



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“APLICACIÓN DISTRIBUIDA WEB-MÓVIL ADMINISTRABLE PARA LA
GESTIÓN Y DIFUSIÓN GEO-LOCALIZADA DE ATRACTIVOS
TURÍSTICOS Y HOTELES PARA LA CIUDAD DE IBARRA, CON
TECNOLOGÍA GIS Y SOFTWARE LIBRE”**

AUTOR: JAIRO ALBERTO MORALES DE LA TORRE

DIRECTOR: ING. IRVING REASCOS

IBARRA – ECUADOR

2013



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

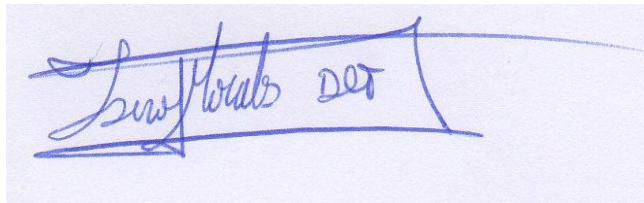
La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

DATOS DEL CONTACTO			
CEDULA DE IDENTIDAD:	100309021-2		
APELLIDOS Y NOMBRES	MORALES DE LA TORRE JAIRO ALBERTO		
DIRECCIÓN:	CALLE MODESTO LARREA Y LINEA FERREA ESQUINA (SAN ROQUE)		
EM@IL:	dmxjairo@hotmail.com		
TELÉFONO FIJO:	062909606	TELÉFONO MÓVIL:	0988081128
DATOS DE LA OBRA			
TÍTULO	Aplicación distribuida web-móvil administrable para la gestión y difusión geo-localizada de atractivos turísticos y hoteles para la ciudad de Ibarra, con tecnología GIS y software libre.		
AUTOR:	MORALES DE LA TORRE JAIRO ALBERTO		
FECHA:	ABRIL -2013		

2.- AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Jairo Alberto Morales de la Torre, con cédula de identidad Nro. 100309021-2, en calidad de autor y titular de los derechos patrimoniales del Trabajo de Grado, descrito anteriormente; hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad, con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior en su Artículo 144.



Firma -----

Nombre: Jairo Alberto Morales de la Torre

Cédula: 100309021-2

Ibarra, Diciembre 2013



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA
UNIVERSIDAD TÉCNICA DEL NORTE

Yo Jairo Alberto Morales del Torre, con cédula de identidad Nro. 100309021-2, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor (es) de la obra o trabajo de grado denominado: “Aplicación distribuida web-móvil administrable para la gestión y difusión geo-localizada de atractivos turísticos y hoteles para la ciudad de Ibarra, con tecnología GIS y software libre”, que ha sido desarrollado para optar por el título de: Ingeniero en Sistemas Computacionales , en la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo ente documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

Firma -----

Nombre: Jairo Alberto Morales de la Torre

Cédula: 100309021-2

Ibarra, Diciembre 2013



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL ASESOR

En calidad de Director del Trabajo de grado presentado por Jairo Alberto Morales de la Torre, para optar por el título de Ingeniero en Sistemas Computacionales, cuyo tema es: **“APLICACIÓN DISTRIBUIDA WEB-MÓVIL ADMINISTRABLE PARA LA GESTIÓN Y DIFUSIÓN GEO-LOCALIZADA DE ATRACTIVOS TURÍSTICOS Y HOTELES PARA LA CIUDAD DE IBARRA, CON TECNOLOGÍA GIS Y SOFTWARE LIBRE”**.

Considero que el presente trabajo reúne los requisitos y méritos suficientes para ser sometidos a la presentación pública y evaluación por parte del tribunal examinador que se designe.

En la ciudad de Ibarra, Diciembre del 2013

Ing. Irving Reascos

DIRECTOR DEL TRABAJO DE GRADO



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asumo la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

En la ciudad de Ibarra, Diciembre del 2013

EL AUTOR:

Jairo Alberto Morales de la Torre

C.I.: 100309021-2



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DECLARACIÓN

Yo, Jairo Alberto Morales de la Torre, declaro bajo juramento que el trabajo aquí escrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Técnica del Norte- Ibarra, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Jairo Alberto Morales de la Torre

C.I.: 100309021-2



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

A mí madre, porque tuvo fe en mí y siempre estuvo a mi lado, dándome motivación y entrega, gracias a esos impulsos constantes en los momentos más difíciles de mi carrera.

A mí padre por todo el apoyo incondicional y consejos magistrales.

A mí abuelito Juan Morales por ser un hombre muy trabajador y honesto, él siempre me decía hijo siempre has tus tareas y después tendrás tiempo para jugar.

A mi hijo, Jairito, quien me motivo con sus sonrisas, para culminar mi meta de ser un profesional.



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

Agradezco A DIOS por concederme la vida a mi familia por estar allí cuando los necesitaba y brindarme su apoyo incondicional en todos los momentos.

Agradezco a la Universidad Técnica del Norte, y a la Carrera de Ingeniería en Ciencias Computacionales, a todos sus Docentes que depositaron sus conocimientos en mí permitiéndome realizar este trabajo.

Agradezco de manera especial a mí Director de Tesis, Ing. Irving Reascos por su paciencia y su guía durante el proceso de creación de la presente.

Agradezco de manera especial a mí Docente de Trabajo de Grado II, Ing. Omar Lara por sus sugerencias durante el proceso de creación de la presente.

RESUMEN

El presente proyecto de tesis consiste en desarrollar e implementar un gestor de contenidos Web, para un aplicativo móvil que disponga de un sistema operativo Android, que sea capaz de promocionar los sitios turísticos del Ecuador de una manera innovadora y eficiente, con especial énfasis en la utilización de Software Libre y Sistemas de Información Geográfica.

En el primer capítulo se describen las fases de la metodología XP (Programación Extrema): Exploración, Planeación, Iteraciones y Producción, que se realizan paso a paso en el desarrollo del sistema, además se utilizó estándares de calidad aceptados por la W3C y la API¹ de Android.

El segundo capítulo se refiere al estudio de factibilidad, que fue realizado aplicando el tamaño de la muestra a los habitantes de la zona urbana de Ibarra, con el fin de conocer el uso de los teléfonos inteligentes, especialmente los teléfonos con Sistema Operativo Android, además se realizó el análisis de resultados de las encuestas realizadas.

En el tercer capítulo se desarrolla el sistema, tomando como referencia las fases de la metodología XP (Programación Extrema).

En la fase de exploración se definen las historias de usuario y la arquitectura del sistema, en la fase de planeación se establece un cronograma de actividades de acorde a los tiempos de desarrollo estimado por el programador, en la fase de iteraciones se desarrolla las tareas establecidas y finalmente en la fase de producción se publica la aplicación web y móvil.

El cuarto capítulo cubre el análisis de costo en base a las herramientas utilizadas y servicios web solicitados.

Para finalizar se redacta las conclusiones y recomendaciones del sistema desarrollado.

¹ Application programming interface: Interfaz de programación de aplicaciones.

SUMMARY

Present project of thesis consists of develop and to implement a Web content manager, for an application mobile that has an Android operating system , for promoting the tourist sites of the Ecuador of an innovative and efficient way, with special emphasis in the utilization of Free Software and Geographical information systems.

In the first chapter the phases of methodology XP are described (Extreme Programming): Exploration, Planning, Iterations and Production, that are made step by step in the development of the system, in addition were used standards of quality accepted by the W3C and the API of Android.

The second chapter talks about the feasibility study, that was made applying the sample size to the inhabitants of the urban zone of Ibarra, with the purpose of knowing the use the intelligent telephones, specially the telephones with Android Operating System, in addition was made the analysis of results of the made surveys.

In the third chapter it is developed to the system, taking like reference the phases of methodology XP (Extreme Programming).

In the phase of exploration user histories are defined and the architecture of the system, in the phase of planning settles down a chronogram of activities of agreed to the times of development considered by the programmer, in the phase of iterations it is developed the established tasks and finally in the phase of production the application is published Web and moving body.

The fourth chapter covers the analysis of cost on the basis of the tools used and web services requested.

The fourth chapter covers the study for analysis of cost on the basis of the used tools and services Web.

Finally the conclusions and recommendations of the developed system are drafted.

ÍNDICE DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO	iv
CERTIFICACIÓN DEL ASESOR	v
CONSTANCIAS	vi
DECLARACIÓN	vii
DEDICATORIA	viii
AGRADECIMIENTO	ix
RESUMEN	x
SUMMARY	xi
ÍNDICE DE CONTENIDOS	xii
ANTEPROYECTO APROBADO POR EL CONSEJO ACADÉMICO	xxiii
CAPÍTULO I	1
1 Metodología y Herramientas de Desarrollo	1
1.1 Metodología XP	1
1.1.1 Roles XP	1
1.1.2 Valores XP	3
1.1.3 Ventajas y Desventajas de la metodología XP	5
1.1.4 Fases de la metodología XP	5
1.2 AJAX	6
1.2.1 JAVASCRIPT	7
1.2.2 Fases en la ejecución de una aplicación AJAX	13
1.2.3 Arquitectura AJAX	14
1.2.4 Ventajas y desventajas de utilizar AJAX	16
1.3 Base de Datos	17
1.3.1 Bases de datos relacionales	18
1.3.2 Ventajas y desventajas de utilizar base de datos relacionales	18
1.3.3 PostgreSQL	19
1.3.4 Arquitectura por capas	19
1.4 Sistemas de Información Geográfica	20
1.4.1 Base de datos espaciales	21
1.4.2 postGIS	22
1.5 Open Layers	23
1.5.1 Proyecciones de un mapa	24
1.5.2 European Petroleum Survey Group	24
1.5.3 Mostar un mapa con OpenLayers	27
1.5.4 Definir proyecciones	28
1.6 Servidor Web	29
1.6.1 Funciones de un servidor Web	29

1.6.2	Servidores web más utilizados	29
1.6.3	Operaciones que realiza un servidor	30
1.6.4	Ventajas de los servidores Apache e IIS	30
1.6.5	Desventajas de los servidores Apache e IIS	31
1.7	Aplicaciones distribuidas	31
1.7.1	La separación física	32
1.7.2	Distribución lógica	32
1.7.3	Servicios Web	32
1.8	Lenguajes de programación y librerías	35
1.8.1	PHP	35
1.8.2	JAVA	35
1.8.3	TWITTER BOOTSTRAP	35
1.9	SSOO Móviles.....	37
1.9.1	SSOO más utilizados	37
1.9.2	Tabla comparativa de SSOO móviles	38
1.9.3	SSOO Android.....	39
1.9.4	Google Play.....	43
CAPÍTULO II		45
2	Estudio de Factibilidad	45
2.1	Mercado	45
2.2	Definición del producto.....	45
2.3	Selección de la muestra	46
2.3.1	Determinación del tamaño de la muestra	46
2.4	Factores de la encuesta	48
2.5	Análisis e interpretación de resultados.....	48
2.6	Conclusiones del estudio realizados	55
CAPÍTULO III		56
3	Desarrollo de la solución	56
3.1	Fase de exploración	56
3.1.1	Historias de usuario.....	56
3.1.2	Modelo Vista Controlador.....	78
3.1.3	Arquitectura funcional del sistema.....	79
3.2	Fase de planeación	80
3.2.1	Cronograma de actividades.....	80
3.2.2	Módulos de sistema	82
3.3	Iteración I.....	83
3.3.1	Cronograma de actividades.....	83
3.3.2	Tareas por historia Iteración I.....	84
3.3.3	Pruebas Iteración I	101
3.3.4	Resultado de la 1ª Iteración de construcción	108

3.3.5 Demo de la versión, Iteración I	109
3.3.6 Pruebas de aceptación	114
3.3.7 Incidencias	114
3.4 Iteración II.....	115
3.4.1 Cronograma de actividades.....	115
3.4.2 Tareas por historia iteración II	116
3.4.3 Pruebas Iteración II	147
3.4.4 Resultado de la 2ª Iteración de construcción	164
3.4.5 Memo de la versión, Iteración II.....	165
3.4.6 Pruebas de aceptación.....	178
3.4.7 Incidencias	180
3.5 Iteración III.....	180
3.5.1 Cronograma de actividades.....	180
3.5.2 Tareas por historia Iteración III	181
3.5.3 Pruebas de Iteración III	207
3.5.4 Resultado de la 3ª Iteración de construcción	216
3.5.5 Demo de la versión.....	218
3.5.6 Pruebas de aceptación.....	224
3.5.7 Incidencias	226
3.5.8 Diagrama de base de Base de datos	227
3.6 Producción	227
3.6.1 Pasos para la configuración de servidor LINUX	228
3.6.2 Publicar la aplicación móvil en Google Play	228
CAPÍTULO IV	229
4 Análisis Costo-Beneficio, Conclusiones y Recomendaciones	229
4.1 Análisis de Costo-beneficio	229
4.1.1 Objetivo	229
Costos de Hardware.....	229
4.1.3 Políticas de precio	230
4.1.4 Ejemplo Ilustrativo	230
4.1.5 Conclusiones del analisis Costo-beneficio	232
4.1.6 Riegos y oportunidades.....	233
4.2 CONCLUSIONES.....	234
4.3 RECOMENDACIONES	235
GLOSARIO.....	236
BIBLIOGRAFÍA	240
ANEXOS	242

ÍNDICE DE ILUSTRACIONES

Figura a: Esquema general del sistema Anteproyecto	xxviii
Figura b: Arquitectura funcional del sistema Anteproyecto	xxix
Ilustración 1: Fases de la metodología XP	5
Ilustración 2: Esquema de tecnologías AJAX.....	7
Ilustración 5: Funcionamiento de una Base de Datos	17
Ilustración 6: Arquitectura en dos capas	19
Ilustración 7: Arquitectura en tres capas	20
Ilustración 8: Sistemas de información geográfica	21
Ilustración 9: Proyecciones de un mapa.....	24
Ilustración 10: Proyección Mercator cilíndrica.	26
Ilustración 11: Capa base de OSM.....	28
Ilustración 12: Arquitectura funcional de un Servidor Web.....	29
Ilustración 13: Arquitectura de una aplicación distribuida.....	31
Ilustración 14: Esquema de los Servicios web	33
Ilustración 15 : Botones bootstrap	36
Ilustración 16: Arquitectura de Android	39
Ilustración 17 : Ciclo de vida de una aplicación Android	42
Ilustración 18: Gráfico estadísticos correspondiente a la pregunta 1	49
Ilustración 19: Gráfico estadísticos correspondiente a la pregunta 2	50
Ilustración 21: Gráfico estadísticos correspondiente a la pregunta 4	52
Ilustración 22: Gráfico estadísticos correspondiente a la pregunta 5	53
Ilustración 23: Gráfico estadísticos correspondiente a la pregunta 6	54
Ilustración 25: Arquitectura del sistema.....	78
Ilustración 26: Arquitectura funcional del sistema	80
Ilustración 27: Módulos del sistema	82
Ilustración 28: Formulario para ingresar al sistema	109
Ilustración 29: Formulario para ingresar una entidad usuario	110
Ilustración 30: Formulario para editar una entidad usuarios.....	110
Ilustración 31: Formulario para ingresar una entidad rol	111
Ilustración 32: Formulario para editar una entidad rol	111
Ilustración 33: Formulario para ingresar una entidad responsabilidad	112
Ilustración 34: Formulario para editar una entidad responsabilidad	112
Ilustración 35: Formulario para gestionar los permisos al usuario.....	113
Ilustración 36: Formulario para ingresar una entidad provincia.....	165
Ilustración 37: Formulario para editar una entidad provincia.....	166
Ilustración 38: Formulario para ingresar una entidad ciudad.....	166
Ilustración 39: Formulario para editar una entidad ciudad.....	167
Ilustración 40: Formulario para ingresar una entidad categoría	167

Ilustración 41: Formulario para editar una entidad categoría	168
Ilustración 42: Formulario para ingresar una entidad sitio	169
Ilustración 43: Formulario para editar una entidad sitio	170
Ilustración 44: Formulario para ingresar una entidad historia.....	171
Ilustración 45: Formulario para editar una entidad historia.....	171
Ilustración 46: Formulario para ingresar una entidad video.....	172
Ilustración 47: Formulario para editar una entidad video.....	172
Ilustración 48: Formulario para ingresar una entidad foto	173
Ilustración 49: Formulario para editar una entidad foto	173
Ilustración 50: Formulario para ingresar una entidad festivo	174
Ilustración 51: Formulario para editar una entidad festivo	174
Ilustración 52: Formulario para ingresar una entidad gastronomía	175
Ilustración 53: Formulario para editar una entidad gastronomía	175
Ilustración 54: Formulario para ingresar una entidad ruta	176
Ilustración 55: Formulario para editar una entidad ruta	177
Ilustración 56: Formulario para buscar sitios turísticos.....	218
Ilustración 57: Formulario para realizar una geo-búsqueda	219
Ilustración 58: Enlaces hacia historias	220
Ilustración 59: Enlaces hacia videos YouTube	220
Ilustración 60: Enlaces hacia imágenes	221
Ilustración 61: Enlaces hacia días festivos	221
Ilustración 62: Enlaces hacia gastronomías	222
Ilustración 63: Enlaces hacia rutas.....	222
Ilustración 64: Reporte plano de sitios turísticos (App móvil)	223
Ilustración 65: Reporte de historias y videos (App móvil)	223
Ilustración 66: Reporte de festivos, fotos y gastronomías (App móvil)	224
Ilustración 67: Modelo físico de la base de datos	227
Ilustración 68: Regresión lineal	232

