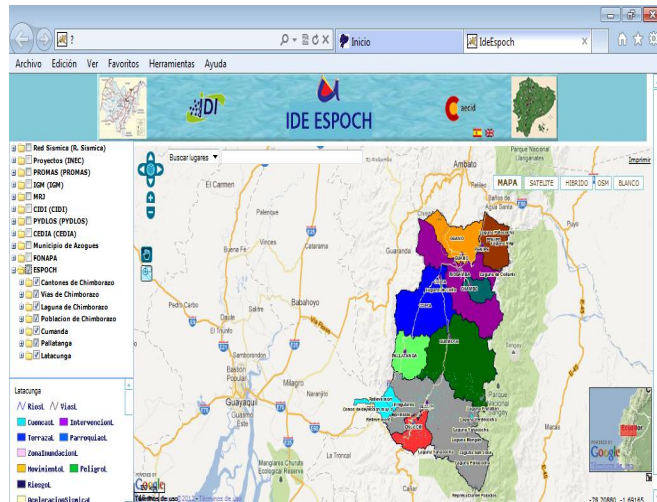


Figura V. 191. Capas de la provincia de Chimborazo
Fuente: Los Autores

Editor de mapas

La segunda parte aideucuenca contiene los archivos para la edicion de capas, este debe ser ubicado dentro del siguiente directorio usr/local, como se muestra en la figura V.192.

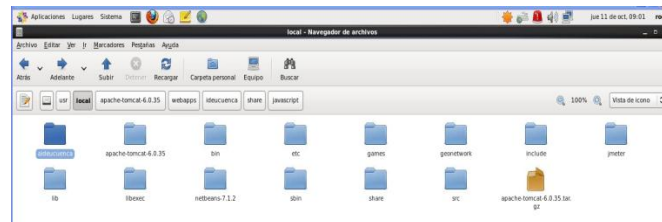


Figura V. 192. Path de editor de mapas (aideucuenca)
Fuente: Los Autores

En la siguiente figura se muestra el archivo usuarios.dbf que contiene los datos de los usuarios que estan autorizados para la gestion de capas, este se encuentra localizado en usr/local/aideucuenca.

	B	C	D	E	F	G	H
	USER	PSSW	CODE	ROLE	MAP	TITLE	
1	ADMIN	X12345	A0001	ADMIND	A0001	aideucuenca	ideUcuenca
2	UNI	U12345	A0002	USER	A0002	Uni demo	
3	DEMO	D12345	A0003	USER	A0003	Demio 2	
4	PRUEBA	12345	A0004	ADMIND	A0004	Prueba	

Figura V. 193. Usuarios.dbf
Fuente: Los Autores

El archivo index.html e index.html1 permite validar si el usuario está o no autorizado para la gestión de capas, como se muestra en la figura V.194.

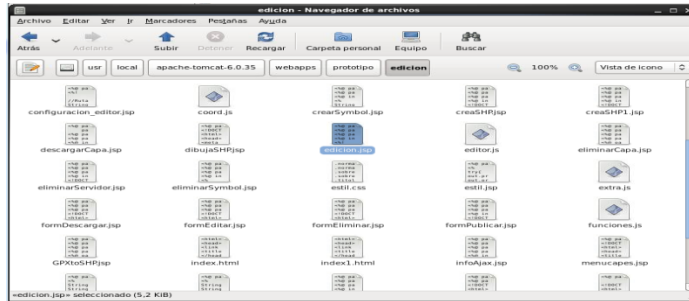


Figura V. 194. Path de index.html e index1.html
Fuente: Los Autores

Dirigase al siguiente path, en donde encontrara el archivo configuracion_editor.jsp, el cual permitira establecer las rutas de los directorios que intervendran en la aplicación, como se muestra en la figura V.195.



Figura V. 195. Path del archivo configuracion_editor.jsp
Fuente: Los Autores

Abrir el archivo configuracion_editor.jsp y configurar las rutas donde están almacenadas las librerías, css entre otros para que así guarde los cambios que se realicen correctamente, como se muestra en la figura V.196.

Ejecución del instalador, como se muestra en la figura V.200.

```
[root@ideespoch Escritorio]# java -jar geonetwork-install-2.4.1-0.jar
```

Figura V. 200. Ejecución de Instalador geonetwork
Fuente: Los Autores

Aceptar los términos de la licencia y dar clic en Next, como se muestra en la figura V.201.

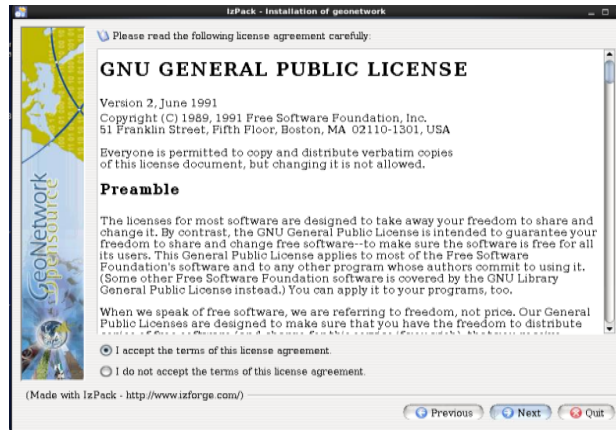


Figura V. 201. Aceptación de licencia
Fuente: Los Autores

Escribir la ruta del directorio donde va hacer instalado geonetwork, como se muestra en la figura V.202.

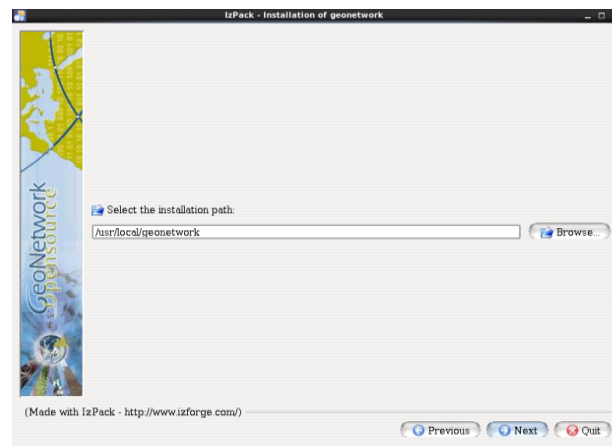


Figura V. 202. Ruta de geonetwork
Fuente: Los Autores

En la siguiente figura se indica el progreso de la instalación.

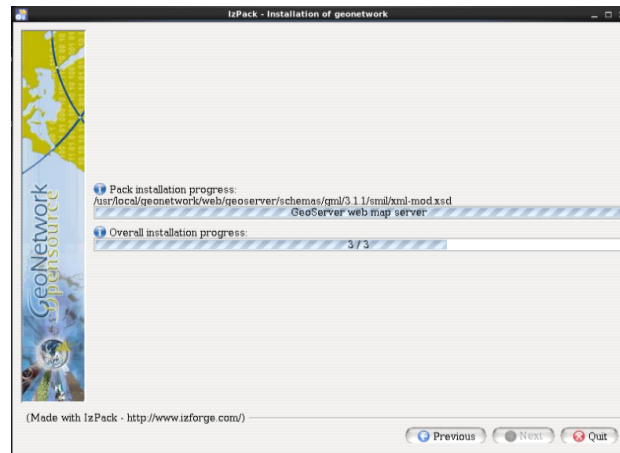


Figura V. 203. Progreso de Instalación
Fuente: Los Autores

La figura V.204 muestra que la Instalación ha finalizado.



Figura V. 204. Fin de Instalación
Fuente: Los Autores

Luego se le dice al servidor tomcat donde se encuentra la aplicación geonetwork instalada, para esto modificar el contexto del siguiente archivo server.xml, como se muestra en la figura V.205, que está situado en **usr/local/ apache-tomcat-6-0.35/conf**.

```
<!-- Define the default virtual host
Note: XML Schema validation will not work with Xerces 2.2.
-->
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true"
      xmlValidation="false" xmlNamespaceAware="false">

<Context path="/geonetwork" docBase="/usr/local/geonetwork/web/geonetwork" crossContext="false" debug="0" reloadable="false" />
<Context path="/intermap" docBase="/usr/local/geonetwork/web/intermap" crossContext="false" debug="0" reloadable="false" />
<Context path="/geoserver" docBase="/usr/local/geonetwork/web/geoserver" crossContext="false" debug="0" reloadable="false" />
```

Figura V. 205. Enlaces Simbólicos
Fuente: Los Autores

Comprobación de enlace geonetwork dentro del servidor tomcat, como se muestra en la figura V.206.

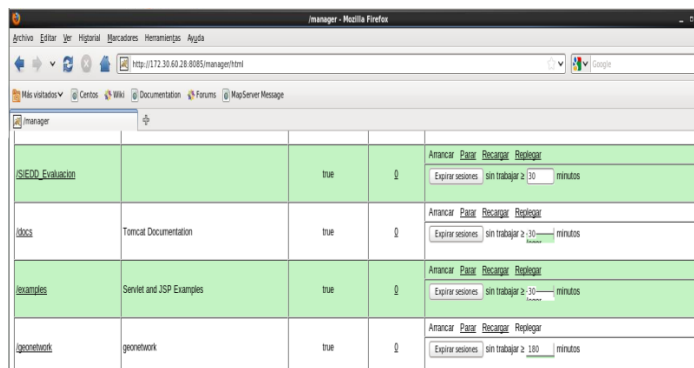


Figura V. 206. Comprobación geonetwork bajo Tomcat
Fuente: Los Autores

Ingresar la siguiente url <http://ide.epoch.edu.ec:8085/geonetwork/srv> para observar el servicio funcionando, como se muestra en la figura V.207.



Figura V. 207. Geonetwork instalado
Fuente: Los Autores

Metadatos creados acerca de las capas publicadas en el visor, como se muestra en la figura V.208.



Figura V. 208. Metadatos creados
Fuente: Los Autores

Geoportal funcionando, como se muestra en la figura V.209.



Figura V. 209. Geoportal
Fuente: Los Autores

5.1.4. Pruebas

Las pruebas se las realizan con el cliente, ya que es la persona que da el visto bueno al geoportal.

5.1.4.1. Pruebas de Aceptación

Las pruebas de aceptación colaboran en la comprobación de la funcionalidad de la aplicación, estas se realizan al final de cada iteración.

A continuación se presenta cada prueba en base a las historias de usuario establecidas.

Caso de prueba 1: Instalar herramientas de software.

Entrada

El usuario ingresa a la pantalla de inicio del geoportal.

Procesos

1. Ingrese la dirección <http://ide.esPOCH.edu.ec>.

Salidas

Las salidas van dirigidas a los usuarios sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.



Figura V. 210. Pantalla inicio Geoportal
Fuente: Los Autores

Los siguientes casos de prueba contendrá el siguiente proceso:

Caso de prueba 2: Georeferenciar los shapes.

Caso de prueba 3: Crear archivo mapfile

Caso de prueba 4: Crear leyenda que contendrá las capas de la provincia de Chimborazo.

Entrada

El geoportal permitirá visualizar información espacial en la WEB.

Procesos

1. Ingrese la dirección <http://ide.esPOCH.edu.ec>.
2. Clic en la opción Visor de Mapas del menú Recursos.

Salidas

Las salidas van dirigidas a los usuarios sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.

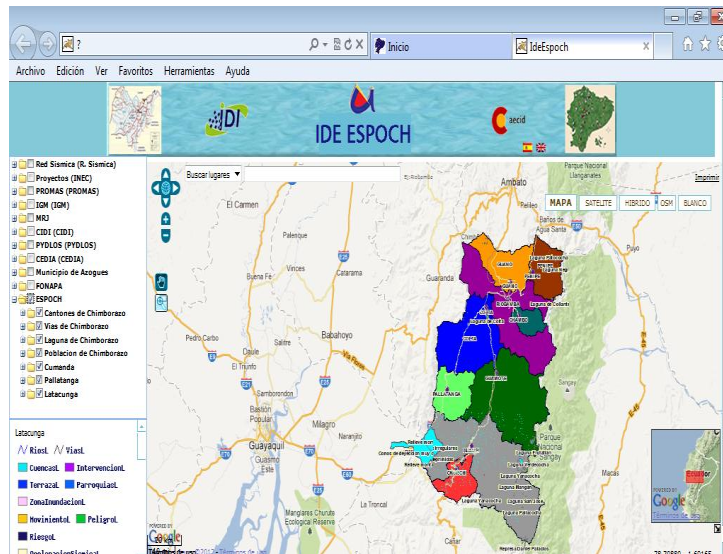


Figura V. 211. Difusión de Capas de Chimborazo

Fuente: Los Autores

Caso de prueba 5: Publicar información general de la IDE.

Entrada

El usuario ingresa a la pantalla de inicio del geoportal.

Procesos

2. Ingrese la dirección <http://ide.espoch.edu.ec>.
3. Menú Principal.

Salidas

Las salidas van dirigidas a los usuarios sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.



Figura V. 212. Publicación de información general
Fuente: Los Autores

Caso de prueba 6: Integrar el visor de mapas.

Entrada

El usuario ingresa a la pantalla de inicio del geoportail.

Procesos

1. Ingrese la dirección <http://ide.esPOCH.edu.ec>.
2. Clic en la opción Visor de Mapas del menú Recursos.

Salidas

Las salidas van dirigidas al usuario sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.



Figura V. 213. Integración visor de mapas
Fuente: Los Autores

Caso de prueba 7: Integrar el editor de mapas.

Entrada

El usuario ingresa a la pantalla de inicio del geoportal.

Procesos

1. Ingrese la dirección <http://ide.esPOCH.edu.ec>.
2. Clic en la opción editor de mapas del menú Recursos.
3. Ingresar usuario y contraseña.
4. Si los datos son correctos
 - 4.1. Permite el ingreso a la interfaz del editor.

Salidas

Las salidas van dirigidas al administrador sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.



Figura V. 214. Integración editor de mapas

Fuente: Los Autores

Caso de prueba 8: Integrar el catálogo de datos al geoportal.

Entrada

El usuario podrá consultar Metadatos.

Procesos

1. Ingrese la dirección <http://ide.esPOCH.edu.ec>.
2. Clic en la opción Catálogo de Datos del menú Recursos.
3. Presione el botón "Buscar".
4. Seleccione el botón "Pagina de Metadatos".

Salidas

Las salidas van dirigidas al usuario sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.



Figura V. 215. Catálogo de GeoNetwork
Fuente: Los Autores

Caso de prueba 9: Visualizar galería de fotos.

Entrada

El geoportal permitirá visualizar la galería de fotos.

Procesos

1. Ingrese la dirección <http://ide.espoche.edu.ec>.
2. Clic en la opción galería de fotos ubicada en la parte izquierda del geoportal.

Salidas

Las salidas van dirigidas al usuario sin ninguna restricción.

Evaluación de la prueba

Prueba satisfactoria.



Figura V. 216. Galería de fotos
Fuente: Los Autores

Tareas a realizar

Se procedió a implantar la aplicación en la Escuela Superior Politécnica de Chimborazo, la misma que será alojada en un servidor situado en DESITEL, permitiendo que el usuario tenga a disposición la aplicación ingresando a la página <http://ide.esPOCH.edu.ec>.