ÍNDICE DE TABLAS

Tabla 1: Roles de la Metodología XP	1
Tabla 2: Integrantes en un Proyecto XP.....	3
Tabla 3: Ventajas y desventajas de la metodología XP	5
Tabla 4: Diferencias entre los estándares HTML y XHTML	8
Tabla 5: propiedades elementales del objeto XMLHttpRequest.....	10
Tabla 6: Representación de los valores de respuesta readyState	10
Tabla 7: Métodos definidos por AJAX para la transferencia de información	11
Tabla 8: Ventajas y desventajas de utilizar AJAX	16
Tabla 9: Ventajas y desventajas de utilizar base de datos relacionales.....	18
Tabla 10: Códigos EPGS utilizados, En la definición de datos de posición	25
Tabla 11: Ventajas de los servidores Apache y IIS	30
Tabla 12: Desventajas de los servidores Apache e IIS	31
Tabla 13: Características principales de los SSOO móviles	38
Tabla 14: Ventajas y desventajas de las aplicaciones móviles	43
Tabla 15: Frecuencia correspondiente a la pregunta 1	48
Tabla 16: Frecuencia correspondiente a la pregunta 2	49
Tabla 17: Frecuencia correspondiente a la pregunta 3	50
Tabla 18: Frecuencia correspondiente a la pregunta 4	51
Tabla 19: Frecuencia correspondiente a la pregunta 5	52
Tabla 20: Frecuencia correspondiente a la pregunta 6	53
Tabla 21: Frecuencia correspondiente a la pregunta 7	54
Tabla 22: Historia Nro. 1	56
Tabla 23: Historia Nro. 2	57
Tabla 24: Historia Nro. 3	58
Tabla 25: Historia Nro. 4	59
Tabla 26: Historia Nro. 5	60
Tabla 27: Historia Nro. 6	61
Tabla 28: Historia Nro. 7	62
Tabla 29: Historia Nro. 8	63
Tabla 30: Historia Nro. 9	64
Tabla 31: Historia Nro. 10	65
Tabla 32: Historia Nro. 11	66
Tabla 33: Historia Nro. 12	67
Tabla 34: Historia Nro. 13	68
Tabla 35: Historia Nro. 14	69
Tabla 36: Historia Nro. 15	70
Tabla 37: Historia Nro. 16	71
Tabla 38: Historia Nro. 17	72

Tabla 39: Historia Nro. 18.....	73
Tabla 40: Historia Nro. 19.....	73
Tabla 41: Historia Nro. 20.....	74
Tabla 42: Historia Nro. 21.....	74
Tabla 43: Historia Nro. 22.....	75
Tabla 44: Historia Nro. 23.....	75
Tabla 45: Historia Nro. 24.....	76
Tabla 46: Historia Nro. 25.....	76
Tabla 47: Historia Nro. 26.....	77
Tabla 48: Cronograma de actividades.....	81
Tabla 49: Cronograma de actividades, Iteración I.....	83
Tabla 50: Tareas de acceso al sistema.....	84
Tabla 51: Acceso al sistema - Tarea Nro. 1.....	84
Tabla 52: Acceso al sistema - Tarea Nro. 2.....	85
Tabla 53: Acceso al sistema - Tarea Nro. 3.....	85
Tabla 54: Acceso al sistema - Tarea Nro. 4.....	86
Tabla 55: Acceso al sistema - Tarea Nro. 5.....	86
Tabla 56: Tareas de gestión de usuarios.....	87
Tabla 57: Gestión de usuarios – Tarea Nro. 1.....	87
Tabla 58: Gestión de usuarios – Tarea Nro. 2.....	88
Tabla 59: Gestión de usuarios – Tarea Nro. 3.....	89
Tabla 60: Gestión de usuarios – Tarea Nro. 4.....	89
Tabla 61: Gestión de usuarios – Tarea Nro. 5.....	90
Tabla 62: Tareas de asignación de roles.....	90
Tabla 63: Asignación de roles – Tarea Nro. 1.....	91
Tabla 64: Asignación de roles – Tarea Nro. 2.....	91
Tabla 65: Asignación de roles – Tarea Nro. 3.....	92
Tabla 66: Asignación de roles – Tarea Nro. 4.....	93
Tabla 67: Asignación de roles – Tarea Nro. 5.....	93
Tabla 68: Tareas de asignación de responsabilidades.....	94
Tabla 69: Asignación de responsabilidades – Tarea Nro. 1.....	94
Tabla 70: Asignación de responsabilidades – Tarea Nro. 2.....	95
Tabla 71: Asignación de responsabilidades – Tarea Nro. 3.....	96
Tabla 72: Asignación de responsabilidades – Tarea Nro. 4.....	96
Tabla 73: Asignación de responsabilidades – Tarea Nro. 5.....	97
Tabla 74: Tareas de asignación de permisos hacia pantallas.....	97
Tabla 75: Asignación de permisos hacia pantallas – Tarea Nro. 1.....	98
Tabla 76: Asignación de permisos hacia pantallas – Tarea Nro. 2.....	98
Tabla 77: Asignación de permisos hacia pantallas – Tarea Nro. 3.....	99
Tabla 78: Asignación de permisos hacia pantallas – Tarea Nro. 4.....	100

Tabla 79: Asignación de permisos hacia pantallas – Tarea Nro. 5.....	100
Tabla 80: Iteración I, Historial de revisiones Nro. 1	101
Tabla 81: Iteración I, Historial de revisiones Nro. 2	102
Tabla 82: Iteración I, Historial de revisiones Nro. 3	104
Tabla 83: Iteración I, Historial de revisiones Nro. 4	106
Tabla 84: Iteración I, Historial de revisiones Nro. 5	107
Tabla 85: Cronograma de actividades Iteración II	115
Tabla 86: Tareas de gestión de provincias.....	116
Tabla 87: Gestión de provincias – Tarea Nro. 1	116
Tabla 88: Gestión de provincias – Tarea Nro. 2	117
Tabla 89: Gestión de provincias – Tarea Nro. 3.....	118
Tabla 90: Gestión de provincias – Tarea Nro. 4.....	118
Tabla 91: Tareas de gestión de ciudades	119
Tabla 92: Gestión de ciudades – Tarea Nro. 1	119
Tabla 93: Gestión de ciudades – Tarea Nro. 2.....	120
Tabla 94: Gestión de ciudades – Tarea Nro. 3.....	120
Tabla 95: Gestión de ciudades – Tarea Nro. 4.....	121
Tabla 96: Gestión de ciudades – Tarea Nro. 5.....	121
Tabla 97: Tareas de gestión de categorías	122
Tabla 98: Gestión de categorías – Tarea Nro. 1	122
Tabla 99: Gestión de categorías – Tarea Nro. 2	123
Tabla 100: Gestión de categorías – Tarea Nro. 3	123
Tabla 101: Gestión de categorías – Tarea Nro. 4	124
Tabla 102: Tareas de Gestión de sitios.....	124
Tabla 103: Gestión de sitios – Tarea Nro. 1	125
Tabla 104: Gestión de sitios – Tarea Nro. 2.....	125
Tabla 105: Gestión de sitios – Tarea Nro. 3.....	126
Tabla 106: Gestión de sitios – Tarea Nro. 4.....	126
Tabla 107: Gestión de sitios – Tarea Nro. 5.....	127
Tabla 108: Gestión de sitios – Tarea Nro. 6.....	127
Tabla 109: Gestión de sitios – Tarea Nro. 7	128
Tabla 110: Tareas de Gestión de historias.....	128
Tabla 111: Gestión de historias – Tarea Nro. 1	129
Tabla 112: Gestión de historias – Tarea Nro. 2.....	129
Tabla 113: Gestión de historias – Tarea Nro. 3.....	130
Tabla 114: Gestión de historias – Tarea Nro. 4.....	131
Tabla 115: Gestión de historias – Tarea Nro. 5.....	131
Tabla 116: Tareas de gestión de URL'S hacia YouTube	132
Tabla 117: Gestión de URL'S hacia YouTube – Tarea Nro. 1	132
Tabla 118: Gestión de URL'S hacia YouTube – Tarea Nro. 2	133

Tabla 119: Gestión de URL'S hacia YouTube – Tarea Nro. 3	133
Tabla 120: Gestión de URL'S hacia YouTube – Tarea Nro. 4	134
Tabla 121: Gestión de URL'S hacia YouTube – Tarea Nro. 5	134
Tabla 122: Tareas de Gestión imágenes.....	135
Tabla 123: Gestión de imágenes – Tarea Nro. 1.....	135
Tabla 124: Gestión de imágenes – Tarea Nro. 2.....	136
Tabla 125: Gestión de imágenes – Tarea Nro. 3.....	136
Tabla 126: Gestión de imágenes – Tarea Nro. 4.....	137
Tabla 127: Gestión de imágenes – Tarea Nro. 5.....	137
Tabla 128: Tareas de Gestión de festivos	138
Tabla 129: Gestión de festivos – Tarea Nro. 1	138
Tabla 130: Gestión de festivos – Tarea Nro. 2.....	139
Tabla 131: Gestión de festivos – Tarea Nro. 3.....	139
Tabla 132: Gestión de festivos – Tarea Nro. 4.....	140
Tabla 133: Gestión de festivos – Tarea Nro. 5.....	140
Tabla 134: Tareas de gestión de gastronomías	141
Tabla 135: Gestión de gastronomías – Tarea Nro. 1	141
Tabla 136: Gestión de gastronomías – Tarea Nro. 2	142
Tabla 137: Gestión de gastronomías – Tarea Nro. 3	142
Tabla 138: Gestión de gastronomías – Tarea Nro. 4	143
Tabla 139: Gestión de gastronomías – Tarea Nro. 5	143
Tabla 140: Tarea de gestión de rutas.....	144
Tabla 141: Gestión de rutas – Tarea Nro. 1	144
Tabla 142: Gestión de rutas – Tarea Nro. 2	145
Tabla 143: Gestión de rutas – Tarea Nro. 3	145
Tabla 144: Gestión de rutas – Tarea Nro. 4	146
Tabla 145: Gestión de rutas – Tarea Nro. 5	146
Tabla 146: Gestión de rutas – Tarea Nro. 6	147
Tabla 147: Iteración II, Historial de revisiones Nro. 1	147
Tabla 148: Iteración II, Historial de revisiones Nro. 2	149
Tabla 149: Iteración II, Historial de revisiones Nro. 3	151
Tabla 150: Iteración II, Historial de revisiones Nro. 4	153
Tabla 151: Iteración II, Historial de revisiones Nro. 5	154
Tabla 152: Iteración II, Historial de revisiones Nro. 6	156
Tabla 153: Iteración II, Historial de revisiones Nro. 7	158
Tabla 154: Iteración II, Historial de revisiones Nro. 8	159
Tabla 155: Iteración II, Historial de revisiones Nro. 9	161
Tabla 156: Iteración II, Historial de revisiones Nro. 10	163
Tabla 157: Cronograma de actividades Iteración III	180
Tabla 158: Tareas de reporte plano de sitios	181

Tabla 159: Reporte plano de sitios – Tarea Nro. 1.....	181
Tabla 160: Reporte plano de sitios – Tarea Nro. 2.....	182
Tabla 161: Reporte plano de sitios – Tarea Nro. 3.....	182
Tabla 162: Reporte plano de sitios – Tarea Nro. 4.....	183
Tabla 163: Tareas de reporte geo-referenciado.....	183
Tabla 164: Reporte geo-referenciado – Tarea Nro. 1.....	184
Tabla 165: Reporte geo-referenciado – Tarea Nro. 2.....	184
Tabla 166: Reporte geo-referenciado – Tarea Nro. 3.....	185
Tabla 167: Reporte geo-referenciado – Tarea Nro. 4.....	185
Tabla 168: Tareas de enlaces hacia historias.....	186
Tabla 169: Reporte geo-referenciado – Tarea Nro. 1.....	186
Tabla 170: Enlaces hacia historias – Tarea Nro. 2.....	186
Tabla 171: Enlaces hacia historias – Tarea Nro. 3.....	187
Tabla 172: Enlaces hacia historias – Tarea Nro. 4.....	187
Tabla 173: Tareas de enlace hacia YouTube.....	188
Tabla 174: Enlaces hacia YouTube – Tarea Nro. 1.....	188
Tabla 175: Enlaces hacia YouTube – Tarea Nro. 2.....	189
Tabla 176: Enlaces hacia YouTube – Tarea Nro. 3.....	189
Tabla 177: Enlaces hacia YouTube – Tarea Nro. 4.....	190
Tabla 178: Tareas de enlaces hacia imágenes.....	190
Tabla 179: Enlaces hacia imágenes – Tarea Nro. 1.....	191
Tabla 180: Enlaces hacia imágenes – Tarea Nro. 2.....	191
Tabla 181: Enlaces hacia imágenes – Tarea Nro. 3.....	192
Tabla 182: Enlaces hacia imágenes – Tarea Nro. 4.....	192
Tabla 183: Tareas de enlaces hacia festivos.....	193
Tabla 184: Enlaces hacia festivos – Tarea Nro. 1.....	193
Tabla 185: Enlaces hacia festivos – Tarea Nro. 2.....	194
Tabla 186: Enlaces hacia festivos – Tarea Nro. 3.....	194
Tabla 187: Enlaces hacia festivos – Tarea Nro. 4.....	195
Tabla 188: Tareas de enlaces hacia gastronomía.....	195
Tabla 189: Enlaces hacia gastronomía – Tarea Nro. 1.....	196
Tabla 190: Enlaces hacia gastronomía – Tarea Nro. 2.....	196
Tabla 191: Enlaces hacia gastronomía – Tarea Nro. 3.....	197
Tabla 192: Enlaces hacia gastronomía – Tarea Nro. 4.....	197
Tabla 193: Tarea de enlaces hacia rutas.....	198
Tabla 194: Enlaces hacia rutas – Tarea Nro. 1.....	198
Tabla 195: Enlaces hacia rutas – Tarea Nro. 2.....	199
Tabla 196: Enlaces hacia rutas – Tarea Nro. 3.....	199
Tabla 197: Enlaces hacia rutas – Tarea Nro. 4.....	200
Tabla 198: Tareas de reporte plano de sitios (App móvil).....	200

Tabla 199: Reporte plano de sitios (App móvil) – Tarea Nro. 1	201
Tabla 200: Reporte plano de sitios (App móvil) – Tarea Nro. 2.....	201
Tabla 201: Reporte plano de sitios (App móvil) – Tarea Nro. 3.....	202
Tabla 202: Reporte plano de sitios (App móvil) – Tarea Nro. 4.....	202
Tabla 203: Tareas de reporte de historias y videos (App móvil).....	203
Tabla 204: Reporte de historias y videos (App móvil) – Tarea Nro. 1	203
Tabla 205: Reporte de historias y videos (App móvil) – Tarea Nro. 2	204
Tabla 206: Reporte de historias y videos (App móvil) – Tarea Nro. 3	204
Tabla 206: Tareas de reporte de festivos, fotos y gastronomías (App móvil).....	205
Tabla 207: Reporte de festivos, fotos y gastronomías (App móvil) – Tarea Nro. 1	205
Tabla 208: Reporte de festivos, fotos y gastronomías (App móvil) – Tarea Nro. 2	206
Tabla 209: Reporte de festivos, fotos y gastronomías (App móvil) – Tarea Nro. 3	206
Tabla 210: Iteración III, Historial de revisiones Nro. 1	207
Tabla 211: Iteración III, Historial de revisiones Nro. 2	208
Tabla 212: Iteración III, Historial de revisiones Nro. 3	209
Tabla 213: Iteración III, Historial de revisiones Nro. 4	210
Tabla 214: Iteración III, Historial de revisiones Nro. 5	211
Tabla 215: Iteración III, Historial de revisiones Nro. 6	212
Tabla 216: Iteración III, Historial de revisiones Nro. 7	212
Tabla 217: Iteración III, Historial de revisiones Nro. 8	213
Tabla 218: Iteración III, Historial de revisiones Nro. 9	214
Tabla 219: Iteración III, Historial de revisiones Nro. 10	215
Tabla 220: Iteración III, Historial de revisiones Nro. 11	216
Tabla 221: Tabla de costos	230
Tabla 222: Tabla de cotización.....	230
Tabla 223: Tabla de valores	231
Tabla 224: Tabla de ingresos	231

ANTEPROYECTO APROBADO POR EL CONSEJO ACADÉMICO

UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

ANTEPROYECTO DE TRABAJO DE GRADO

DATOS PERSONALES:

Tema: Aplicación distribuida web-móvil administrable para la gestión y difusión geo-localizada de atractivos turísticos y hoteles para la ciudad de Ibarra, con tecnología GIS y Software Libre.	
ÁREA / LÍNEA DE INVESTIGACIÓN: Innovación Tecnológica y en Productos	
ENTIDAD QUE AUSPICIA:	
DIRECTOR: Ing. Irving Reascos.	
AUTOR: Morales de la Torre Jairo Alberto.	
DIRECCIÓN: San Roque Calle Modesto Larrea y Línea Férrea Esquina.	
TELÉFONO: 062 906 606	
CORREO ELECTRÓNICO: dmxjairo@hotmail.com	
DURACIÓN (Estimada): 7 meses	
INVESTIGACIÓN: Nueva() Continuation(x)	
PRESUPUESTO (estimado): 1450 \$	
PARA USO DEL CONSEJO	
FECHA DE ENTREGA:	FECHA DE REVISIÓN:
APROBADO: SI () NO ()	FECHA DE APROBACIÓN:
OBSERVACIONES:	

UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
PLAN DEL PROYECTO DE TITULACIÓN

Propuesto por: Morales de la Torre Jairo Alberto	Áreas Técnicas del tema: Innovación Tecnológica y en Productos
Director sugerido: Ing. Irving Reascos.	Fecha: 20-Diciembre-2012

1. TEMA

APLICACIÓN DISTRIBUIDA WEB-MÓVIL ADMINISTRABLE PARA LA GESTIÓN Y DIFUSIÓN GEO-LOCALIZADA DE ATRACTIVOS TURÍSTICOS Y HOTELES DE LA CIUDAD DE IBARRA, CON TECNOLOGÍA GIS Y SOFTWARE-LIBRE.

1.1. PLANTEAMIENTO DEL PROBLEMA

1.1.1 ANTECEDENTES

Existe un gran número de sitios turísticos en la ciudad de Ibarra que por falta de geo-localización, son desconocidos por los turistas nacionales y extranjeros.

A pesar de que la tecnología móvil se ha desarrollado en el país, no se ha logrado una investigación en el campo de la informática orientada a la construcción de aplicaciones móviles, que incorporen el uso de cartografía básica y temática.