CONCLUSIONES

- La determinación de los parámetros fue en base a la investigación teórica de las herramientas de estudio y a la construcción del prototipo con cada visor permitiendo llevar a cabo el análisis estadístico.
- El establecer parámetros comprobables ayudó a la comparación entre CartoWeb, OpenLayers y el Visor web de la universidad de Cuenca, lo cual permitió determinar al visor web de la universidad de Cuenca con 88,73% como el visor más adecuado para la visualización de información geoespacial de la provincia de Chimborazo frente a CartoWeb que obtuvo el 64,90% y OpenLayers que consiguió el 44,86%.
- La metodología XP permitió el logro de los objetivos planteados para el desarrollo del geoportal a corto plazo ya que permite que la misma se inicie con algo simple y se ponga en producción rápidamente para luego realizar incorporaciones de funcionalidades.
- El haber levantado un geoportal no solo afecta a la comunidad universitaria, sino a todo el entorno, ya que su acceso es a través del internet facilitando así la consulta y el análisis de información referente a la provincia de Chimborazo cubriendo de esta manera las distintas necesidades del usuario, todo esto sin ningún costo.

RECOMENDACIONES

- Georeferenciar las capas .shp en las mismas coordenadas con que trabaja google maps, ya que si no se encuentra en la misma referencia las capas no podrán ser mostradas sobre la capa base de google.
- Es totalmente recomendable darle mantenimiento al geoportal prosiguiendo con la integración de cartografía sobre la provincia de Chimborazo con el fin de disponer de información actualizada.
- Se recomienda usar herramientas tecnológicas open source para que se reduzcan costos a la institución.
- Se debería integrar al visor web de la universidad de Cuenca más validaciones en la parte administrativa, ya que carece de seguridad al momento de iniciar y cerrar sesión.
- Se sugiere visitar las páginas en internet descritas en la sección de bibliografía o recurrir a medios bibliográficos como libros, etc, para el que quiera profundizar en el funcionamiento de este tipo de aplicaciones SIG.

RESUMEN

El presente trabajo de investigación se ha centrado en el análisis estadístico de los visores CartoWeb, OpenLayers y el visor web de la universidad de Cuenca, propiciando así la selección de la herramienta más adecuada para la difusión o visualización de la información geográfica de la provincia de Chimborazo en la implementación de una Infraestructura de Datos Espaciales para la ESPOCH con la pertinencia de satisfacer las necesidades del usuario.

Se aplicó el método analítico e investigativo para obtener los resultados de los parámetros de acuerdo a las pruebas realizadas en el prototipo hecho con cada uno de los visores, además de hacer uso de una técnica estadística para la comprobación de la hipótesis.

Las herramientas utilizadas para el análisis y desarrollo fueron las siguientes: el sistema operativo Centos 6.0, servidor Apache, servidor de mapas MapServer, ArcGis que ayudó a la georeferenciación de coordenadas y proyección de las capas.

De acuerdo a los resultados obtenidos en base al análisis estadístico realizado arrojó el siguiente resultado general: CartoWeb consiguió el 64,90%, OpenLayers obtuvo el 44,86%, y el visor web de la universidad de Cuenca alcanzó el 88,73%. Entonces se concluye que el visor web de la universidad de Cuenca, es el más adecuado para difundir información espacial de la provincia de Chimborazo en la Infraestructura de Datos Espaciales en la ESPOCH, ya que obtuvo el mayor puntaje.

Es totalmente recomendable hacer acuerdos institucionales para que se entregue información sobre la provincia de Chimborazo con el fin de darle mantenimiento al geoportal prosiguiendo así con la integración de cartografía en el visor permitiendo disponer de información actualizada.

SUMMARY

This researched work has been focused on the statistics analysis of the CartoWeb, OpenLayers and web Visor from the university of Cuenca viewers, favoring in this way the selection of the most useful tool for the diffusion or display of the geographical information of the Chimborazo Province in the implementation of an infrastructure of space data for the ESPOCH with the relevance of the user necessities.

The analytical and researched method was applied to obtain the results of the standards according to the test carried out in the prototype made with each viewer, besides to make use of a statistical technic for hypothesis testing.

The tools used for the analysis and development were the following ones: the operating system Centos 6.0, Apache server, MapServer, ArcGis that helped to the coordinated geo-referencing and projection of the layers.

According to the obtained results based on the statistical analysis carried out the following result: CartoWeb got 64,90%, OpenLayers got 44,86% and web Visor from the university of Cuenca reached 88,73%. Then we concluded that the web Visor from the university of Cuenca is the most suitable for disseminate spatial information of the Chimborazo Province in the infrastructure of spatial data in the ESPOCH, because it obtained the biggest score.

It is completely recommended to make institutional agreements to deliver information about Chimborazo Province in order to give maintenance to the geo-portal continuing in this way the Mapping Integration in the Viewer allowing to have up – date information.

GLOSARIO

API: Application Programming Interface. Un API no es más que una serie de servicios o funciones que el Sistema Operativo ofrece al programador.

ArcGIS: Es un software GIS para visualizar, crear, manipular y gestionar información geográfica.

Base de datos: Es una colección estructurada de datos y forma parte de un sistema de información.

Cartografía: Es el conjunto de estudios, operaciones científicas y técnicas que intervienen en la producción o análisis de mapas, modelos en relieve o globos que representan la Tierra, parte de ella o cualquier parte del Universo.

Cobertura: Es la forma de agrupar objetos de características similares en un espacio.

Google Maps: Es un servidor de aplicaciones de mapas en web gratuito de google. Ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo e incluso la ruta entre diferentes ubicaciones. Asimismo, ofrece la posibilidad de que cualquier propietario de una página web integre muchas de sus características a su sitio web.

Open source: Código abierto. Permite que el software sea estudiado y distribuido para que la comunidad se beneficie.

Píxel: Es una combinación de picture y element. Menor unidad posible con la que se compone cualquier imagen digital en una computadora.

Sistemas de coordenadas: los sistemas de coordenadas son las herramientas que permiten localizar un punto sobre la superficie terrestre.

Visor Cuenca: Visor Web de la Universidad de Cuenca

BIBLIOGRAFÍA

1. **BUZAI, G., Y DURÁN, D.** Enseñar e investigar con Sistemas de Información Geográfica (SIG). Edit. Troquel, Buenos Aires. 1997 191 p.
2. **MASSER, I., BORRERO, S., HOLLAND, P., Y OTROS.** Developing Spatial Data Infrastructures, From concept to reality, p. 59-77.

BIBLIOGRAFÍA DE INTERNET

3. APACHE TOMCAT

<http://tomcat.apache.org/download-60.cgi>

2012/07/15

4. CARTOWEB

<http://cartoweb.org/documentation.html>

2012/05/15

5. COMPONENTES DE UN IDE

<http://idet.tucuman.gob.ar>

2012/05/12

6. EJEMPLOS DE OPENLAYERS

<http://openlayers.org/dev/examples/>

2012/06/3

7. FGS

<http://dl.maptools.org/dl/fgs/releases/9.5/self-installers/>

2012/07/12

8. FUNDAMENTOS DE UN IDE

http://idesc.cali.gov.co/download/curso_fundamentos_ide/10.Manual_PDF

[_MapSever.pdf](#)

2012/05/13

9. GEOPORTAL

<http://dugi-doc.udg.edu/bitstream/handle/10256/3459/a5.pdf>

2012/07/12

10. GEONETWOK

http://sourceforge.net/projects/geonetwork/files/GeoNetwork_opensource/v2.6.4/geonetwork-install-2.6.4-0.jar/download

2012/07/18

11. IDE

http://www.ccidep.gob.pe/archiv/doc/DocDifusion_IDEP.pdf

2012/05/18

12. JOOMLA

<http://www.joomlaspanish.org/>

2012/07/18

13. LIBRERIAS DE OPENLAYERS

<http://trac.osgeo.org/openlayers/browser/trunk/openlayers/lib/OpenLayers/Layer/MapServer.js>

2012/06/5

14. LINUX

<http://download.filesystems.org/linux/centos/>

2012/07/14

15. MAPSERVER

<http://www.stack-es.com/stackoverflow/es/good-examples-of-mapserver-openlayers-2723290.html>

2012/05/12

16. OPENLAYERS

<http://openlayers.turbolinux.org/tag/mapserver>

2012/05/20

17. VENTAJAS DE OPENLAYERS

<http://comments.gmane.org/gmane.comp.gis.openlayers.user/15618>

2012/05/25

ANEXOS

ANEXO I

Archivo MapFile

Código del archivo mapFile

```
MAP
#Proyeccion en coordenadas geograficas
PROJECTION
    "init=epsg:4326"
END
STATUS ON
TRANSPARENT ON

SIZE 400 300

#Extension mapa [minx] [miny] [maxx] [maxy]

EXTENT -81.304375 -4.8882981455 -75.008507525 1.4075752

    #Symbolos
SYMBOLSET "/opt/fgs/www/htdocs/symbols/symbols.sym"
FONTSET '/opt/fgs/apps/gmap-demo-cvs_MS_VERSION_54/etc/fonts.txt'

#Formato de salida de la imagen
IMAGETYPE PNG

#Color de fondo
IMAGECOLOR 255 255 0

#Unidades del mapa
UNITS METERS

#Ruta libreria PROJ
CONFIG "PROJ_LIB" '/opt/fgs/share/proj/'

#Ruta de la cartografia
SHAPEPATH "data"

WEB
    IMAGEPATH "/opt/fgs/tmp/ms_tmp/"
    IMAGEURL "/ms_tmp/"
    METADATA
        "wms_title" "Ecuador"
        "wms_onlineresource" "http://localhost:8080/cgi-
bin/mapserv?map=/opt/fgs/www/htdocs/ideespoch.map&"
        "wms_srs" "EPSG:32717 EPSG:4326 EPSG:900913"

    END
END

OUTPUTFORMAT
    NAME "png"
    MIMETYPE "image/png"
    DRIVER "GD/PNG"
    EXTENSION "png"
    IMAGEMODE PC256
    TRANSPARENT FALSE
END
```

```
LEGEND
  KEYSIZE 18 12
  LABEL
    Type Bitmap
    Size MEDIUM
    COLOR 0 0 89
  END
  STATUS ON
END #LEGEND
```

```
OUTPUTFORMAT
  NAME "png"
  MIMETYPE "image/png"
  DRIVER "GD/PNG"
  EXTENSION "png"
  IMAGEMODE PC256
  TRANSPARENT TRUE
END
```

```
SYMBOL
  NAME 'square'
  TYPE VECTOR
  FILLED TRUE
  POINTS
    0 1
    0 0
    1 0
    1 1
    0 1
  END
END #SYMBOL
```

```
SYMBOL
  NAME "diagonal-auf"
  TYPE vector
  POINTS
    0 1
    1 0
  END
END
```

```
#INICIO LAYERS
```

```
LAYER
  NAME "Chimborazo"
  PROJECTION
    "init=epsg:4326"
  END
  METADATA
    "wms_title" "Chimborazo"
    "wms_srs" "EPSG:4326"
  END
  STATUS ON
  #TEMPLATE
  DUMP TRUE
  DATA Chimborazo
  UNITS METERS
```

```
SIZEUNITS PIXELS
TOLERANCEUNITS PIXELS
TYPE LINE
LABELMAXSCALE 202994
SYMBOLSCALE 25000
CLASS
    NAME "Chimborazo"
    COLOR 0 255 0
    STYLE
        ANTIALIAS TRUE
    COLOR 0 0 30
    SIZE 20
    MINSIZE 1
    MAXSIZE 100
    OUTLINECOLOR 110 110 110
END
END
END
```

```
LAYER
NAME "Riobamba"
PROJECTION
    "init=epsg:4326"
END
TYPE POLYGON
STATUS ON
DATA Riobamba
DUMP TRUE
TOLERANCE 9
SIZEUNITS PIXELS
TOLERANCEUNITS PIXELS
SYMBOLSCALE 25000
UNITS METERS
LABELITEM "CANTON"
CLASS
    NAME "Riobamba"
    COLOR 0 0 255
    STYLE
        SYMBOL 'solido'
        ANTIALIAS TRUE
        SIZE 10
        MINSIZE 10
        MAXSIZE 10
        COLOR 153 0 153
        OUTLINECOLOR 110 110 110
    END
    LABEL
        ANTIALIAS TRUE
        POSITION AUTO
        TYPE TRUETYPE
        FONT arial-bold
        MINSIZE 6
        MAXSIZE 10
        SIZE 6
        BUFFER 2
```

COLOR 255 0 0
OUTLINECOLOR 255 255 255
END

END
END

#FIN LAYERS OK

end of map file
END

ANEXO II

Encuesta

ENCUESTA

ESPOCH

FACULTAD DE INFORMATICA Y ELECTRONICA

ESCUELA DE INGENIERIA EN SISTEMAS

OBJETIVO: Determinar si los Visores permiten mostrar adecuadamente la información espacial en una infraestructura de datos espaciales.

CUESTIONARIO

1. ¿Cree usted que CartoWeb tiene una estructura organizada para interaccionar?

SI__ NO__

2. ¿Cree usted que el CartoWeb web consideras que tiene una interfaz comprensible en la parte textual?

SI__ NO__

3. ¿CartoWeb posee todas las herramientas de navegación Volver al inicio, Zoom, Paneo, Mapa de Referencia, Escala, Impresión y Coordenadas geográficas?

SI__ NO__

4. ¿Cree usted que OpenLayers tiene una estructura organizada para interaccionar?

SI__ NO__

5. ¿Cree usted que el OpenLayers consideras que tiene una interfaz comprensible en la parte textual?

SI__ NO__

6. ¿OpenLayers posee todas las herramientas de navegación Volver al inicio, Zoom, Paneo, Mapa de Referencia, Escala, Impresión y Coordenadas geográficas?

SI__ NO__

7. ¿Cree usted que el visor Cuenca tiene una estructura organizada para interaccionar?

SI__ NO__

8. ¿Cree usted que el Visor Cuenca web consideras que tiene una interfaz comprensible en la parte textual?

SI__ NO__

9. ¿Visor Cuenca posee todas las herramientas de navegación Volver al inicio, Zoom, Paneo, Mapa de Referencia, Escala, Impresión y Coordenadas geográficas?

SI__ NO__

Tabla a.0. Tabulación de las encuestas

# Encuesta	CARTOWEB			OPENLAYERS			VISOR CUENCA		
	Calidad estética	Comprensibilidad	Herramientas de Navegación	Calidad estética	Comprensibilidad	Herramientas de Navegación	Calidad estética	Comprensibilidad	Herramientas de Navegación
1	1	1	0	1	1	0	1	1	1
2	1	0	0	0	1	0	1	1	1
3	0	1	0	0	1	0	1	1	1
4	1	1	0	0	1	0	0	1	1
5	0	1	0	0	0	0	1	1	1
6	1	1	0	0	0	0	1	1	1
7	1	1	0	0	1	0	1	1	1
8	1	1	0	0	1	0	0	1	1
9	1	0	0	0	1	0	1	0	1
10	1	1	0	0	1	0	1	1	1
11	1	1	0	0	1	0	1	1	1
12	0	1	0	0	1	0	1	1	1
13	1	1	0	1	1	0	1	1	1
14	1	1	0	0	1	0	0	1	1
15	1	1	0	0	0	0	1	1	1
16	1	0	0	0	1	0	1	0	1
17	1	1	0	0	1	0	1	1	1
18	1	1	0	0	1	0	1	1	1
19	1	1	0	0	1	0	1	1	1
20	1	1	0	1	1	0	1	1	1
21	1	1	0	0	1	0	0	1	1
22	0	1	0	0	1	0	1	1	1
23	1	0	0	0	1	0	1	0	1
TOTAL	19	19	0	3	19	0	19	20	23

ANEXO III

Cálculos Estadísticos

Cálculo de la muestra

$$n = \frac{NZ^2pq}{(N - 1)E^2 + Z^2pq}$$

Dónde:

n =tamaño de la muestra

N =universo o Población

p =proporción esperada (en este caso 5%=0.05)

$q=1-p$ (en este caso 1-0.05=0.95)

Z =nivel de confianza deseado 1.96 (si la seguridad es 95%)

E =límite aceptable de error muestral en este caso tenemos un error de 8% que se encuentra en un rango de]0%,10% [

$$n = \frac{119 * (1.96)^2 * 0.05 * 0.95}{(119 - 1)(0.08)^2 + (1.96)^2 * 0.05 * 0.95}$$
$$n = \frac{21.71}{0.93}$$

$$n = 23.34$$

$$n = 23$$

Comprobación de la hipótesis

El proyecto permitirá mantener disponible la información geográfica para el usuario, además de brindar soporte al gobierno, ya que traerá consigo ser un apoyo para la toma de decisiones objetiva y para una sólida política de administración de tierras.