A medida que aumentan las capacidades, prestaciones y posibilidades de los dispositivos y terminales móviles, se está produciendo una demanda cada vez mayor de nuevos servicios y aplicaciones. El sistema operativo *Android* constituye una de las apuestas más importantes del mercado, que proporciona un completo soporte para el desarrollo de

aplicaciones móviles fiables, robustas y seguras y que pueden ser ejecutadas sobre diferentes tipos de dispositivos móviles y tabletas.

1.1.2. SITUACIÓN ACTUAL

Actualmente no existe información adecuada para los turistas locales y extranjeros, que por desconocimiento de la forma cómo llegar a un destino o atractivo turístico, se desmotiva y no puede planificar adecuadamente su permanencia en el destino turístico, hospedaje, alimentación, transporte, entre otros.

Ibarra cuenta con lugares turísticos que son mostrados de una forma convencional, mediante fotografías y texto, mientras que la inclusión de mapas interactivos con información actualizada y veraz hace que el turista tenga más incentivos para visitar el destino turístico.

1.1.3. PROSPECTIVA

La forma comunicacional tradicional de promover sitios turísticos como es la TV, la radio, etc., no permite un desarrollo más tecnológico, abierto a las nuevas tendencias de la información y la comunicación en tiempo real.

La falta de un flujo constante y dinámico de información acerca del turismo en el país, no permite incrementar la economía del sector turístico del país. El uso de este sistema, hará que las personas tengan una mejor perspectiva de dónde, cuándo y cómo llegar a estos sitios a la vez que podrán planificar un presupuesto adecuado.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL.

Implementar la aplicación distribuida web-móvil administrable para la gestión y difusión geo-localizada de atractivos turísticos y hoteles de la ciudad de Ibarra, con tecnología GIS y Software Libre.

1.2.2. OBJETIVOS ESPECÍFICOS

1.2.2.1. Recopilar información sobre destinos y atractivos turísticos en la ciudad de Ibarra.

1.2.2.2. Realizar el Estudio de servicios Web geográficos (Geo-servicios) y estándares que definirán las reglas de transmisión de información geográfica.

1.2.2.3. Desarrollar y mejorar la administración web, mediante el uso de tecnología AJAX.

1.2.2.4. Diseñar la arquitectura de la aplicación móvil para *Android*.

1.2.3.5. Elaborar la documentación técnica y de usuario del sistema, utilizando la metodología de desarrollo ágil XP.

1.3. ALCANCE DEL PROYECTO

1.3.1. ALCANCE

El presente proyecto está destinado a contribuir al desarrollo de tecnología nueva en el campo de la difusión de la información turística, es así que se plantea llevarlo a cabo mediante la implementación de módulos para la aplicación web-móvil, los mismos que se detallan a continuación y que son presentados gráficamente en la figura 1.3.1.3

1.3.1.1 Sistema Web

a) Módulo de Autenticación

El sistema web de Turismo se integrara dentro del modelo de autenticación de la plataforma.

De esta forma, los usuarios que necesiten usar la aplicación web deberán autenticarse en el sistema utilizando un usuario que disponga de suficientes permisos para poder realizar las operaciones deseadas.

La aplicación brindara acceso a una pantalla inicial (formulario) donde el usuario deberá consignar el login/password para acceder al aplicativo.

b) Módulo Carga de información

Este módulo será el encargado de obtener la información que previamente sea generada desde la aplicación web (previa generación desde el Editor de cartografía de la información) y que constituirá la base de trabajo para los técnicos de campo de los sitios

a agregarse. Este módulo recibirá toda la información de mapas disponibles e información alfanumérica asociada y la almacenará en la base de datos del VPS en línea, para su posterior utilización.

c) Módulo Administración de formularios

Este módulo se encargara de enviar la información generada por los usuarios registrados en el sistema. El modulo enviara los datos en un formato JSON hacia el servidor.

Este módulo brindara una interfaz gráfica que permitirá a los usuarios acceder a través de los menús de la aplicación hacia varias pantallas y permitirá visualizar todos los datos que han sido generados.

d) Módulo Visualización alfanumérica

Este módulo es el encargado de visualizar la información en forma de reportes relativa a la entidad solicitada por el usuario (por ejemplo un atractivo turístico o un hotel).

En la visualización de la información el sistema utilizará los valores definidos por el operador de la aplicación, además el usuario podrá exportar la información hacia formato PDF.

1.3.1.2 Aplicación móvil Android

a) Módulo de Búsquedas

Se refiere a obtener la información que previamente ha sido generada desde la aplicación web además, los datos alfanumérica que se podrá buscar será referente a atractivos turísticos y hoteles.

b) Módulo de Visualización de la información cartográfica

La aplicación móvil dispone de un módulo genérico dentro del aplicativo de movilidad para poder visualizar la información cartográfica almacenada en la base de datos. Este módulo permite:

- Visualización de ficheros cartográficos almacenados previamente la base de datos.
- Zoom Más y Menos.
- Punto de información. Visualización de la información asociada al punto seleccionado por el usuario.

Esta modulo proporcionará un acceso para poder visualizar información de forma online para evitar los problemas de almacenamiento de los dispositivos móviles.

ESQUEMA GENERAL DEL SISTEMA

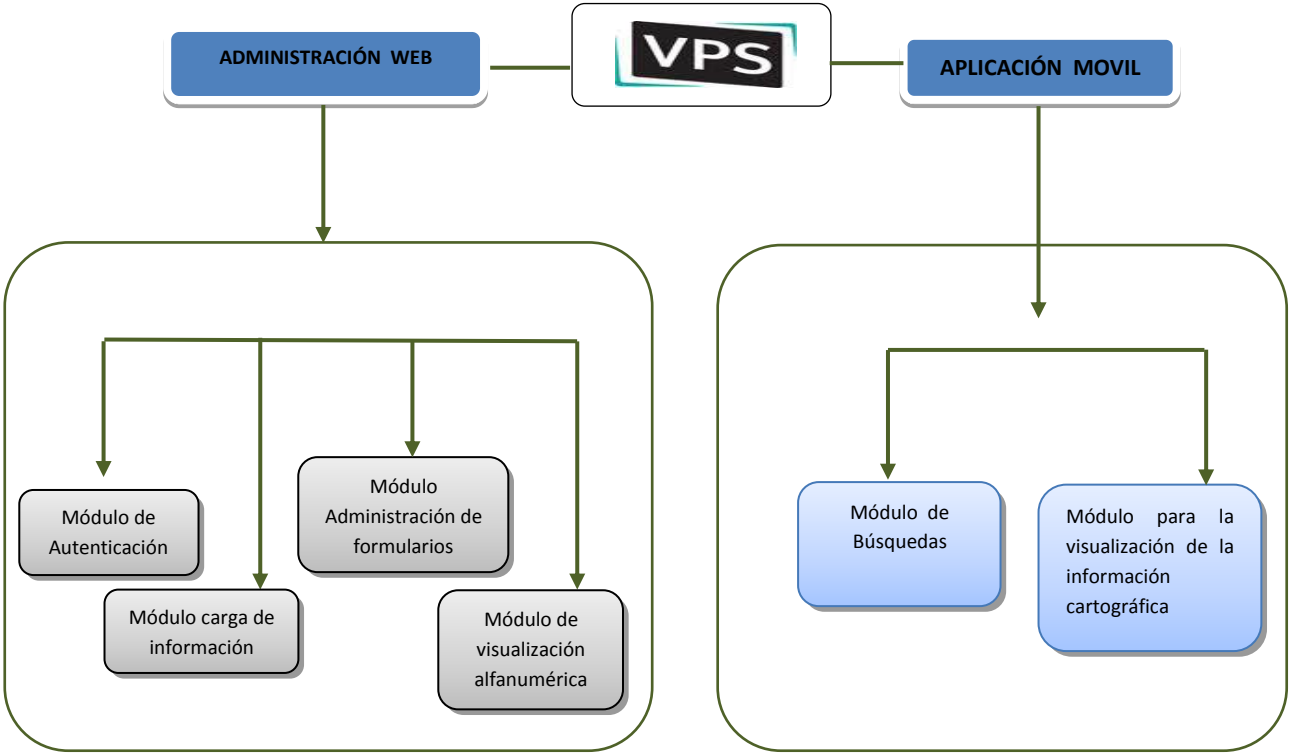


Figura a: Esquema general del sistema Anteproyecto

Fuente: Propia

1.3.1.4 ARQUITECTURA FUNCIONAL DEL SISTEMA

A continuación se describe la aplicación web-móvil, misma que está basada en dos componentes, una administración web para gestionar la información espacial y alfanumérica relevante de sitios turísticos y hoteles. El sistema utilizará *PostgreSQL* para trabajar con datos geográficos como también de un servidor de mapas *mapnick* de *Open Street maps*.

La aplicación móvil se publicara, y estará disponible en android market, Además tendrán acceso hacia la información del servidor privado virtual, como también a los módulos establecidos en la aplicación móvil **figura a**).

A continuación se detalla gráficamente la arquitectura funcional del sistema, véase la siguiente figura.

ARQUITECTURA FUNCIONAL DEL SISTEMA

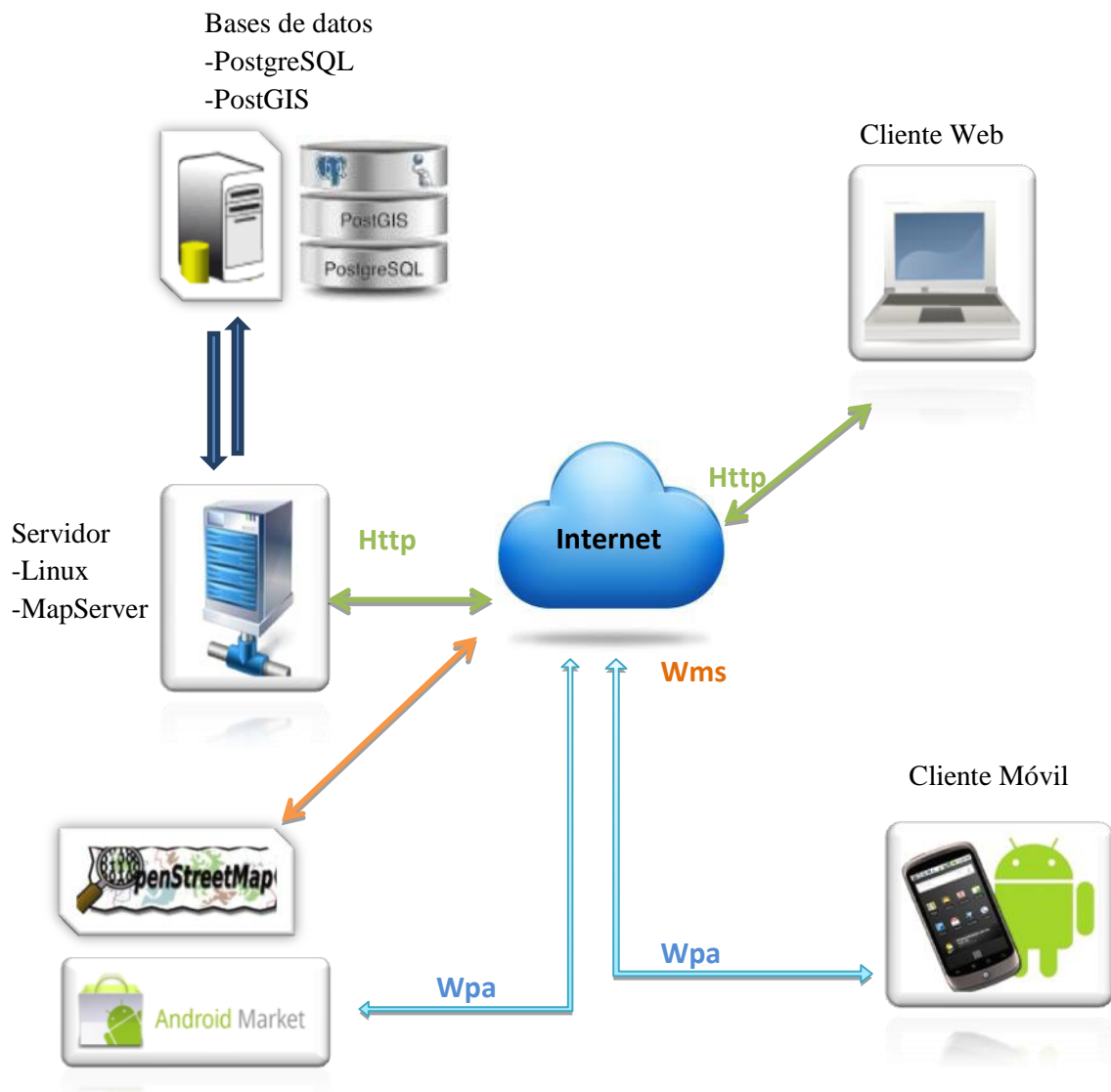


Figura b: Arquitectura funcional del sistema Anteproyecto

Fuente: Propia

1.4. JUSTIFICACIÓN

1.4.1. JUSTIFICACIÓN TEÓRICA DEL PROYECTO

Para la mayoría de los desarrolladores, la interfaz de administración es un tema delicado, pues la información se la inserta de manera estática; este método convierte a la aplicación en monótona y difícil de administrar, pues la misma crece sin ningún control y dominio por parte de los usuarios.

Para que exista un sitio que ofrezca alternativas turísticas, la aplicación debe cumplir ciertos estándares de calidad, pues la computación móvil cumple una de esas expectativas, ya que el usuario tiene su aplicación prácticamente en su bolsillo, y de esta manera se abre un abanico de posibilidades para buscar información veraz y exacta de algún sitio a donde se desee llegar.

Los logros a obtener mediante la aplicación funcional del sistema web-móvil, son los siguientes:

- Agilizar el proceso de publicación de la información geográfica.
- Llevar un control y orden de los datos de los atractivos turísticos y hoteles.
- Proveer agilidad y seguridad en la información de sitios específicos.
- Mayor control por parte de los usuarios sobre su información publicada.
- Disponer de una aplicación móvil para la promoción y difusión de sitios turísticos.

1.4.2. JUSTIFICACIÓN TECNOLÓGICA DEL PROYECTO

1.4.2.1 HERRAMIENTAS DE DESARROLLO

Para la elaboración del presente tema; “Aplicación distribuida web-móvil administrable para la gestión y difusión geo-localizada de atractivos turísticos y hoteles de la ciudad de Ibarra, con tecnologías GIS y Software-Libre” se utilizará PostgreSQL como gestor de base de datos por ser potente y altamente eficiente para trabajar con datos espaciales, puede ejecutarse en distintas plataformas, posee un lenguaje de base de datos muy potente como lo es (Lenguaje procedural) que permite implementar funciones con una

integridad referencial bastante potente. Considerada como una de las bases de datos espaciales más potentes.

1. 4.2.1.1 Herramientas de desarrollo web

Java Script (Lenguaje de programación): Es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

Twitter Bootstrap (Conjunto de librerías): se trata de un conjunto de recursos (estilos, tipográfico, elementos de formularios, botones, tablas, barras de navegación, mas) que podemos utilizarlos en el diseño de administradores, logrando una mejora muy profesional y economizando tiempo de diseño y maquetado. Es esencialmente gratuito.

Ajax (Tecnología): Acrónimo de *asynchronous javascript and XML (javascript asíncrono y XML)*, es una técnica de desarrollo web para crear aplicaciones interactivas o rias (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.

PHP (Lenguaje de Programación): 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.

OSM (Open Street Maps): Es un servicio de mapas y geo-localización de uso libre, que contiene datos espaciales de cualquier región del mundo, parte de la idea de que cada habitante de este planeta puede mapear la región donde vive o por las que transita (ya que puede hacerlo de forma más precisa y detallada) y por lo tanto se brinda una infraestructura tecnológica para que se puedan trazar fronteras, calles, manzanas, edificios, entre otros elementos que lo rodean.

Openlayers (Conjunto de librerías): Puede mostrar fragmentos del mapa y marcadores cargados de cualquier fuente. Openlayers se ha desarrollado para promover el uso de la información geográfica de todo tipo.

Adodb5_(Conjunto de librerías): Diseñado para la velocidad, Es probablemente la más rápida biblioteca para base de datos de código abierto, disponible para PHP. Utiliza muchas convenciones de *ado*, una de ellas es el amplio control de calidad.

PostgreSQL (Motor de base de datos): Es un sistema de gestión de bases de datos, fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido, más tarde en otros sistemas de gestión comerciales. Incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

PostGIS (Extensión espacial para PostgreSQL): Biblioteca código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos; en los navegadores web.

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

Mapnik (Servidor de mapas): Es un servicio gratuito de herramientas para acceder a mapas, además ofrece AGG anti-aliasing renderizado con precisión subpixel. Puede leer las formas de ESRI, PostGIS, TIFF rasters.

Apache: Es un software que traduce archivos de páginas web y las transforma en HTML y las sirve mediante un protocolo http.

1.4.2.1.2 Herramientas Android

SDK (Software Development Kit): Kit de desarrollo necesario para programar e implementar todo tipo de aplicaciones Android (sistema operativo), para teléfonos móviles, Este paquete o kit de desarrollo incluye las API's y herramientas necesarias para desarrollar las aplicaciones.

Android 2.3 (Superior) Nivel de API 9: Incorpora un nuevo interfaz de usuario con un diseño actualizado. Además introduce un nuevo recolector de basura que minimiza las pausas de la aplicación, ayudando a garantizar un mejor tiempo de respuesta.

Dalvik: es una máquina virtual intérprete que ejecuta archivos en el formato Dalvik Ejecutable (*.dex), un formato optimizado para el almacenamiento eficiente y ejecución mapeable en memoria.

ADV (Administración de dispositivos virtuales): Es una configuración de emulador que permite modelar un dispositivo real mediante la definición de opciones de hardware y software para ser emulados por el Emulador de Android.