Hipótesis

Ho:”El análisis estadístico de visores web no permite establecer el más adecuado para mostrar información espacial en una infraestructura de datos espaciales”.

H1: "El análisis estadístico de visores web permite establecer el más adecuado para mostrar información espacial en una infraestructura de datos espaciales".

Nivel de significación

Una vez establecida la hipótesis nula y la alternativa, se debe determinar el nivel de significancia, que para el caso del presente análisis se utilizara un nivel de significación estadística de $\alpha=0.05$, ya que es probable que el visor ganador no presente una usabilidad adecuada para todos los usuarios.

Criterio

De acuerdo al análisis desarrollado en la presente investigación se ha seleccionado como estadístico de prueba de hipótesis la técnica "Chi Cuadrado". La fórmula que da el estadístico es la siguiente:

$$x^2 = \sum_i \frac{(\text{observada } i - \text{esperada } i)}{\text{esperada } i}$$

Para conocer las frecuencias teóricas o esperadas, se calculan a través del producto de los totales marginales (*total del renglón * total de columna*), dividido por el número total de casos (*gran total*).

$$fe = \frac{(\text{total del renglón}) * (\text{total de la columna})}{\text{gran total}}$$

En la tabla se pueden observar los resultados de los cálculos, tanto de la frecuencia esperada, como la del valor de " x^2 calculado", luego de haber aplicado las formulas anteriores.

Tabla a. Matriz (3r * 3k) sobre resultado de la encuesta

Visualización de datos	CARTOWEB	OPENLAYERS	VISOR CUENCA	TOTAL
Visor Web				
Calidad estética.	19	3	19	41
Comprensibilidad de texto o información.	19	19	20	58
Soporte de herramientas.	0	0	23	23
TOTAL	38	22	62	122

Elaborado por: Los Autores

Tabla b. Valores esperados

Visualización de datos	CARTOWEB	OPENLAYERS	VISOR CUENCA	TOTAL
Visor Web				
Calidad estética.	13	7	21	41
Comprensibilidad de texto o información.	18	11	29	58
Soporte de herramientas.	7	4	12	23
TOTAL	63	6	62	122

Elaborado por: Los Autores

Ahora, se diseña la tabla para aplicar la fórmula de chi-cuadrado

Tabla c. Frecuencia Observada/Frecuencia Esperada

	Fo	Fe	(fo-fe) ² /fe
CartoWeb tiene calidad estética	19	13	2.76
CartoWeb tiene Comprensibilidad de texto o información.	19	18	0.05
CartoWeb tiene herramientas de navegación.	0	7	7
OpenLayers tiene calidad estética	3	7	2.28
OpenLayers tiene Comprensibilidad de texto o información.	19	11	5.81
OpenLayers tiene herramientas de navegación.	0	4	4
Visor Cuenca tiene calidad estética	19	21	0.19
Visor Cuenca tiene Comprensibilidad de texto o información.	20	29	2.79
Visor Cuenca tiene herramientas de navegación.	23	12	10.08
TOTAL	122	122	34.96

Elaborado por: Los Autores

Ahora es necesario determinar el **criterio de decisión**. Entonces se acepta **H₀** cuando:

$x^2_{\text{calculado}} < x^2_{\text{tabla}}$, en caso contrario se rechaza **H₀**

Donde el valor de x^2_{tabla} representa el valor proporcionado por la tabla de “distribución x^2 ”, según el nivel de significación elegido y los grados de libertad.

Como se mencionó anteriormente, el nivel de significancia adoptado para esta investigación es de $\alpha = 0.05$.

Para la determinación de los grados de libertad (gl) se debe aplicar la siguiente fórmula:

$$gl = (r - 1) * (k - 1)$$

Donde r el numero de filas o renglones y k el numero de columnas. La investigación generó una matriz de $3r * 3k$.

Entonces:

$$gl = (r - 1) * (k - 1)$$

$$gl = (3 - 1) * (3 - 1)$$

$$gl = 4$$

De acuerdo a la tabla estadística de distribución de chi-cuadrado, con un nivel de significancia 0.05 a 2 grados de libertad, genera un valor de $x^2_{tabla} = 9.488$

Tabla

Tabla d. Distribución de x^2

Grado de Dificultad	90%	80%	70%	50%	30%	20%	10%	5%	1%
1	0.016	0.064	0.148	0.455	1.074	1.642	2.706	3.841	6.635
2	0.211	0.446	0.713	1.386	2.408	3.219	4.605	5.991	9.210
3	0.584	1.005	1.424	2.366	3.665	4.642	6.251	7.815	11.341
4	1.064	1.649	2.195	3.357	4.878	5.989	7.779	9.488	11.341
5	1.064	2.343	3.000	4.351	6.064	7.289	9.236	11.070	15.086

Elaborado por: Los Autores

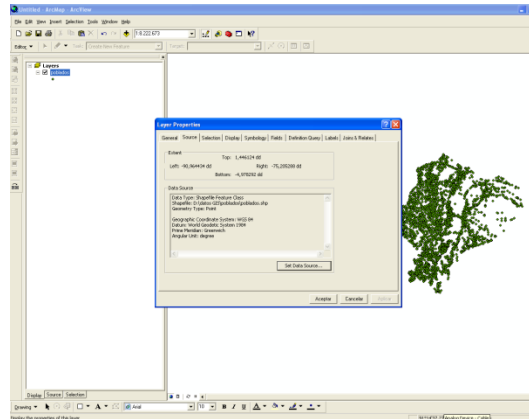
La regla de decisión es entonces: No rechazar H_0 si el valor que se encuentre para x^2 calculado es menor que 9.488. Si el valor calculado es igual o mayor al valor crítico, se rechaza H_0 y se acepta H_1 .

ANEXO IV

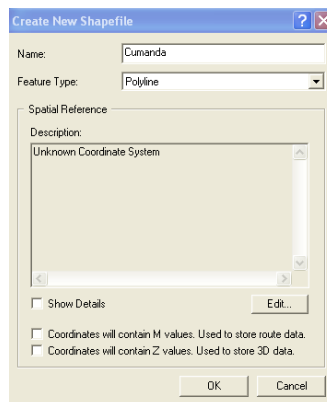
Georeferenciación de capas

Para poder visualizar las capas de la provincia de chimborazo es necesario que esten en coordenadas geograficas lo cual se realizó con la ayuda de la herramienta ArcGis.

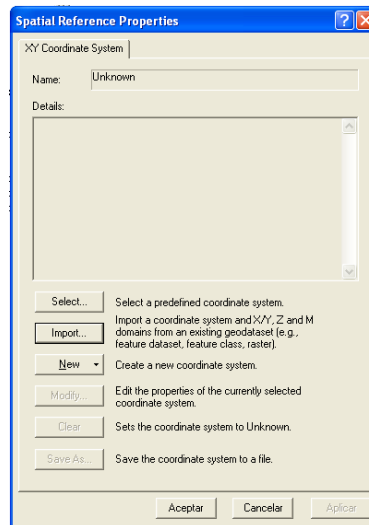
Plantilla para seleccionar el Sistema de Referencia Espacial.



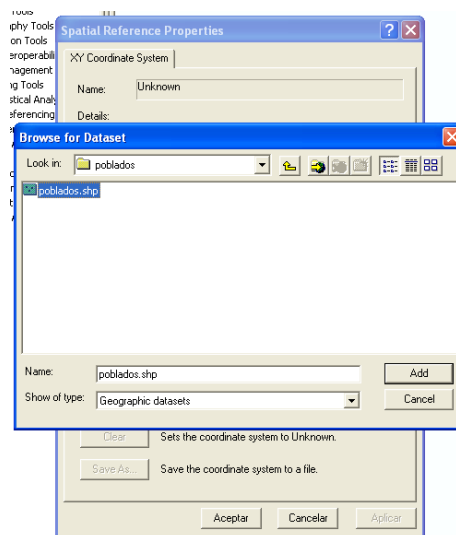
Crear un nuevo shape por ejemplo Cumanda y asignar el Sistema de Referencia Espacial además se debe especificar un dominio espacial más amplio, para lo cual se utilizó la capa poblaciones del IGM como plantilla para seleccionar el Sistema de Referencia Espacial.



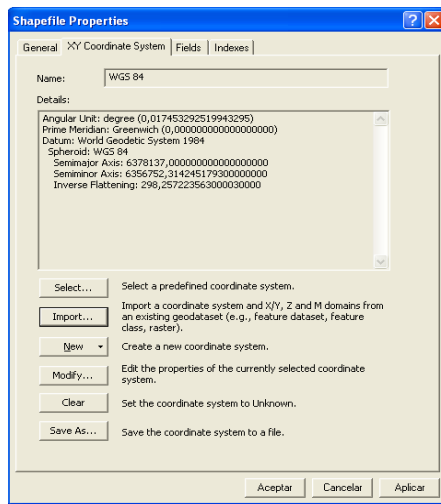
Asignar un sistema de coordenadas en este caso como se está utilizando una capa como plantilla escoja la opción **Import** la cual permite importar un sistema de referencia de un shape ya existente.



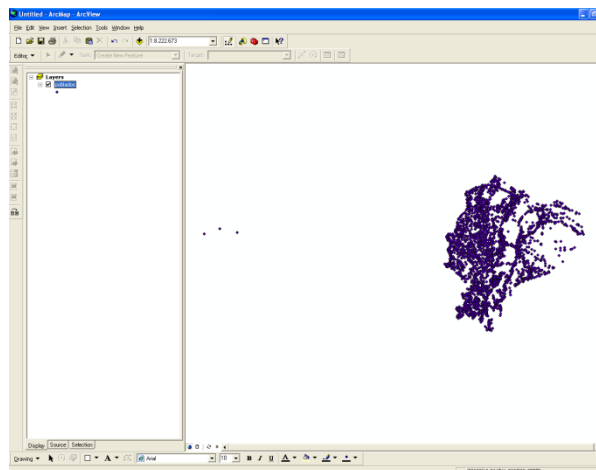
Seleccione la capa para importar el sistema de referencia.



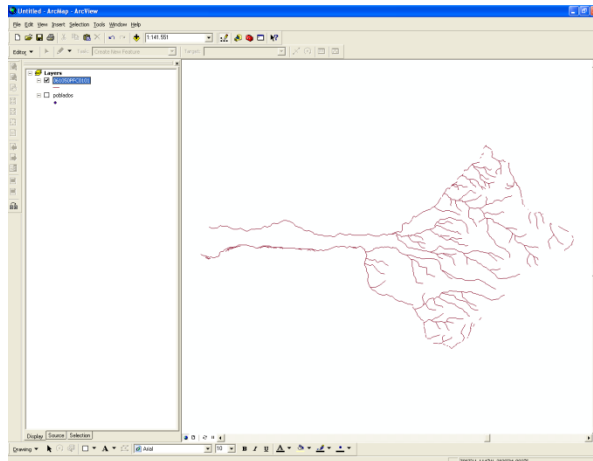
Aplicar y Aceptar el sistema de coordenada.



Añadir primero la capa que se utilizó como plantilla en este caso poblados.

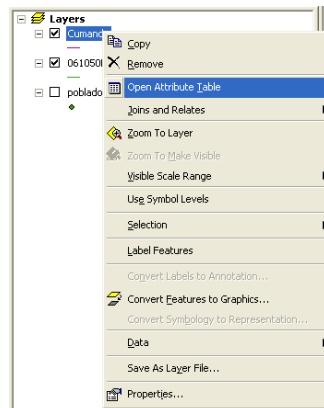


Luego añade la capa que desea cambiar del sistema de coordenadas en este caso 061050PFC0101 que contiene los datos de cumanda.

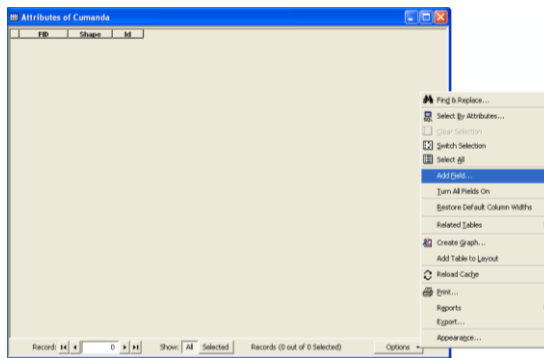


Y por último añadir la capa Cumanda para cargar los datos de la capa 061050PFC0101 pero en con otro sistema de referencia.

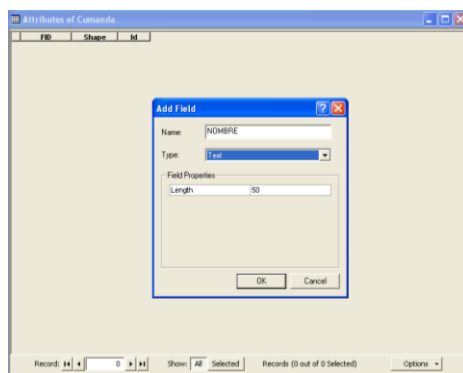
Clic derecho sobre la capa **Cumanda** y seleccione **Open Attribute Table** para añadir los mismos atributos de la capa 061050PFC0101.



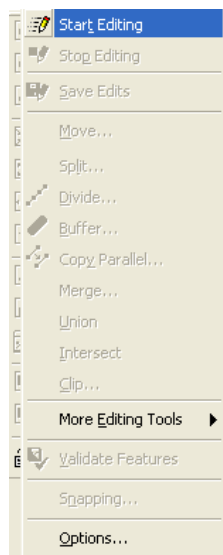
Añadir los campos



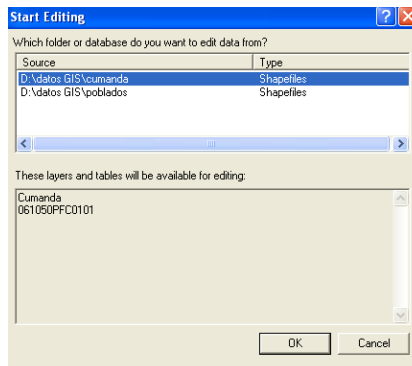
Ingrese el nombre y tipo de dato.



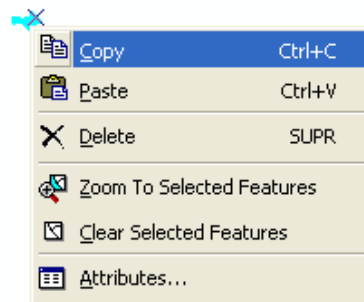
Para proceder a pasar los datos tiene que iniciar edición sobre la capa Cumanda



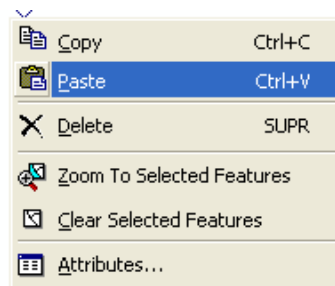
Elija la carpeta de la capa a editar.