OSMDROID: Open Street maps para teléfonos móviles, permiten la integración de mapas en una terminal móvil. Con ciertas ventajas sobre Google Maps

Java: Es un lenguaje de programación interpretado de alto nivel orientado a objetos.

1.4.3 JUSTIFICACIÓN METODOLÓGICA DEL PROYECTO

1.4.3.1 METODOLOGÍA DE DESARROLLO XP

XP (Programación Extrema) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

Las características fundamentales de ésta metodología son:

- Desarrollo iterativo e incremental: pequeñas mejoras, (unas tras otras).
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

CAPÍTULO I

1 Metodología y Herramientas de Desarrollo

1.1 Metodología XP

XP describe un conjunto de prácticas para un desarrollo óptimo, ya que define con exactitud los requerimientos del usuario. Esta metodología difiere de las demás por que se basa en la adaptabilidad y la previsión, pues propone que, cambiar los requerimientos del sistema durante el desarrollo constituye un mejor acercamiento con el usuario; surge como una solución a los problemas derivados del cambio constante en los requerimientos de un sistema. (Aplicable, 2008).

1.1.1 Roles XP

XP define una serie de roles que son los siguientes:






Programador	Cliente	Encargado de Pruebas	(Traker)	Entrenador (Coach)
				

Tabla 1: Roles de la Metodología XP

Fuente: (Aplicable, 2008)

Programador

- Pieza básica en desarrollos XP
- Más responsabilidad que en otros modos de desarrollo
- Responsable sobre la generación del código fuente.
- Responsable sobre el diseño y maquetado de la aplicación.
- Responsable de administrar la bases de Datos.
- Responsable sobre la integridad del sistema (pruebas)
- Acepta críticas (código colectivo)

Cliente

- Pieza básica en desarrollos XP
- Define especificaciones
- Influye sin controlar
- Confía en el grupo de desarrollo
- Define pruebas funcionales

Encargado de Pruebas

- Apoya al cliente en la preparación/realización de las pruebas funcionales
- Ejecuta las pruebas funcionales y publica los resultados

Tracker (Encargado de Seguimiento)

- Recoge, analiza y publica información sobre la marcha del proyecto sin afectar demasiado el proceso.
- Supervisa el cumplimiento de las estimaciones en cada iteración.
- Informa sobre la marcha de la iteración en curso.
- Controla la marcha de las pruebas funcionales, de los errores reportados, de las
- Responsabilidades aceptadas y de la prueba añadida por los errores.

Entrenador (Coach)

- Experto en Metodología XP
- Responsable del proceso en su conjunto
- Identifica las desviaciones y reclama atención sobre las mismas
- Guía al grupo de forma indirecta (sin dañar su seguridad ni confianza)
- Interviene directamente si es necesario
- Atajar rápidamente el problema

Integrantes XP

Para el desarrollo de la siguiente aplicación, él estudiante juega un papel muy importante pues desempeña todas los roles XP, en vista de que el proyecto de tesis es individual, a continuación se describe los roles.

Nombre	Descripción	Rol XP
Jairo Morales	Quien desarrolla en sistema.	Programador
Sitios Turísticos	Quienes solicitan los requerimientos.	Cliente
Jairo Morales	Ejecuta el aplicativo, en busca de errores.	Encargado de Pruebas
Ing. Irving Reascos.	Controla que las actividades desarrollen en coordinación.	<i>Tracker</i> (Encargado del seguimiento)
Ing. Irving Reascos.	Experto en metodología <i>XP</i> .	Coach

Tabla 2: Integrantes en un Proyecto XP

Fuente: Propia

1.1.2 Valores XP

Mejora un proyecto de software inicialmente en sus cuatro valores esenciales, descritos a continuación, un valor fue adicionado "Respeto" en la segunda edición del libro XP, autoría de Kent Beck quien desarrollo la metodología. (ONess, 2005).

- Comunicación

Todos son parte del equipo y la comunicación es esencial, desde los requerimientos hasta la programación.

En los métodos tradicionales de desarrollo de software, la comunicación de los requerimientos a los desarrolladores se realiza a través de la documentación, por ejemplo las Especificaciones de Diseño en el Rational Unified Process (RUP). XP rompe con este esquema, la comunicación se realiza por medio de transferencia de conocimientos en reuniones frecuentes cara a cara entre usuarios y desarrolladores, lo que le da a ambos una visión compartida del sistema.

- Simplicidad

Nos dirigiremos a nuestro objetivo a pasos simples y pequeños, mitigando las fallas a medida que ocurran. Crearemos algo de lo cual podamos sentirnos orgullos y que pueda mantenerse en el largo plazo a costos razonables.

Asimismo, un diseño y programación simple mejora la calidad de las comunicaciones, pues es más fácil de implementar y entender por todos en el equipo.

- Retroalimentación (Feedback).

Nos tomaremos seriamente los compromisos con el usuario establecidos en todas las iteraciones, entregando software en funcionamiento en cada una. Mostraremos al usuario nuestro software frecuentemente y de forma temprana, escuchando cuidadosamente sus observaciones y realizando los cambios que sean necesarios. Adaptaremos nuestros procesos al proyecto y no al contrario”.

Retroalimentación del sistema: Por medio de la ejecución de pruebas unitarias y de integración, los programadores reciben retroalimentación directa del estado del sistema.

Retroalimentación del cliente (usuario): Las pruebas de aceptación, son diseñadas conjuntamente por el cliente y los analistas de pruebas, obteniendo en conjunto retroalimentación del estado actual del sistema. Esta revisión puede hacerse cada 1 o 2 semanas, permitiendo así que el cliente sea quien guíe el desarrollo del software.

Retroalimentación del equipo: Cuando el cliente trae nuevos requerimientos, el equipo puede directamente proporcionar la estimación del tiempo que tomará implementarlos.

- Coraje

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave.

- Respeto

El valor del respeto en XP establece: “Todos en el equipo dan y reciben el respeto que merecen como integrantes del equipo y los aportes de cada integrante, son valorados por todos. Todos contribuyen, así sea simplemente con entusiasmo. Los desarrolladores respetan la experticia de los clientes y viceversa. La Gerencia respeta el derecho del equipo de asumir responsabilidad y tener autoridad sobre su trabajo”.

Respeto es tanto por el trabajo de los demás como por el trabajo de uno mismo, por ejemplo, los desarrolladores nunca deben subir cambios que impidan la compilación de la versión, que hagan fallar pruebas unitarias ya realizadas o que de alguna otra forma retrasen el trabajo de sus pares, esto significa tener respeto por el trabajo (y el tiempo) de los demás.

Asimismo, los desarrolladores respetan su propio trabajo por medio de su compromiso con una alta calidad y buscando el mejor diseño para la solución por medio de la refactorización constante.

En cuanto al trabajo en equipo, nadie debe sentirse poco apreciado o ignorado, todos deben colaborar en esto, tratando con respeto a sus compañeros y mostrando respeto por sus opiniones, esto asegura altos niveles de motivación y lealtad hacia el proyecto.

1.1.3 Ventajas y Desventajas de la metodología XP

Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ Programación organizada. ▪ Menor tasa de errores. ▪ Satisfacción del programador. ▪ Solución de errores de programas ▪ Versiones nuevas. ▪ Implementa una forma de trabajo donde se adapte fácilmente a las Circunstancias. 	<ul style="list-style-type: none"> ▪ Es recomendable emplearlo solo en proyectos a corto plazo ▪ Altas comisiones en caso de fallar ▪ Imposible prever todo antes de programar ▪ Demasiado costoso e innecesario

Tabla 3: Ventajas y desventajas de la metodología XP
Fuente:(Rodríguez, 2011)

1.1.4 Fases de la metodología XP

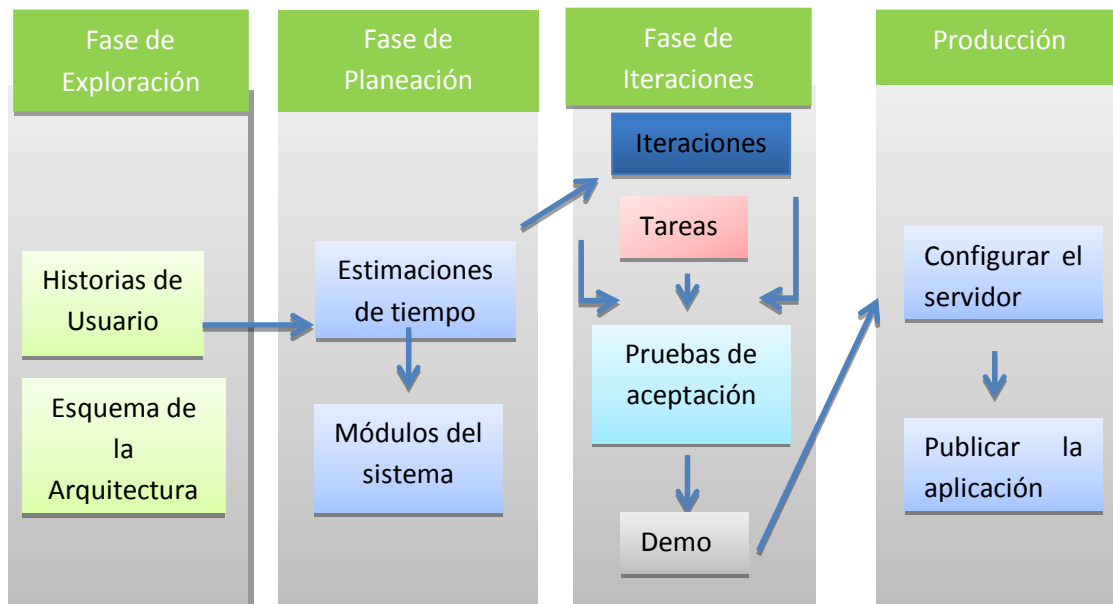


Ilustración 1: Fases de la metodología XP
Fuente:(Attribution, 2013)

- Exploración

Los clientes plantean a grandes rasgos las historias de usuario que son de interés para las entregas del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas de desarrollo que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema, construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

- Planeación

En esta fase el equipo XP establece la prioridad de cada historia de usuario y dividiendo el sistema en módulos, además se diseña un cronograma de actividades para optimizar los tiempos de desarrollo del sistema. Se toman acuerdos sobre el contenido de la primera entrega y se asigna tareas por cada historia de usuario. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días.

- Iteraciones

Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El programador responsable debe cumplir con las tareas asignadas de acuerdo al cronograma de actividades, y no exceder de tres meses en cada iteración o entrega. En la primera iteración se desarrolla las tareas de cada historias de usuario, después de realizar las pruebas de aceptación y la demos del sistema al final de cada iteración el sistema estará listo para presentarlo al usuario final.

- Producción

La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno real. Al mismo tiempo se toma decisiones sobre las características del servidor de aplicaciones a implementar.

1.2 AJAX

Es una técnica de desarrollo web para crear aplicaciones interactivas, las cuales se ejecutan en el cliente, y mantiene una comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página web sin necesidad de recargarla. Esto significa un aumento en la interactividad, velocidad y usabilidad en la misma.

“AJAX no es una tecnología, En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes“. [HTTP 3].

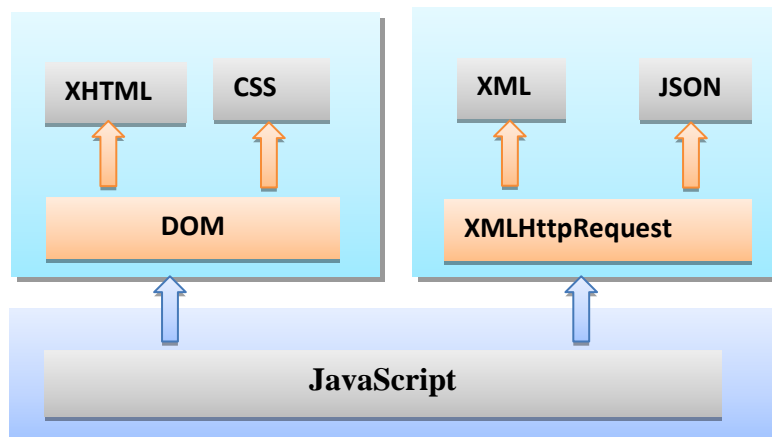


Ilustración 2: Esquema de tecnologías AJAX

Fuente:(librosweb, 2013)

1.2.1 JAVASCRIPT

Es un lenguaje de programación basado en objetos con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes.

Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos y escribir funciones que respondan al movimientos del mouse, aperturas, utilización de teclas, cargas de página, utilizando DOM² dinámico.

Técnicamente, es un lenguaje de programación interpretado, porque no es necesario compilar los programas para ejecutarlos, ya que sus fuentes se ejecutan en todos los navegadores sin necesidad de procesos intermedios (Librosweb, 2013).

- DOM.

Modelo de Objeto Documento, para la interacción y manipulación dinámica de la presentación, que surgió de la necesidad de procesar manipular el contenido de los archivos XML.

También Se utiliza para crear y diseñar especificaciones y etiquetas HTML de una manera más rápida y eficiente (JOE, JEREMY, & C, 2006)

² Document Object Model: Permite crear componentes *HTML* de una manera dinámica.

- XHTML

Lenguaje de marcado de hipertexto extensible, similar a HTML, con características que lo hacen más robusto, básicamente es HTML expresado como XML valido.

✚ Diferencias entre los estándares HTML y XHTML

La sintaxis XHTML es más estricta, los cambios que se deben hacer a documento HTML para convertirlo en XHTML son las siguientes: [LIBRO 1].

Característica	HTML	XHTML
Todas las etiquetas deben ser cerradas	<code>
</code>	<code>
</code>
Atributos debe ir entre comillas	<code><table width=75% bgcolor=ff0000></code>	<code>< table width="75" bgcolor="ff0000"></code>
Utiliza atributo ID en vez de NAME	<code></code>	<code></code>
No se puede usar Sintaxis abreviada.	<code><input checked></code>	<code><input checked="checked"></code>
Usar sentencia DOCTYPE para indicarle al navegador como Procesar el documento.		a) Documentos estrictos <code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>
Los nombres de elementos y atributos deben ir en minúsculas.	<code>Domname</code>	<code>Domname</code>

Tabla 4: Diferencias entre los estándares HTML y XHTML

Fuente: (JOE, JEREMY, & C, 2006)

- **CSS**

CSS³ es una tecnología desarrollada por la W3C con el fin de separar la estructura de la presentación.

- **XMLHttpRequest**

Es una **API** que puede invocarse desde *JAVASCRIPT*, y otros lenguajes de script incluidos en un navegador web. Su objetivo es permitir a *JAVASCRIPT* formular peticiones **HTTP** y enviarlas al servidor, de modo tradicional en las aplicaciones web, estas peticiones se hacen de modo sincrónico junto con un evento que inicia el usuario, que puede ser por ejemplo hacer *click* en un enlace o enviar un formulario, el resultado de esto es una página nueva o la actualización de la página que es servida desde el navegador.(JOE, JEREMY, & C, 2006)

- **Crear un objeto XMLHttpRequest**

Todos los navegadores modernos (*IE7 +, Firefox, Chrome, Safari y Opera*) tienen un objeto incorporado *XMLHttpRequest*.

- **Sintaxis para crear un objeto XMLHttpRequest:(HOLZER, 2009).**

```
var httpRequest;
if(window.XMLHttpRequest)
{
    //El explorador implementa la interfaz de forma nativa
    httpRequest=new XMLHttpRequest();
}
elseif(window.ActiveXObject)
{
    //El explorador permite crear objetos ActiveX
    try{
        httpRequest=new ActiveXObject("MSXML2.XMLHTTP");
    }
    catch(e)
    {
        try{
            httpRequest=new ActiveXObject("Microsoft.XMLHTTP");
        }catch(e){}
    }
}
if(!httpRequest)
{
    alert("No ha sido posible crear una instancia de XMLHttpRequest");
}
```

³ Cascading Style Sheet: Es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla.

✚ Métodos y propiedades del objeto XMLHttpRequest

Las propiedades definidas para el objeto XMLHttpRequest son:

Propiedad	Descripción
readyState	Valor numérico (entero) que almacena el estado de la petición
responseText	El contenido de la respuesta del servidor en forma de cadena de texto
responseXML	El contenido de la respuesta del servidor en formato XML. El objeto devuelto se puede procesar como un objeto DOM
status	El código de estado HTTP devuelto por el servidor (200 para una respuesta correcta, 404 para "No encontrado", 500 para un error de servidor, etc.)
statusText	El código de estado HTTP devuelto por el servidor en forma de cadena de texto: "OK", "Not Found", "Internal Server Error", etc.

Tabla 5: propiedades elementales del objeto XMLHttpRequest

Fuente: (JOE, JEREMY, & C, 2006)

✚ Los valores definidos para la propiedad readyState

Valor	Descripción
0	No inicializado (objeto creado, pero no se ha invocado el método open)
1	Cargando (objeto creado, pero no se ha invocado el método send)
2	Cargado (se ha invocado el método send, pero el servidor aún no ha respondido)
3	Interactivo (se han recibido algunos datos, aunque no se puede emplear la propiedad responseText)
4	Completo (se han recibido todos los datos de la respuesta del servidor)

Tabla 6: Representación de los valores de respuesta readyState

Fuente: (JOE, JEREMY, & C, 2006)

🚩 Los métodos disponibles para el objeto XMLHttpRequest

Método	Descripción
abort()	Detiene la petición actual
getAllResponseHeaders()	Devuelve una cadena de texto con todas las cabeceras de la respuesta del servidor
getResponseHeader("cabecera")	Devuelve una cadena de texto con el contenido de la cabecera solicitada
onreadystatechange	Responsable de manejar los eventos que se producen. Se invoca cada vez que se produce un cambio en el estado de la petición HTTP. Normalmente es una referencia a una función JavaScript
open("Método", "URL")	Establece los parámetros de la petición que se realiza al servidor. Los parámetros necesarios son el método HTTP empleado y la URL destino (puede indicarse de forma absoluta o relativa)
send(contenido)	Realiza la petición HTTP al servidor

Tabla 7: Métodos definidos por AJAX para la transferencia de información

Fuente:(JOE, JEREMY, & C, 2006)

- XML

XML (Extendible Markup Lenguaje), diseñado para transportar y almacenar datos de codificación de información fácilmente entre varias plataformas que utilicen una misma lógica interna.