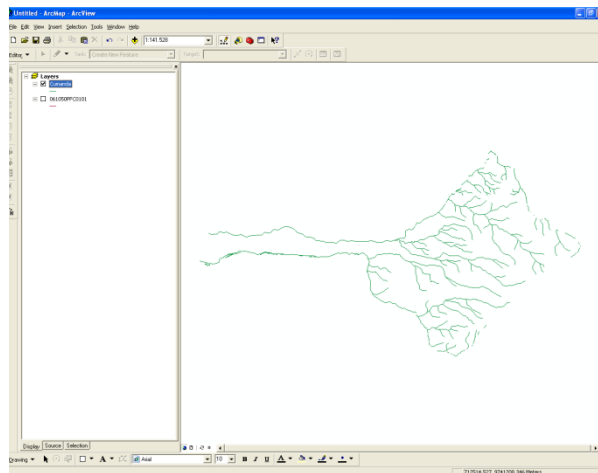
Copie los datos de la capa 061050PFC0101.



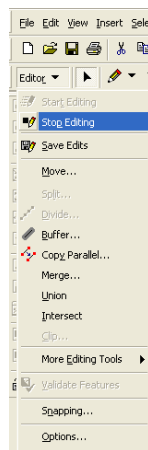
Pegue los datos en la capa Cumanda.



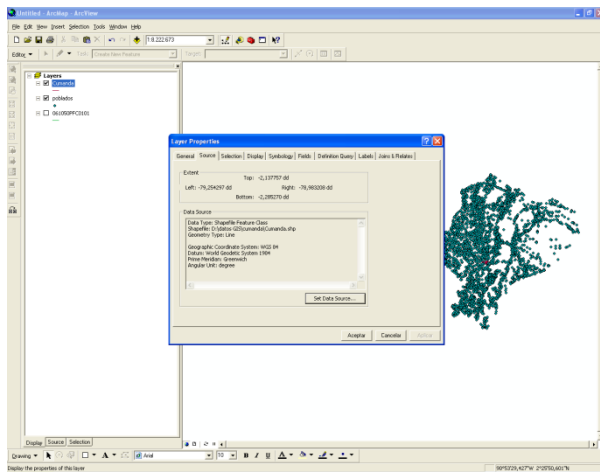
Resultado la nueva capa Cumanda



Guardar y finalizar edición



La capa Cumanda con un sistema con coordenadas geográficas.



ANEXO V

Manual de Usuario Geoportal

Manual de Usuario

Los componentes fundamentales del Geoportal proporcionan al usuario acceso a contenidos básicos de un portal estándar, teniendo como aplicativo central el Visor de Mapas, que desplegará la información geográfica de la provincia de Chimborazo.

El objetivo del manual es brindar al usuario una guía a seguir con el fin de la consecución y satisfacción de sus necesidades.

Ingrese la siguiente dirección <http://ide.epoch.edu.ec>, para acceder al geoportal.



Menú Principal

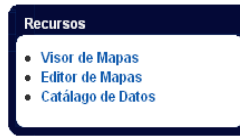
Contiene enlaces informativos sobre el geoportal.



Menú Recursos

Consta de lo siguiente:

- Visor de Mapas – Editor de mapas
- Catálogo de Metadatos

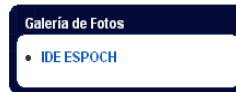


Menú Investigación

Contiene las tesis realizadas en la ESPOCH relacionadas con SIG.



Menú galería de fotos

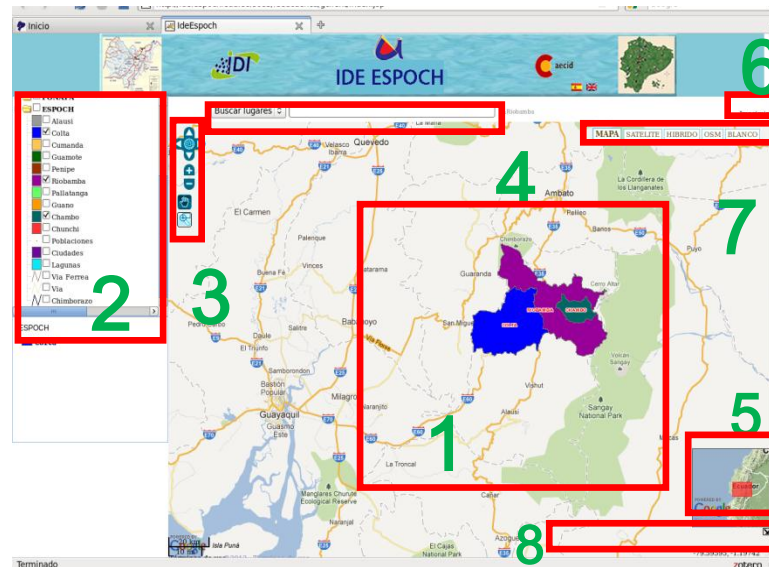


1. Visor de mapa

El visor es una aplicación que sirve para visualizar cada una de las capas. Mediante el visor se podrán superponer dichas capas de planeamiento sobre el mapa base que proporciona google maps. A todo ello se suma una serie de funcionalidades que permiten al usuario realizar zoom sobre los mapas, consultarlos, descargar o dibujar sobre ellos, obtener la leyenda de las capas activas y realizar búsquedas.

El conjunto total de características y funcionalidad que tendrá el geoportal es el siguiente:

Interfaz visor

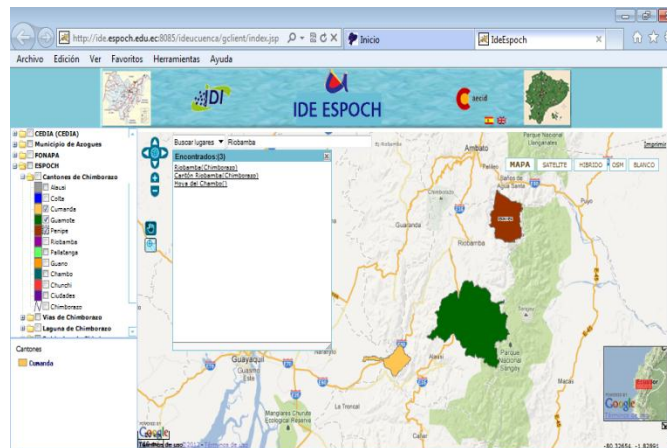


El visor dispondrá de las herramientas que se detallan a continuación.

1. **Ventana de visualización:** Área principal de trabajo.
2. **Panel de leyenda:** Representa las capas vigentes en el servidor, permitiendo controlar su estado de visualización.
3. **Barra de Herramientas:** Contiene herramientas que permiten la navegación por la cartografía, consta de la barra de zoom que permite aproximar o alejar la imagen y el paneo que permite desplazar la imagen.
4. **Búsqueda:** Permite ingresar el nombre de un objeto para su búsqueda, en caso de que el objeto sea encontrado automáticamente se mostrara el resultado.
5. **Mapa de Referencia:** Muestra la ubicación o posición del objeto en el instante de la navegación.
6. **Imprimir:** Permite imprimir la imagen del mapa que se está consultando en ese momento.
7. **Opciones de Visualización:** Despliega diferentes capas base.
8. **Panel de coordenadas:** Muestra referencia espacial.

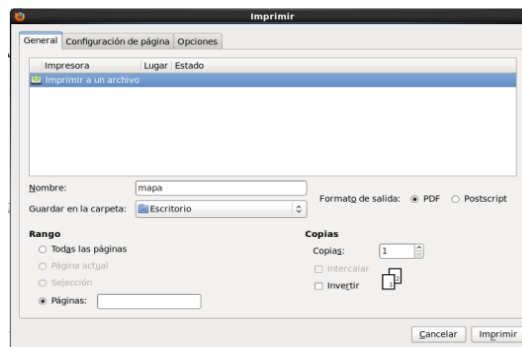
Búsqueda

Se ingresa en el campo de texto el nombre Riobamba y como se observa nos ubica en la posición donde se encuentra el objeto.



Impresión

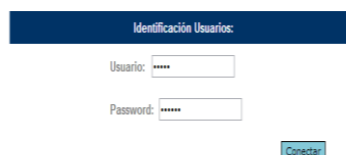
Se debe contestar o rellenar todas las opciones de configuración para la impresión y luego dar clic en imprimir.



Editor de objetos

Proporciona las herramientas necesarias para la edición de capas, como modificación o eliminación, así como el ingreso de capas.

Para realizar estas operaciones se requiere la autenticación del usuario.



Interfaz edición de mapas



El editor dispondrá de las siguientes herramientas:

1. Herramienta de Edición: Permite editar la capa.

A continuación se indicara la función de cada icono:



Iniciar edición.



Finalizar edición.



Borrar objeto.



Dibujar el objeto según el tipo que sea la capa: punto, línea o polígono.



Cargar archivos tipo .shp.



Eliminar capas de edición.




Descargar datos en formato .shp.

2. **Capas a editar:** En la columna izquierda se puede ver las capas que se encuentran en el editor, cabe mencionar que las capas que se van creando se agregan a esta lista. Para activar alguna de ellas dar clic sobre la capa que se desee.

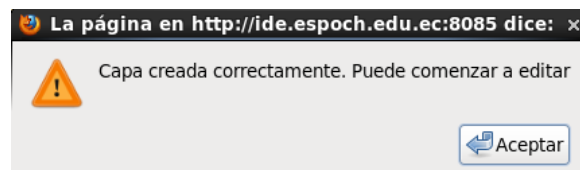
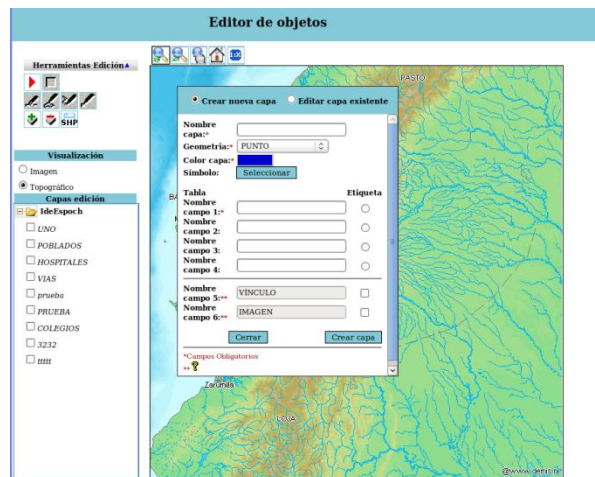
3. **Barra de herramientas:** Contiene herramientas que permiten la navegación por la cartografía, consta de la barra de zoom que permite aproximar o alejar la imagen y el paneo que permite desplazar la imagen.

4. **Área de Visualización:** Área principal de trabajo.

Creación de nueva capa

Para crear una nueva capa dar clic en .

La creación de una nueva capa implica dividir en dos partes principales: la primera crear los atributos de la capa, tales como nombre, tipo de geometría (línea, punto o polígono), símbolo con el cual se desea representar el objeto, luego dar clic en crear capa.



La nueva capa se ubicará en la columna izquierda, dar clic sobre la misma y después clic en la posición que se desee ubicar al objeto luego de lo cual se visualizará la siguiente

ventana en donde debería rellenar sus atributos con la información correspondiente, después dar clic en crear objeto.

Editar atributos

Nom capa: **COLEGIOS**

ID:

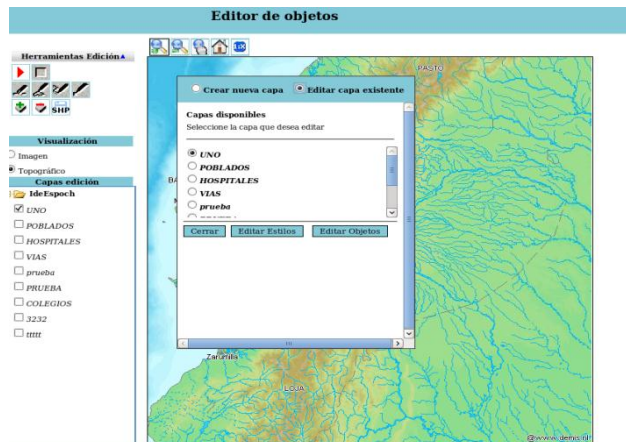
NOMBRE:

Luego se visualiza así el punto nuevo creado (azul), sobre el mapa.

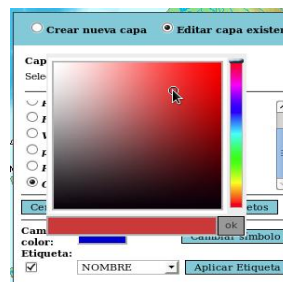


Modificación de capa

Se debe escoger la capa a editar.



Luego dar clic en cambiar estilos, para que permita cambiar el color y además se muestre la información del nombre sobre el punto.

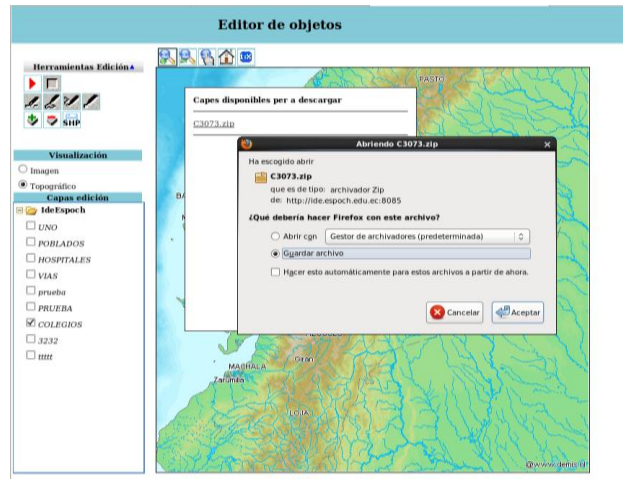


Es así como la capa colegio es modificada correctamente sobre el mapa.



Descarga de datos

Para la descarga de datos en formato .shp dar clic en



Galería de fotos

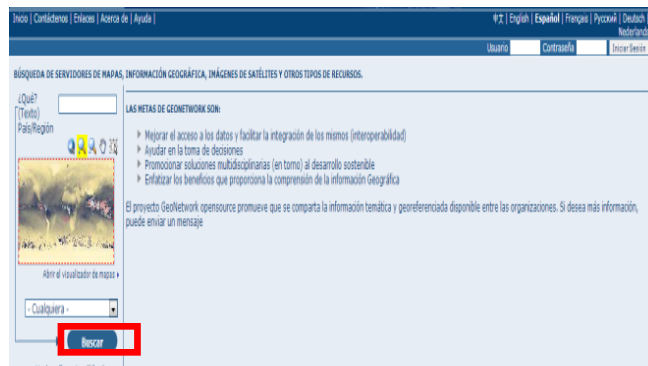
En parte izquierda, dar clic en el enlace Galería de fotos para poder visualizar las fotos.



Catálogo de datos

Permite difundir información referente a los datos publicados de la provincia de Chimborazo.

Dar clic en Buscar.



En caso de que exista información se visualiza la siguiente página que muestra los resultados de una búsqueda simple, indicando el título, parte del resumen, página de metadatos y palabras clave.