La información que contiene los documentos XML son archivos de texto organizados de forma de árbol: cada rama puede tener unos atributos propios y servir de base para otras ramas.(JOE, JEREMY, & C, 2006).

✚ Datos de libros almacenados en un XML. (JOE, JEREMY, & C, 2006)

```
<?xml version="1.0" encoding=" utf-8"?>
<bookList>
  <book isbn="047177781">
    <title>Professional Ajax</title>
    <author>Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett</author>
    <publisher>Wrox</publisher>
  <book isbn="1861002025">
    <title>Professional Visual Basic 6 Databases</title>
    <author>Charles Williams</author>
    <publisher>Wrox</publisher>
  </book>
</bookList>
```

- JSON

JSON es un formato de datos muy ligero basado en un subconjunto de la sintaxis de JAVASCRIPT, que utiliza literales de matriz y objeto para incluirlas dentro de los atributos de JAVASCRIPT, y que pueden accederse sin el análisis gramatical de los archivos XML.(JOE, JEREMY, & C, 2006)

✚ Literales de matriz

Son especificaciones usadas para adjuntar información hacia una estructura de datos tipo matriz.

Se puede definir un valor

```
var nombreEst= ["Benjamin", "Michael", "Ana"];
Modo de acceso
nombreEst [0];           //retorna
"Benjamin"
```

✚ Literales de Objeto

Se utiliza para guardar la información en los pares del nombre-valor, cada par debe seguirse de una coma, salvo el último campo, por ejemplo.

```

var nombreEst = {
    "Nombre": Ana,
    "Edad"      :
    24,
    "Estado"    :
    true
};

```

Literales mixtos

Es posible mezclar literales de matriz y objetos, en el caso de tener una estructura de dato compleja.

```

var arregloCarros = [
    {
        "ID" : "Cod1",
        "puertas" : 3
    },
    {
        "ID" : "Cod2",
        "puertas" : 4
    },
    {
        "ID" : "Cod3",
        "puertas" : 6
    }
];

```

1.2.2 Fases en la ejecución de una aplicación AJAX

El objeto Ajax tiene una secuencia de pasos que debes ejecutarse, para establecer una buena comunicación entre el cliente y el servidor. (SIERRA, 2012).

- Creación y configuración del objeto XMLHttpRequest.

El primer paso a realizar cuando se produce el evento que desencadena la ejecución de una aplicación AJAX consiste en crear el objeto XMLHttpRequest. El cual establecerá una conexión con el servidor.

Una vez creado el objeto, deben configurarse una serie de parámetros del mismo, como la URL del recurso a solicitar o la función que va a procesar la respuesta.

- Realización de la petición.

Tras configurar los parámetros adecuados del objeto XMLHttpRequest, se procede a lanzar la petición al servidor, esta operación puede realizarse en modo síncrono o asíncrono, siendo este último el modo de funcionamiento mayoritariamente utilizado por las aplicaciones AJAX.

- **Procesamiento de la petición en el servidor.**

El servidor recibe la petición y ejecuta el componente correspondiente que, a partir de los datos recibidos, deberá realizar algún tipo de procesamiento, incluyendo consultas a bases de datos, y generar una respuesta con los resultados obtenidos.

Este componente es un archivo de PHP o cualquier otro tipo de aplicación que se pueda ejecutar en un servidor Web.

Cuando la petición se realiza en modo asíncrono, mientras el componente del servidor se está ejecutando para realizar su función, el usuario puede seguir interactuando con la página sin necesidad de mantenerse bloqueado a la espera de recibir la respuesta.

- **Recepción de los datos de respuesta.**

Una vez completada la ejecución del código de servidor, se envía una respuesta HTTP al cliente con los resultados obtenidos en el formato adecuado para su manipulación. En ese momento, el navegador invoca a la función de retro-llamada definida por el objeto XMLHttpRequest.

- **Manipulación de la página cliente.**

A partir de los datos recibidos en la respuesta y mediante código JavaScript de cliente, se modifican las distintas zonas de la página XHTML que sea necesario actualizar.

1.2.3 Arquitectura AJAX

AJAX tiene la comodidad de instanciar una conexión con el servidor en cualquier evento, al presionar sobre un botón o link. (JOE, JEREMY, & C, 2006).

- Flujo de información

Las llamadas hacia el servidor del modelo tradicional se efectúan una sola vez, en el nuevo modelo el objeto XMLHttpRequest es quien gestiona las peticiones continuas hacia el servidor.

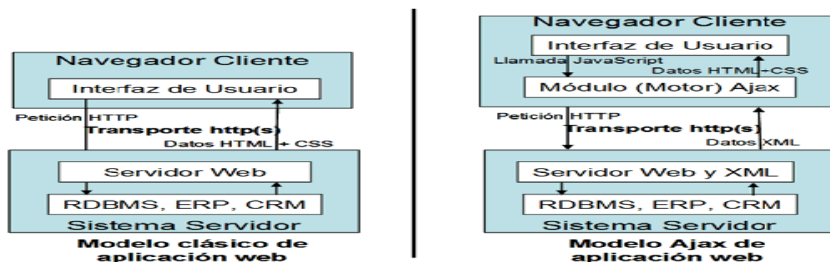


Ilustración 3: Modelo convencional y propuesto por AJAX

Fuente:(Librosweb, Introducción a AJAX, 2003)

- **Transmisión síncrona**

La invocación espera el resultado, mientras tanto no se recibe ninguna respuesta de la aplicación, este modo de espera convierte a las páginas web en poco interactivas.

- **Transmisión asíncrona**

Realiza la invocación, pero continua con la ejecución, la invocación continúa y por lo general se define un callback (o sea algún método o función) que recibirá la respuesta o el error para tomar alguna acción

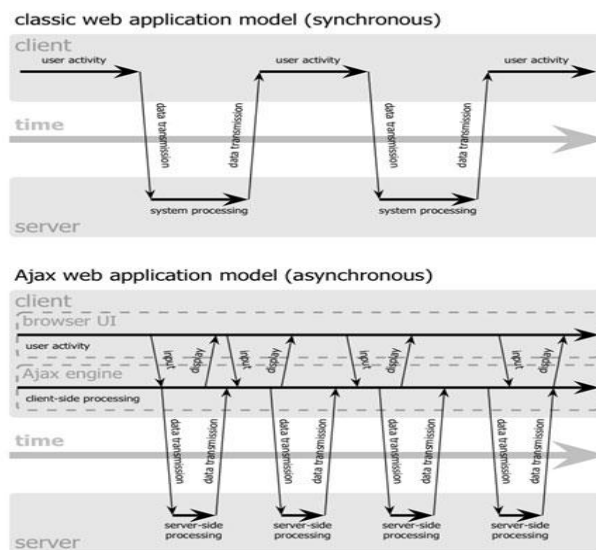


Ilustración 4: Transmisión síncrona y asíncrona
Fuente: (JOE, JEREMY, & C, 2006)

Diferencias entre una asíncrona y síncrona

Las aplicaciones web son sincrónicas, pues la presentación que se les entrega a los usuarios representa una "foto" de un momento específico de un sistema dinámico. Esta foto se vuelve obsoleta entre las interacciones del usuario, y no necesariamente brinda una vista exacta del estado actual del sistema. Incluso aunque se utilicen técnicas AJAX, el proceso sincrónico no cambia. Usando el objeto XMLHttpRequest y técnicas AJAX logramos tener un modelo de interacción de grano más fino que refrescar una página completa, pero las peticiones se siguen generando basándonos en interacciones del usuario, por lo que el proceso sigue siendo sincrónico, y persiste el problema de tener vistas potencialmente obsoletas.

La Web Asíncrona es fundamentalmente diferente, y es esta diferencia la que revoluciona el comportamiento de las aplicaciones web. En la web Asíncrona es posible entregar al usuario cambios espontáneos en la presentación, a medida que cambia el estado de un sistema dinámico, sin la necesidad de que el usuario interactúe con la interfaz.

Las ventajas son obvias, ya que se puede mantener una representación exacta del sistema en el navegador del usuario. Hay varios ejemplos, como una interfaz Web que brinda una vista de un sistema dinámico, como ser un portafolio de acciones, un inventario, o un calendario. Cuando hay muchos usuarios interactuando en el mismo sistema, las interacciones de un usuario pueden impactar espontáneamente lo que ven otros usuarios, creando así una aplicación verdaderamente colaborativa, la esencia de la Web 2.0. Hay varios ejemplos, como un cliente de chat simple, o una subasta por eBay. En última instancia, la mayoría de los sistemas con las que interactúan los humanos son colaborativos por naturaleza, por lo que también debería serlo la interfaz que utilizan. (JOE, JEREMY, & C, 2006).

1.2.4 Ventajas y desventajas de utilizar AJAX

Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ El tráfico mínimo: Las aplicaciones de <i>AJAX</i> pueden minimizar la cantidad de tráfico entre el cliente y el servidor. Asegurándose que la aplicación no envíe y reciba información innecesaria. ▪ Accesibilidad: Las peticiones no tienen ningún tipo de prioridad, pues el evento <i>click</i> puede instanciar una conexión en cualquier etiqueta u objeto de la página, de una manera segura y veraz. ▪ Evite las descargas de la página enteras: <i>AJAX</i> evita que se recargue toda la página, cuando se establece una comunicación con el servidor, ya que las llamadas hacia el servidor se las realiza en un punto intermedio entre el cliente y el servidor. ▪ Tiempo: El tiempo de espera para una petición se reduce, pues el servidor nos envía solo la información requerida. 	<ul style="list-style-type: none"> ▪ Botón volver anterior: Se pierde el concepto de volver a la página anterior. ▪ Navegador web: Problemas si el usuario ha deshabilitado el uso de JavaScript en su navegador. Hay que tener esto en cuenta cuando desarrollamos nuestro sitio web, para enfrentarnos a esta situación. ▪ Links: No poder recomendar links específicos. Si hace que toda la web sea interactiva, no podremos recomendar algún link, ya que el contenido fue generando dinámicamente. Se debería encontrar un equilibrio dependiendo de las necesidades de su escenario.

Tabla 8: Ventajas y desventajas de utilizar AJAX

Fuente:(JOE, JEREMY, & C, 2006)

1.3 Base de Datos

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

Desde el punto de vista informático, un DBMS⁴ es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queremos guardar en la tabla, cada fila de la tabla conforma un registro. (KOTH, 2002).

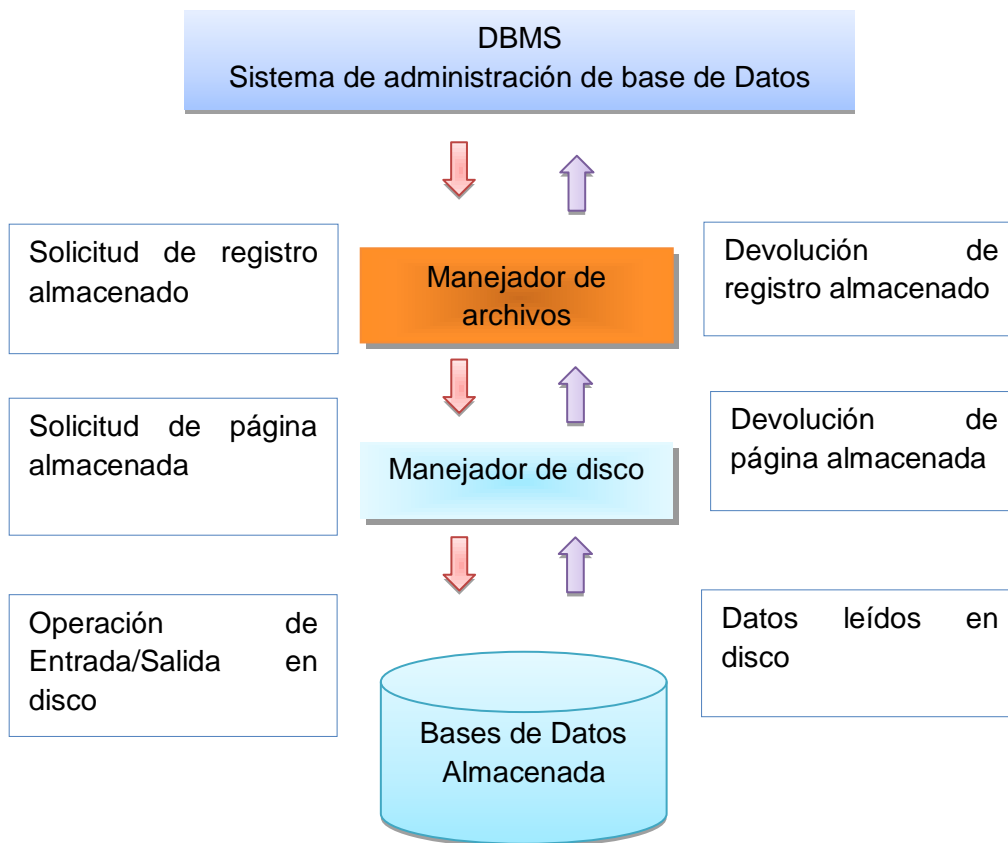


Ilustración 5: Funcionamiento de una Base de Datos

Fuente:(KOTH, 2002)

⁴ Database Management System: Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.

1.3.1 Bases de datos relacionales

Una base de datos relacional es una base de datos que cumple con el modelo relacional, el cual es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos (que están guardados en tablas), Una base de datos relacional se compone de varias tablas o relaciones.

No pueden existir dos tablas con el mismo nombre ni registro. Cada tabla es a su vez un conjunto e registros (filas y columnas).(KOTH, 2002).

La relacione entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y foráneas.

Clave primaria: Es una clave única elegida entre todas las candidatas que define unívocamente a todos los demás atributos de la tabla, para especificar los datos que serán relacionados con las demás tablas.

Clave Foránea: Es una referencia a una clave en otra tabla. Las claves foráneas no necesitan ser claves únicas en la tabla donde están, por ejemplo, el código de una entidad provincia puede ser clave foránea en una entidad ciudad, obviamente se permite que la provincia tenga varias ciudades.

1.3.2 Ventajas y desventajas de utilizar base de datos relacionales

Ventajas	Desventajas
<ul style="list-style-type: none">▪ Elimina información redundante y duplicada que podría ocasionar inconsistencias en los sistemas de bases de datos.▪ Compartir información entre varios usuarios, al emplear la misma entidad.▪ Mantener la seguridad en la información almacenadas.▪ Capacidad para alterar la información sin cambiar por ello la aplicación que accede a la base de datos, y viceversa, lo que supone menor coste de mantenimiento.▪ Restricciones de seguridad, pues nos permite acceder ni efectuar operaciones sobre datos a cualquier usuario.	<ul style="list-style-type: none">▪ Presentan deficiencias con datos gráficos, multimedia, y sistemas de información geográfica.▪ No se manipulan de forma manejable los bloques de texto como tipo de datos.▪ El uso de una base de datos propietaria, tiene costos muy elevados.▪ Algunas bases de datos relacionales tienen límites en cuanto a la longitud de los campos.

Tabla 9: Ventajas y desventajas de utilizar base de datos relacionales.

Fuente: (KOTH, 2002)

1.3.3 PostgreSQL

Es un sistema de gestión de base de datos objeto-relacional, distribuida bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

Utiliza un modelo Cliente/Servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema.

1.3.4 Arquitectura por capas

La programación por capas es una arquitectura cliente/servidor en el que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño; un ejemplo básico de esto consiste en separar la capa de datos de la capa de presentación al usuario.(Kay., 2004).

- **Arquitectura en dos capas**

La aplicación se divide en dos partes, Cliente/Servidor, el cliente llama a la funcionalidad del sistema de bases de datos en la maquina servidor mediante la instrucciones de lenguaje de consultas, gracias a los conectores ODBC y JDBC que hacen posible el acceso hacia la base de datos.

Ejemplo: Aplicación JAVA y Base de datos PostgreSQL (Entorno Local).

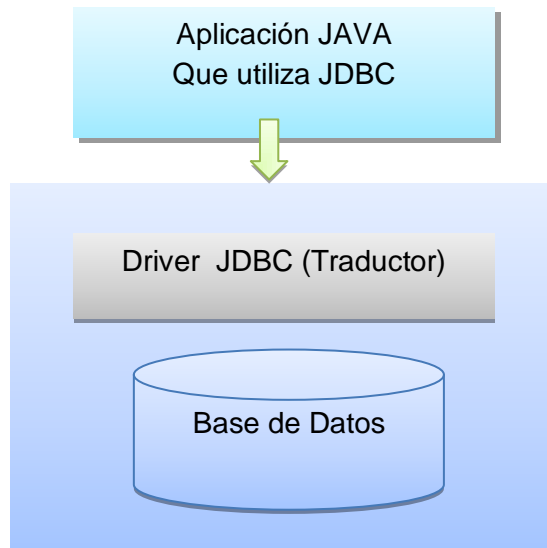


Ilustración 6: Arquitectura en dos capas

Fuente:(Kay., 2004)

- Arquitectura en tres capas

La máquina cliente actúa simplemente como frontal y no contiene ninguna llamada directa a la base de datos. En su lugar, el cliente se comunica con el servidor de aplicaciones, usualmente mediante una interfaz de formularios. El servidor de aplicaciones a su vez, se comunica con el sistema de bases de datos para acceder a los datos.

La lógica de negocio es establecida por el servidor de aplicaciones, que determina las acciones y operaciones a realizar bajo determinadas condiciones.

Ejemplo: Aplicación web (Entorno en producción)

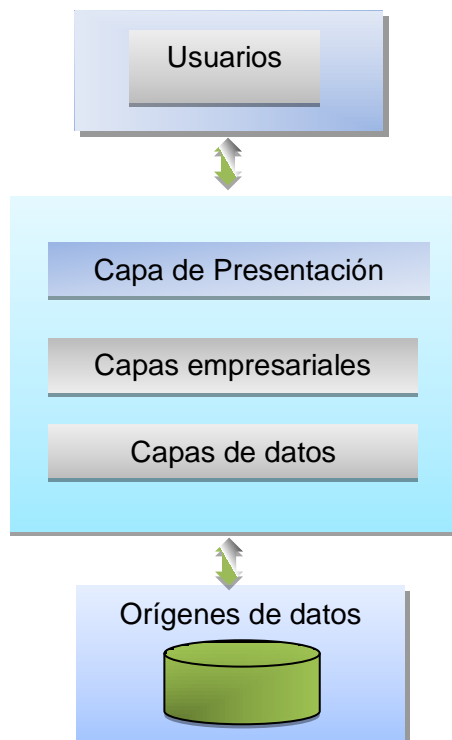


Ilustración 7: Arquitectura en tres capas

Fuente:(Kay., 2004)

1.4 Sistemas de Información Geográfica

Los SIG son una nueva tecnología que nos 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, algunas de ellas acentúan sus componentes en bases de datos relaciones, entre otras funcionalidades enfatizan el hecho de ser una herramienta de apoyo en la toma de decisiones, sin embargo 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, incluyendo la relacionada con la infraestructura turística de un destino o atractivo turístico.

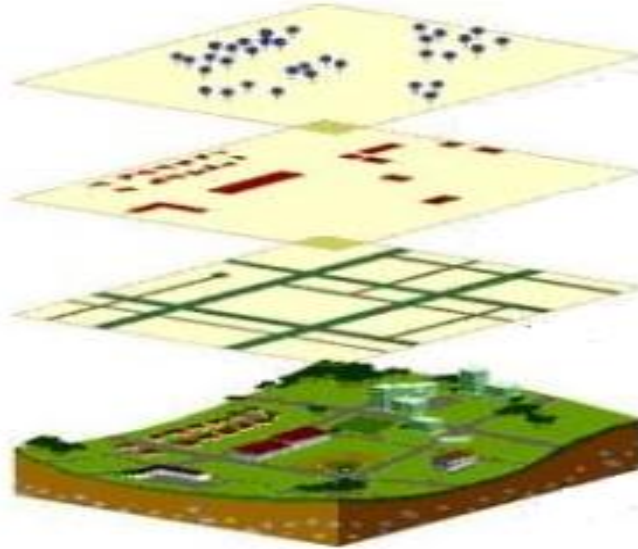


Ilustración 8: Sistemas de información geográfica

Fuente:(GIS, 2013)

Un sistema de información geográfica (SIG) integra datos para capturar, gestionar, analizar y mostrar todas las formas de información geográficamente referenciada, que nos permite ver, comprender, cuestionar, interpretar y visualizar datos de muchas maneras que revelan las relaciones, patrones y tendencias en forma de mapas, globos terráqueos, informes y gráficos. (GIS, 2013).

1.4.1 Base de datos espaciales

Son base de datos que almacenan datos espaciales o Geo-referenciados, mediante una plataforma integrada, y previa transformación hacia un sistema de referencia espacial establecida por el motor de base de datos. (HSU, 2011).

Datos espaciales.

Son datos que están asociados a un punto de la superficie terrestre. Y que son representados según la utilidad que se vaya a dar a la información geográfica.

- ✚ Punto: Representa a un punto sobre la superficie terrestre determinado por sus coordenadas latitud y longitud.
- ✚ Línea: Se representa por una sucesión de puntos.

- ✚ Polígono: Se representan por un conjunto de líneas y puntos cerrados para formar una zona perfectamente definida.

1.4.2 postGIS

Es una extensión que convierte el sistema de base de datos PostgreSQL en una base de datos espacial. La combinación de ambos es una solución perfecta para el almacenamiento, gestión y mantenimiento de datos espaciales.(HSU, 2011).

PostGIS está desarrollado en lenguaje C, C++ y PL/PgSQL, y utiliza entre otras las bibliotecas de software libre GEOS para el análisis espacial y proyecciones Proj4

Los datos espaciales requieren ser almacenados y gestionados en bases de datos a través del lenguaje SQL extendido, ya que se obtienen un rendimiento significativamente superior, que a manejar estos datos dentro de archivos en formato SHAPE(LLARIO, 2012).

PostGIS es software libre, pues tienen una licencia GNU y es compatible con los estándares de OGC.(HSU, 2011).

- Soporta tipos de datos espaciales, índices espaciales y tiene cientos de funciones espaciales.
- Permite importar y exportar datos a través de varias herramientas conversoras (shp2pgsql, pgsq2sp).

Existen herramientas de escritorio para visualizar archivos shape: QGIS, mezoGIS, OpenJUMP, ZlgGIS, ArcGIS⁵ y entre otros.

Tipos de postGIS

- geometry: Tipo definido por postGIS para almacenar geometrías en el formato binario interno.
- geography: Tipo definido por postGIS para almacenar geometrías en coordenadas geográficas (WGS 84), para realizar análisis espacial sobre el esferoide.
- raster: Tipo personalizado por postGIS para almacenar información en formato raster.
- Cargar datos GIS en la base de datos espacial.(LLARIO, 2012).

⁵ ArcGIS es el nombre de un conjunto de productos de software en el campo de los Sistemas de Información Geográfica

- ✚ Como primer paso es crear una tabla.

```
CREATE TABLE info(id serial, fecha date)
```

- ✚ Modificamos la tabla creada, añadiendo una columna de tipo geométrico

```
SELECT  
ADDGEOMETRYCOLUMN("nombre_base_datos","nombre_tabla","nombre_columna","TY  
PE", "COORD_DIMENSION ", "DIMENSION")
```

- ✚ Subir un archivo shape

```
shp2pgsql -g geog -G "RUTA" | psql -h localhost -U postgres -p 5432 -d postgis_in_action
```

```
shp2pgsql -g geog -G /data/roadtrl020.shp ch01.roads_geog | psql -h localhost -U postgres -p  
5432 -d postgis_in_action
```

1.5 Open Layers

Es una librería de JavaScript para la visualización de mapas en nuestro navegador web. Altamente extensible, que sirve como la base de todas las interfaces de mapas Web. OpenLayers accede a los datos a través de estándares de la industria, dejándola libre de dependencias del servidor.

Al ser una librería del lado del cliente, es un visor de mapas en JavaScript, la descarga de estos se realizar directamente desde el navegador a través de AJAX; no genera tráfico en el servidor, los mapas se descargan directamente del servidor de mapas en línea.

Ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, Mapas comerciales (Google Maps, Bing, Yahoo, OpenStreetMap y entre otros).(PÉREZ, 2012).

Características

- OpenLayers.Layer.WMS : Hace referencia al div en donde se imprimirá el mapa.
- OpenLayers.Layer.WMS: Definimos una capa base, la cual accederá hacia los servidores de Open Streetmaps.
- addLayer: Agregamos la capa base.
- OpenLayers.Control.LayerSwitcher: Podemos agregar controles hacia nuestro mapa.
- OpenLayers.Control.MousePosition: Muestra el movimiento del puntero sobre el mapa.

- `OpenLayers.Control.ScaleLine`: Permite regular la escala de nuestro mapa.
- `OpenLayers.LonLat`: Especifica las coordenadas para visualizar el mapa.

1.5.1 Proyecciones de un mapa

Los mapas de la tierra son representaciones realmente poco perfectas; todos los mapas tienen alguna distorsión o margen de error. La razón de esto es porque se representa un objeto tridimensional, en dos dimensiones. La única forma de evitar las distorsiones sería usando un mapa esférico pero, en la mayoría de los casos, sería demasiado grande y poco útil.

Una proyección es una representación de todo, o las partes de la superficie de una esfera tridimensional sobre un plano dimensional.

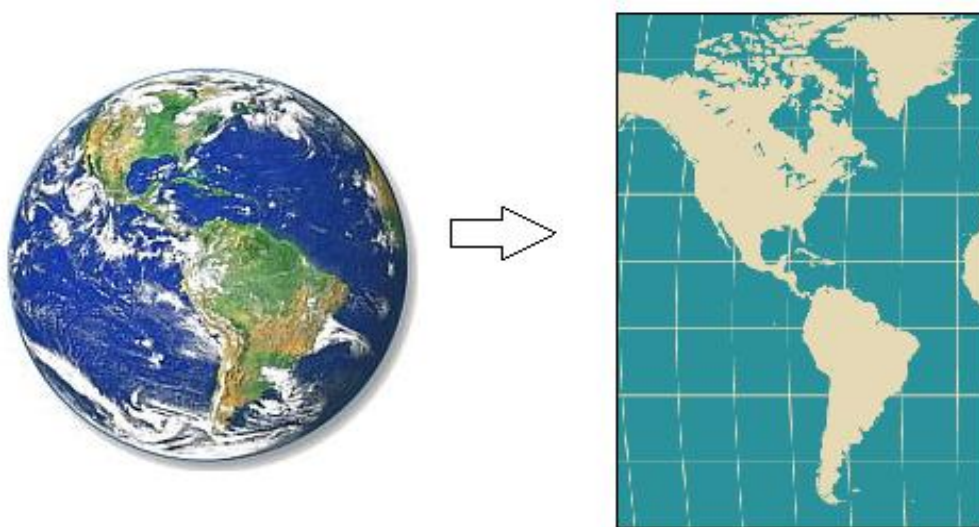


Ilustración 9: Proyecciones de un mapa

Fuente:(HSU, 2011)

Identificadores de referencia espacial (SRID)

Un identificador de sistema de referencia espacial (SRID) es un valor único que se utiliza para identificar de forma más exacta una proyección. Estos sistemas de coordenadas son el centro de todos los sistemas de información geográfica. (HSU, 2011).

1.5.2 European Petroleum Survey Group

EPSG es el acrónimo de European Petroleum Survey Group, organización relacionada con la industria petrolera en Europa. Este organismo estuvo formado por especialistas en geodesia, topografía y cartografía aplicadas al área de exploración y desarrolló un repositorio de parámetros geodésicos que contiene información sobre sistemas de referencia, proyecciones cartográficas y elipsoides de todo el mundo.

En cuanto a la importancia de los códigos EPSG podemos decir que son ampliamente utilizados en la definición de datos de posición, en los Sistemas de Información Geográfica, por lo que es muy útil conocerlos para todas aquellas actividades que requieran gestionar o manipular datos espaciales en ambientes digitales. (varstein, 2013).

Códigos EPSG	Códigos EPSG (European Petroleum Survey Group)
EPSG:3034	ETRS89/ETRS-LCC Para la cartografía pan-Europeo a escalas menores o iguales a 1:500.000 IDEE
EPSG:3035	ETRS89/ETRS-LAEA Para representación y análisis estadístico pan-Europeos IDEE
EPSG:3395	WGS84 Mercator
EPSG:4230	Coordenadas Geográficas ED50
EPSG:4258	Coordenadas Elipsoidales ETRS89 IDEE
EPSG:4267	Coordenadas Geográficas North American Datum 1927 (NAD 27)
EPSG:4269	Coordenadas Geográficas North American Datum 1983 (NAD 83)
EPSG:4324	Coordenadas Geográficas WGS 72BE Transit Broadcast Ephemeris
EPSG:4326	Coordenadas Geográficas WGS84
EPSG:23028	Proyección UTM ED50 Huso 28 N
EPSG:23029	Proyección UTM ED50 Huso 29 N
EPSG:23030	Proyección UTM ED50 Huso 30 N
EPSG:23031	Proyección UTM ED50 Huso 31 N
EPSG:25828	Proyección UTM ETRS89 Huso 28 N
EPSG:25829	Proyección UTM ETRS89 Huso 29 N
EPSG:25830	Proyección UTM ETRS89 Huso 30 N
EPSG:25831	Proyección UTM ETRS89 Huso 31 N
EPSG:32628	Proyección UTM WGS84 Huso 28 N
EPSG:32629	Proyección UTM WGS84 Huso 29 N
EPSG:32630	Proyección UTM WGS84 Huso 30 N
EPSG:32631	Proyección UTM WGS84 Huso 31 N
EPSG:900913	Proyección Google Mercator

Tabla 10: Códigos EPSG utilizados, En la definición de datos de posición

Fuente: (Rediam, 2013)

- **Proyecciones de tipo Mercator**

Es el tipo de proyección utilizado por Google y Open Streetmaps. La idea de Mercator responde a las exigencias matemáticas de la proyección cilíndrica. La característica más destacable de esta proyección es que tanto los meridianos como los paralelos son líneas rectas y se cortan perpendicularmente. Los meridianos son líneas rectas paralelas entre sí dispuestas verticalmente a la misma distancia unos de otros. Los paralelos son líneas rectas paralelas entre sí dispuestas horizontalmente pero aumentando la escala a medida que nos alejamos del ecuador. Este aumento de escala hace que no sea posible representar en el mapa las latitudes por encima de los 80°.

El mapa de Mercator es realmente conforme, la forma de los países es real, pero su superficie aumenta exageradamente en las latitudes altas.

El éxito de la proyección de Mercator se debe a que cualquier línea recta que se trace marca el rumbo real, con lo cual se puede navegar siguiendo con la brújula el ángulo que se marca en el mapa. A esta línea de rumbo se llama loxodrómica.(Thetuzaro, 2011).

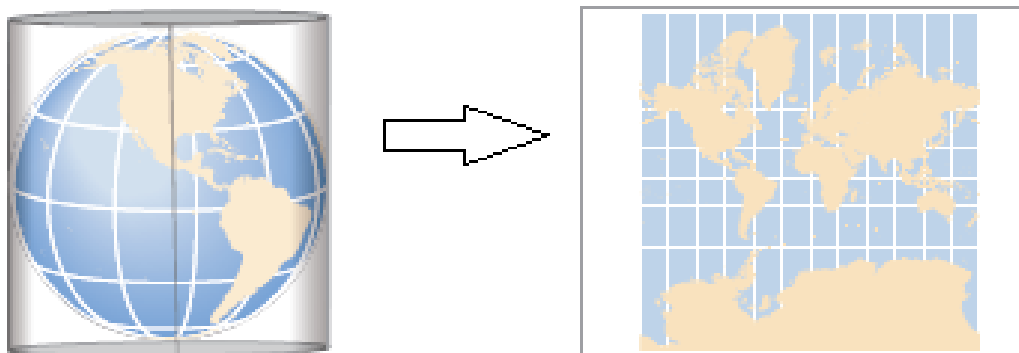


Ilustración 10: Proyección Mercator cilíndrica.

Fuente:(Thetuzaro, 2011).

Proyección cilíndrica

Los mapas se dibujan proyectando puntos de la esfera sobre un cilindro.

Es el mejor sistema para representar las zonas del Ecuador y los trópicos, pero a medida que nos acercamos a los polos la deformación de la Tierra es mayor.

Coordenadas Geográficas WGS84

El WGS84 es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia) por medio de tres unidades

dadas. WGS84 son las siglas en inglés de World Geodetic System 84 (que significa Sistema Geodésico Mundial 1984). (HSU, 2011).

1.5.3 Mostar un mapa con OpenLayers

Para incluir nuestro primer mapa no tenemos más que disponer una capa en el código HTML identificándola con un id.

✚ Agregamos la ruta de la librería.

```
<script type="text/javascript"
  src="http://www.openlayers.org/api/OpenLayers.js">
</script>
```

✚ Incluir un div el cual mostrará el mapa.

```
<div id="miMapa" style="width: 512px; height:512px;">
```

✚ Código JavaScript que define el objeto mapa a mostrarse.(PÉREZ, 2012).

```
<script type="text/javascript">
var mapa = new OpenLayers.Map("miMapa");
var layerBase = new OpenLayers.Layer.WMS(
  "OpenLayers WMS",
  "http://vmap0.tiles.osgeo.org/wms/vmap0",
  {layers: "basic"});
mapa.addLayer(layerBase);
mapa.addControl(new OpenLayers.Control.LayerSwitcher(true));
mapa.addControl(new OpenLayers.Control.MousePosition({ numDigits: 2 }));
mapa.addControl(new OpenLayers.Control.ScaleLine());
mapa.setCenter(new OpenLayers.LonLat(0, 40), 5);
</script>
```

✚ Como resultado tenemos una capa base.

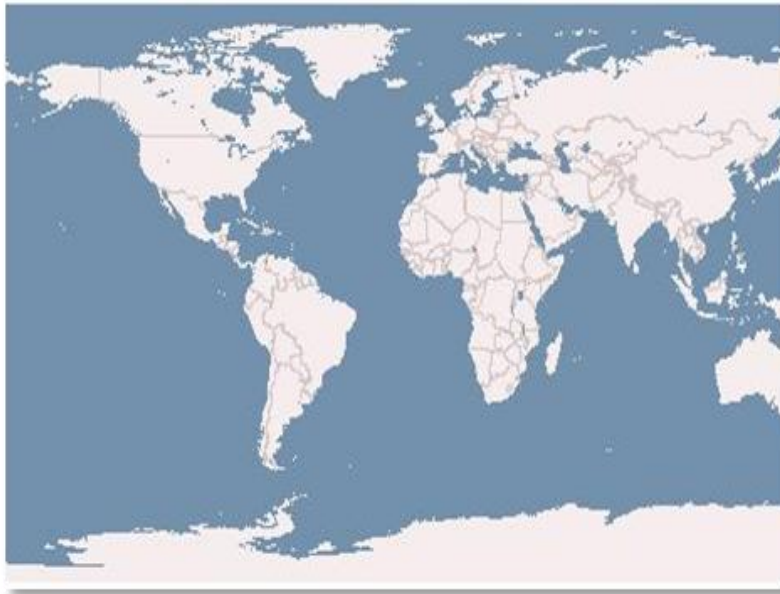


Ilustración 11: Capa base de OSM

Fuente:(PÉREZ, 2012)

1.5.4 Definir proyecciones

A continuación se detalla cómo se transforma un sistema de coordenadas hacia un sistema de referencia, utilizando las proyecciones EPSG:4326 aceptados por Google y Open Streetmaps.

✚ Definir una variable a la cual se le asigna un objeto proyectado.(PÉREZ, 2012).

Método que recibe como parámetros: latitud y longitud (EPSG:4326) y realiza la transformación hacia EPSG:900913, para poder imprimir el punto sobre el visor de mapas.

```
function tranforma(lati, long)
{
    varpoint1;
    point1 = new OpenLayers.Geometry.Point(lati,long).transform(Geographic, Mercator);
    return point;
}
```

1.6 Servidor Web

Un servidor web o servidor HTTP es un programa que establece una comunicación entre el cliente y servidor mediante la ejecución de una aplicación, realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

1.6.1 Funciones de un servidor Web

La función principal es de mantener a la espera peticiones de ejecución que le hará un cliente o usuario de Internet. El servidor se encarga de contestar las peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a la información disponible.(Franco, 2001).

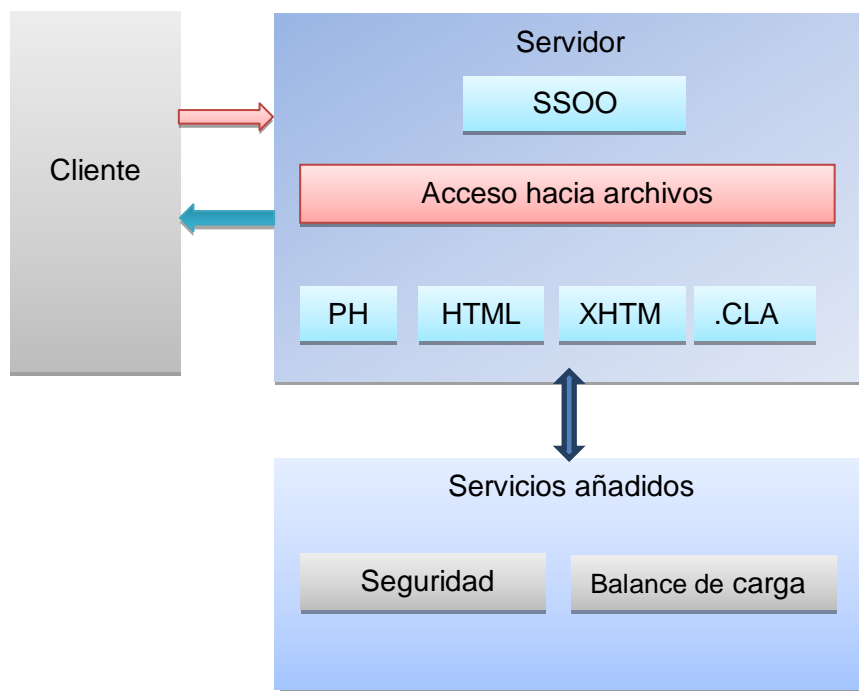


Ilustración 12: Arquitectura funcional de un Servidor Web

Fuente:(Franco, 2001)

1.6.2 Servidores web más utilizados

- Apache: Este es el más común y utilizado por la comunidad de desarrolladores en todo el mundo. Además es gratuito y de código abierto, así que podríamos ejecutarlo dentro de sistema operativo Windows o Linux.
- Microsoft IIS: servidor que funciona sobre plataformas Windows, para empatarlo sobre otro sistema operativo, se recomienda utilizar una máquina virtual.

1.6.3 Operaciones que realiza un servidor

- Start: Lanza o inicia el servidor web.
- Stop: Para el servidor web.
- Restart: Si se ha hecho cambios en la configuración, se deberá parar el servidor y volver a arrancarlo.
- Services: Abrir una nueva ventana para ver los servicios del ordenados (entre ellos está el del servidor web)
- Connect: Conectarse a un servidor web remoto (instalado en otro ordenador) para administrarlo desde el ordenador local.
- Disconnect: desconectarse del servidor remoto.

1.6.4 Ventajas de los servidores Apache e IIS

Ventajas.	
Servidor <i>Apache</i>	Servidor <i>IIS</i>
<ul style="list-style-type: none"> ▪ Es personalizable 	<ul style="list-style-type: none"> ▪ Es confiable, seguro y administrable.
<ul style="list-style-type: none"> ▪ La arquitectura modular de apache permite construir un servidor hecho a la medida 	<ul style="list-style-type: none"> ▪ Al momento de la instalación permite elegir sobre que servidor web va a correr (<i>Apache</i> o <i>IIS</i>)
<ul style="list-style-type: none"> ▪ Permite la implementación de SSOO últimos y más nuevos protocolos. 	<ul style="list-style-type: none"> ▪ Desarrollada y es compatible con las aplicaciones beneficiándose con un único entorno de alojamiento de aplicaciones integrado con compatibilidad total.
<ul style="list-style-type: none"> ▪ En cuanto a la administración, los archivos de configuración de apache están en <i>ASCII</i>, por lo que pueden ser editados tan solo desde un editor de texto. 	<ul style="list-style-type: none"> ▪ Interfaz amigable con el usuario y más intuitiva.
<ul style="list-style-type: none"> ▪ El servidor puede ser administrado vía línea de comandos, lo que hace la administración remota muy conveniente. 	<ul style="list-style-type: none"> ▪ Posee mayor soporte técnico e información sobre los modos de configuración.

Tabla 11: Ventajas de los servidores Apache y IIS

Fuente:(Espinoza, 2013)

1.6.5 Desventajas de los servidores Apache e IIS

Desventajas	
Servidor Apache	Servidor IIS
<ul style="list-style-type: none"> Esta incluye formatos de configuración no estándar. 	<ul style="list-style-type: none"> Tiende a limitarse en las versiones que no so de la familia "Server"
<ul style="list-style-type: none"> No cuenta con una buena administración 	<ul style="list-style-type: none"> Posee vulnerabilidades
<ul style="list-style-type: none"> Falta de integración 	<ul style="list-style-type: none"> Este servidor no es multiplataforma, solo funciona bajo Windows.

Tabla 12: Desventajas de los servidores Apache e IIS

Fuente:(Espinoza, 2013)

1.7 Aplicaciones distribuidas

Es una aplicación desarrollada con distintos componentes que se ejecutan en entornos separados, con frecuencia en distintas plataformas y se comunican entre sí a través de Internet.(Javier, 2011).

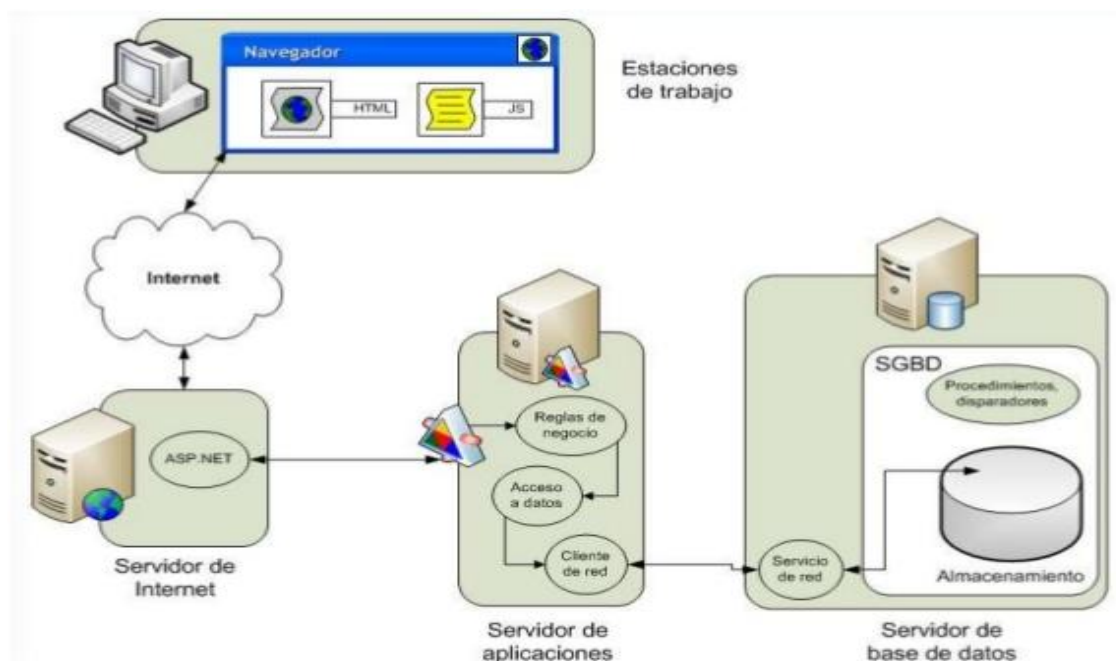


Ilustración 13: Arquitectura de una aplicación distribuida

Fuente:(Javier, 2011)

Distribución

La distribución se refiere al diseño y construcción de sistema por componentes, a las cuales le son asignadas un conjunto específico de funcionalidades dentro de un sistema.

Esta distribución de componentes se basa en que la plataforma y arquitectura de desarrollo es diferente, sin embargo, lo que tiene implícito esta definición, es que para realizar esta separación física primero debe tenerse clara la separación lógica de las partes de una aplicación, esto quiere decir que programáticamente existe una forma de separar o agrupar los componentes.(Javier, 2011).

1.7.1 La separación física

No es en todas la ocasiones en “maquinas diferentes” de acuerdo a la arquitectura también puede ser la ubicación de un conjunto de funcionalidades en archivos, rutas o montadas sobre tecnologías diferentes dentro de la misma máquina.

1.7.2 Distribución lógica

Lo entenderemos como separación por “capas” y cuando hablemos de distribución física usaremos el término separación en “niveles”.

“las capas dentro de una arquitectura son un conjunto de servicios especializados que pueden ser accesibles por múltiples clientes y que deben ser fácilmente reutilizables”.

Una capa puede contener muchos componentes, un mismo componente puede ubicarse en varias capas de acuerdo a su naturaleza y a las consideraciones explicitas de la arquitectura.

1.7.3 Servicios Web

Son aplicaciones cliente servidor que en su mayoría utilizan el protocolo HTTP para transferir información en tiempo real entre distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de Servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. Los servicios de Web ofrecen comunicación para sistemas con diferentes sistemas operativos y plataformas de desarrollo

La W3C define a un “Servicio web” como un sistema de software diseñado para permitir interoperabilidad maquina a máquina en un red. Se trata de API que son publicadas, localizadas e invocadas a través de la web, es decir una vez desarrolladas, son instaladas en un servidor, y otras aplicaciones, pueden descubrirlas desde otros ordenadores o dispositivos e invocar uno de sus servicios. (JOE, JEREMY, & C, 2006).

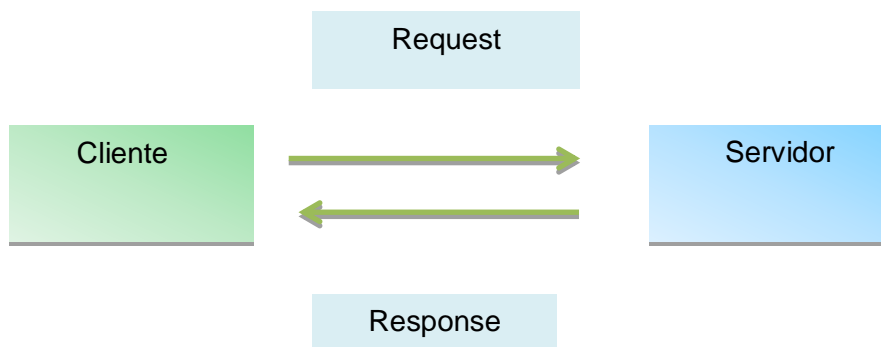


Ilustración 14: Esquema de los Servicios web

Fuente: (JOE, JEREMY, & C, 2006)

Características

- Son interoperables y se pueden construir y operar en múltiples tecnologías y plataformas, que no necesariamente pueden ser móviles.
- Son reglamentados por las organizaciones OASIS y W3C.
- Fomentar la independencia entre la aplicación que usa el servicio web y el propio servicio.
- Los formatos de respuestas normalmente están basados en texto.

Operaciones

- GET hace la operación READ.
- DELETE hace la operación DELETE.
- PUT se usa normalmente para hacer UPDATE
- POST se usa normalmente para hacer CREATE.

Estándares

- Web Services Protocol Stack: Un servicio web pila de protocolos es una pila de protocolos, que se utiliza para definir, localizar, implementar y hacer que los servicios Web interactúan unos con otros.

- XML: Es el formato estándar para intercambiar los datos entre aplicaciones.
- SOAP: los mensajes del protocolo se definen utilizando un escrito formato XML, que ha de ser consensuado por ambas parte.
- WSDL: Se utiliza para la localización y ubicación de servicios en Internet.
- UDDI: Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.
- WS-Security: Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

Interoperabilidad

Distintas aplicaciones, en lenguajes de programación diferente, ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos.

- La interoperabilidad se consigue mediante el uso de estándares abiertos.
- Servicios web se asientan sobre protocolos y estándares ya existentes y muy difundidos HTTP, XML y entre otros.
- Uso de protocolos específicos entendibles, que no imponen restricciones sobre las aplicaciones a las que dan acceso ni sobre las tecnologías que las implementan.
- OASIS y W3C son Organizaciones responsables de definir la arquitectura y estándares para los servicios web

- SOAP (Siglas de Simple Object Access Protocol).

Es un protocolo estándar que define cómo dos objetos en diferentes procesos, pueden comunicarse por medio de intercambios de datos XML, el punto identificativo de SOAP es que las operaciones son definidas como puertos WSDL (Web Services Description Language). Es por esto que será aconsejable utilizar este protocolo en entornos donde se establecerá un contrato formal y donde se describirán todas las funciones de la interfaz así como el tipo de datos utilizados tanto de entrada como de salida. El lenguaje WSDL nos permitirá definir claramente cualquier detalle de las funciones de nuestro WS. (GIRONÉS, 2011).

- REST (Representational State Transfer).

Es un estilo de arquitectura de software para sistemas distribuidos tales como la web, a diferencia de SOAP, se centra en el uso de los estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional. Las operaciones (o funciones) se solicitarán mediante GET, POST, PUT y DELETE, por lo que no requiere de implementaciones especiales para consumir estos servicios. Además se podrá utilizar JSON en vez de XML como contenedor de la información, por lo que será aconsejable

utilizar este protocolo cuando busquemos mejorar el rendimiento, o cuando disponemos de escasos recursos, como sería el caso de los dispositivos móviles. (GIRONÉS, 2011).

1.8 Lenguajes de programación y librerías

1.8.1 PHP

Lenguaje de programación primario del servidor, es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor, actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Para el uso de este programa, en primer lugar se requiere de un servidor que soporte PHP. Luego el cliente debe crear los archivos y guardarlos en el directorio web y el servidor envía la petición al intérprete de PHP quien los analiza y compila. El resultado es enviado al servidor y al cliente. El PHP, concluido el trabajo, se traduce en la sigla HTML (el estándar para la codificación de páginas web; es decir, el PHP hace el trabajo internamente en el servidor, para entregar al usuario o navegador web el resultado en código estándar).(HOLZER, 2009).

1.8.2 JAVA

Java es un lenguaje de programación y la primera plataforma informática creada por SUN Microsystems en 1995. Es la tecnología subyacente que permite la creación de aplicaciones web y escritorio, además sirve de plataforma para el desarrollo de aplicaciones móviles para dispositivos móviles ANDROID.(GIRONÉS, 2011).

1.8.3 TWITTER BOOTSTRAP

Es una colección de herramientas de software libre para la creación de sitios y aplicaciones web dinámicas y con altos estándares de calidad. Contiene plantillas de diseño basadas en HTML5 y CSS 3 con tipografías, formularios, botones, gráficos, barras de navegación y demás componentes de interfaz, así como extensiones opcionales de JavaScript.

Es el proyecto más popular en GitHub y es usado por la NASA y la MSNBC junto a demás organizaciones. La mayor ventaja es que podemos crear interfaces que se adapten a los distintos navegadores (responsive design) apoyándonos en un framework potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo. (Rodríguez T. , 2012).

Característica

- ✚ Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- ✚ Se integra perfectamente con las principales librerías JAVASCRIPT, por ejemplo JQUERY.
- ✚ Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- ✚ Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
- ✚ Funciona con todos los navegadores, incluido Internet Explorer usando HTML para que reconozca los tags HTML5
- ✚ Dispone de distintos layout predefinido con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.
- ✚ Convierte los componentes de una web en full responsive.

Componentes web

Bootstrap ofrece distintos componentes que podemos usar con unos estilos predefinidos y fáciles de configurar. Además de los distintos plugins integrados con JQUERY que disponemos en la versión 2.0 de Bootstrap.



Ilustración 15 : Botones bootstrap

Fuente:(Rodríguez T. , 2012)

- ✚ Botones: Se pueden aplicar tanto a enlaces como a etiquetas button/input simplemente usando la clase btn. Así podemos distinguir el propósito de cada botón con los distintos estilos prefijados o variar su tamaño.
- ✚ Dropdown: Con ellos podemos asociar un menú desplegable a un botón de forma sencilla agrupando distintas opciones.
- ✚ Formularios Bootstrap: Cuenta con distintos layout que podemos adaptar con las principales necesidades. Además de incluir distintos elementos para remarcar distintas acciones sobre los formularios: focused, disabled, control-group y entre otros.
- ✚ Plugin de jQuery Bootstrap: Cuenta con distintos plugins que nos permiten crear ventanas modales, o crear el típico tooltip sobre algún elemento de la página.

1.9 SSOO Móviles

Un sistema operativo es un programa que se encarga de gestionar y controlar los procesos básicos de un dispositivo móvil, optimizando el uso de sus recursos. Inicialmente fue desarrollado para las computadoras, pero hoy son utilizados en los teléfonos celulares para tener esa misma interlocución entre el dispositivo y el usuario.

Gran parte del programa está diseñado para controlar aplicaciones. También administrar las partes físicas de un dispositivo. Es importante destacar que no todos los sistemas operativos son iguales, por lo que un programa que corre en un sistema operativo específico, probablemente no funcionará en otro. (GIRONÉS, 2011).

Características

Kernel

El núcleo o kernel proporciona el acceso a los distintos elementos del hardware del dispositivo. Ofrece distintos servicios a las superiores como son los controladores o drivers para el hardware, la gestión de procesos, el sistema de archivos y el acceso y gestión de la memoria.

Middleware

El middleware es el conjunto de módulos que hacen posible la propia existencia de aplicaciones para móviles. Es totalmente transparente para el usuario y ofrece servicios claves como el motor de mensajería y comunicaciones, códecs multimedia, intérpretes de páginas web, gestión del dispositivo y seguridad.

Entorno de ejecución de aplicaciones

El entorno de ejecución de aplicaciones consiste en un gestor de aplicaciones y un conjunto de interfaces programables abiertas y programables por parte de los desarrolladores para facilitar la creación de software.

Interfaz de usuario

Las interfaces de usuario facilitan la interacción con el usuario y el diseño de la presentación visual de la aplicación. Los servicios que incluye son el de componentes gráficos (botones, pantallas, listas y entre otros)

1.9.1 SSOO más utilizados

- **ANDROID:** Actualmente Android pertenece a Google, pero es un sistema abierto cualquier fabricante puede desarrollar en él sus productos.
- **IOS:** (Anteriormente denominado iPhone OS) es un sistema operativo móvil de Apple desarrollado originalmente para el iPhone siendo después usado en el iPod Touch y en el iPad.

- MAEMO: Basado en Linux Debían (casi todo en código abierto) y desarrollado por Nokia para Smartphone y tablets.
- ALM WEB: aprovecha tecnología web como XHTML, JAVASCRIPT Y CSS pertenece a HP.
- RIM: Este Sistema Operativo ha sido pionero en la gestión del correo electrónico y está muy orientado a un uso profesional.
- SYMBIAN: Es el sistema más extendido, nacido de la alianza de varias compañías de móviles. El actual propietario es Nokia.
- WINDOWS: La mayor ventaja de este sistema es la de cualquier producto de Microsoft compatibilidad, facilidad de uso, integración con los sistemas Windows.

1.9.2 Tabla comparativa de SSOO móviles



	Apple iOS 6	Android 4.2	Windows Phone 7	BlackBerry OS 7	Symbian 9.5
Compañía	Apple	Open Handset Alliance	Windows	RIM	Symbian Foundation
Núcleo del SO	Mac OS X	Linux	Windows CE	Mobile OS	Mobile OS
Familia CPU soportada	ARM	ARM, MIPS, Power, x86	ARM	ARM	ARM
Lenguaje de programación	Objective-C, C++	Java, C++	C#, muchos	Java	C++
Licencia de software	propietaria	software libre y abierto	Propietaria	propietaria	software libre
Año de lanzamiento	2007	2008	2010	2003	1997
Motor del navegador web	WebKit	WebKit	Pocket Internet Explorer	WebKit	WebKit
Soporte Flash	No	Sí	No	Sí	Sí
HTML5	Sí	Sí	Sí	Sí	No
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry App World	Ovi Store
Número de aplicaciones	400.000	300.000	50.000	30.000	50.000
Coste publicar	\$99 / año	\$25 una vez	\$99 / año	sin coste	\$1 una vez
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux
Actualizaciones automáticas del S.O.	Sí	depende del fabricante	depende del fabricante	Sí	Sí
Soporte memoria externa	No	Sí	No	Sí	Sí
Fabricante único	Sí	No	No	Sí	No
Variedad de dispositivos	modelo único	muy alta	baja	baja	muy alta
Tipo de pantalla	capacitativa	capacitiva /resistiva	capacitativa	/resistiva capacitativa	capacitiva /resistiva
Aplicaciones nativas	Sí	Sí	No	No	Sí

Tabla 13: Características principales de los SSOO móviles

Fuente:(GIRONÉS, 2011)

1.9.3 SSOO Android

Es un sistema operativo basado en el código libre, y no es un teléfono móvil como algunos piensan, que está presente en numerosos dispositivos móviles de última generación. Actualmente soporta dispositivos móviles como: HTC, SONY, SAMSUNG, MOTOROLA, PANASONIC, LG y entre otros.(GIRONÉS, 2011).

Arquitectura Android

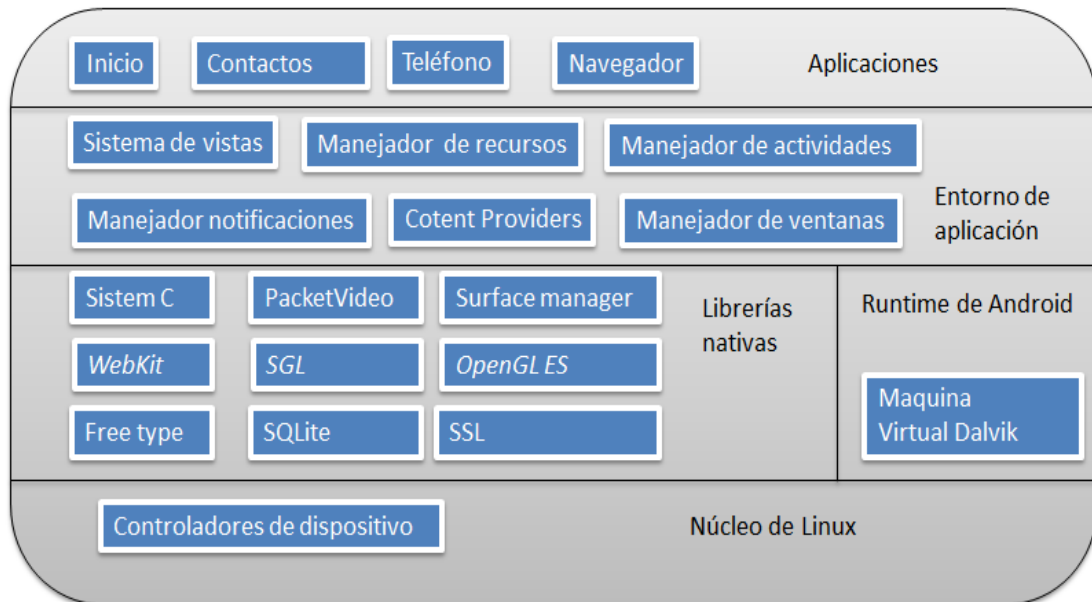


Ilustración 16: Arquitectura de Android

Fuente: (GIRONÉS, 2011)

El Núcleo Linux

El núcleo de Android está formado por el sistema operativo Linux versión 2.6. Esta capa proporciona servicios como la seguridad, el manejo de la memoria, el multiproceso, la pila de protocolos y el soporte de drivers para dispositivos.

Esta capa del modelo actúa como capa de abstracción entre el hardware y el resto de la pila. Por lo tanto, es la única que es dependiente del hardware.

Runtime de Android

Está basado en el concepto de máquina virtual utilizado en Java. Dado las limitaciones de los dispositivos donde ha de correr Android (poca memoria y procesador limitado) no fue posible utilizar una máquina virtual Java estándar. Google tomó la decisión de crear una nueva, la máquina virtual Dalvik, que respondiera mejor a estas limitaciones.

Algunas características de la máquina virtual Dalvik que facilitan esta optimización de recursos son: que ejecuta ficheros Dalvik ejecutables (.dex) –formato optimizado para ahorrar memoria. Además, está basada en registros. Cada aplicación corre en su propio proceso Linux con su propia instancia de la máquina virtual Dalvik. Delega al kernel de Linux algunas funciones como threading y el manejo de la memoria a bajo nivel.

También se incluye en el Runtime de Android el “core libraries” con la mayoría de las librerías disponibles en el lenguaje Java.

Librerías nativas

Incluye un conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Muchas de las librerías utilizan proyectos de código abierto. Algunas de estas librerías son:(GIRONÉS, 2011).

- ✚ System C library: Una derivación de la librería BSD de C estándar (libc), adaptada para dispositivos embebidos basados en Linux.
- ✚ Media Framework: Librería basada en PacketVideo's OpenCORE; soporta codecs de reproducción y grabación de multitud de formatos de audio vídeo e imágenes MPEG4, H.264, MP3, AAC, AMR, JPG y PNG.
- ✚ Surface Manager: Maneja el acceso al subsistema de representación gráfica en 2D y 3D.
- ✚ WebKit: soporta un moderno navegador web utilizado en el navegador Android y en la vista webview. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.
- ✚ SGL: Motor de gráficos 2D.
- ✚ Librerías 3D: Implementación basada en OpenGL ES 1.0 API. Las librerías utilizan el acelerador hardware 3D si está disponible, o el software altamente optimizado de proyección 3D.
- ✚ FreeType: Fuentes en bitmap y renderizado vectorial.
- ✚ SQLite: Potente y ligero motor de bases de datos relacionales disponible para todas las aplicaciones.
- ✚ SSL: Proporciona servicios de encriptación Secure Socket Layer.

Entorno de aplicación

Proporciona una plataforma de desarrollo libre para aplicaciones con gran riqueza e innovaciones (sensores, localización, servicios, barra de notificaciones,).

Esta capa ha sido diseñada para simplificar la reutilización de componentes. Las aplicaciones pueden publicar sus capacidades y otras pueden hacer uso de ellas (sujetas

a las restricciones de seguridad). Este mismo mecanismo permite a los usuarios reemplazar componentes.

Una de las mayores fortalezas del entorno de aplicación de Android es que se aprovecha el lenguaje de programación Java. El SDK de Android no acaba de ofrecer todo lo disponible para su estándar del entorno de ejecución Java (JRE), pero es compatible con una fracción muy significativa de la misma.(GIRONÉS, 2011).

Características (SORIANO, 2012)

- ✚ Views: Extenso conjunto de vistas, (parte visual de los componentes).
- ✚ Resource Manager: Proporciona acceso a recursos que no son en código.
- ✚ Activity Manager: Maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- ✚ Notification Manager: Permite a las aplicaciones mostrar alertas personalizadas en la barra de estado.
- ✚ Content Providers: Mecanismo sencillo para acceder a datos de otras aplicaciones (como los contactos).

Aplicaciones

Este nivel está formado por el conjunto de aplicaciones instaladas en una máquina Android. Todas las aplicaciones han de correr en la máquina virtual Dalvik para garantizar la seguridad del sistema.

Normalmente las aplicaciones Android están escritas en Java. Para desarrollar aplicaciones en Java podemos utilizar el Android SDK. Existe otra opción consistente en desarrollar las aplicaciones utilizando C/C++. Para esta opción podemos utilizar el Android NDK (Native Development Kit).

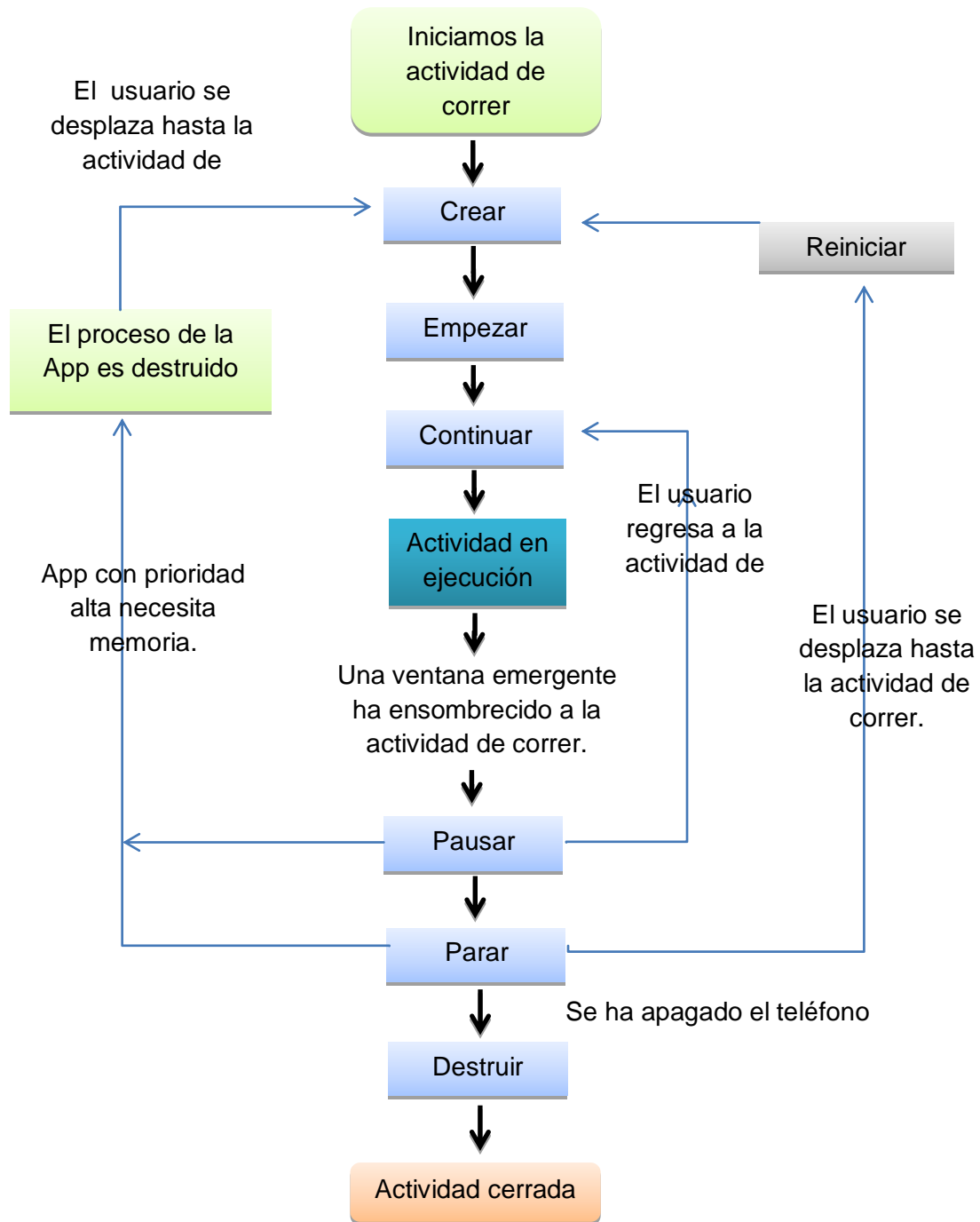


Ilustración 17 : Ciclo de vida de una aplicación Android
 Fuente:(Web-developía, 2013)

✚ Características de las aplicaciones móviles.

Ventajas	Desventajas
<ul style="list-style-type: none">▪ Aunque no dispongamos de Wi-Fi o de 3G podremos utilizar la aplicación que hemos descargado en nuestro dispositivo.▪ Una aplicación puede utilizar toda la pantalla del dispositivo y proporcionar una mejor interfaz al usuario.▪ La aplicación siempre estará visible en el escritorio del dispositivo, por lo que las oportunidades de que la utilice son mayores.	<ul style="list-style-type: none">▪ Para que un usuario pueda utilizar una aplicación primero debe descargársela, lo que requiere un mayor conocimiento y lealtad a la marca.▪ Una aplicación tiene un desarrollo más costoso que un sitio web móvil ya que tienen que ser desarrolladas por separado.▪ Las aplicaciones no están disponibles para ser descargadas en todos los dispositivos móviles.▪ Como las aplicaciones se instalan en los dispositivos de los usuarios no pueden realizarse actualizaciones, así que las versiones antiguas deben mantenerse.

Tabla 14: Ventajas y desventajas de las aplicaciones móviles

Fuente:(Veiss, 2013)

1.9.4 Google Play

Google Play (antes *Android Market*) es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo *Android*. Es una aplicación que está preinstalada en la mayoría de los dispositivos *Android* y que permite a los usuarios buscar, obtener información y descargar aplicaciones publicadas por desarrolladores terceros. En octubre de 2012, Google Play disponía de más de 700.000 aplicaciones.(Developers, 2013).

Interfaz

Google Play tiene un acceso fácil y rápido a sus aplicaciones. Las aplicaciones son creadas por desarrolladores de todo el mundo y posteriormente puntuadas por los usuarios de *Android*.

El menú tiene las siguientes opciones:

Mostrado: avanza por los iconos de la parte superior para ver las aplicaciones mostradas.

Aplicaciones: examina todas las aplicaciones o busca aplicaciones por categorías.

Juegos: examina todos los juegos o busca juegos por categorías.

Búsqueda: busca aplicaciones en Google Play.

Mis descargas: visualiza las aplicaciones que están instaladas en el dispositivo.

Las categorías de juegos y aplicaciones, del menú principal proporcionan submenús para que la búsqueda sea más sencilla. Los usuarios tienen la posibilidad de valorar las aplicaciones mediante un sistema similar a YouTube, con una escala del 1 al 5, también ofrece la posibilidad de poner comentarios sobre la aplicación. La novedad de la nueva versión de Google Play es también la posibilidad de añadir capturas de pantalla de su aplicación.

Aplicaciones

Las aplicaciones móviles están especialmente diseñadas para dispositivos portátiles. Estas pueden ser gratuitas o de pago. Inicialmente, las aplicaciones tenían una función puramente recreativa. Sin embargo, han ido evolucionando en aplicaciones más útiles, como son las aplicaciones para el registro de gastos, información deportiva, guías de restaurantes, callejeros. Actualmente las nuevas aplicaciones más innovadoras son las llamadas de realidad aumentada que combinan elementos reales y virtuales.

Aplicaciones de pago

Para vender aplicaciones debes crearte una cuenta de comerciante de Google Checkout, y subir el archivo de la aplicación a los servidores. Para crearte una cuenta de comerciante deberás proporcionar información privada, de contacto y financiera. El precio de la aplicación se puede cambiar en cualquier momento siempre y cuando no la hayas publicado anteriormente como gratuita. Los intervalos de precios permitidos son entre 0,99 y 200 dólares estadounidenses. Los desarrolladores de las aplicaciones de pago reciben un 70% del precio total de la aplicación, mientras que el 30% restante es destinado a las empresas. El beneficio obtenido de 'Google Play' es pagado a los desarrolladores a través sus cuentas en el sistema Google Checkout.

Desarrolladores

La gran novedad que aporta Google Play hace referencia a los desarrolladores: estos serán capaces de hacer su contenido disponible en un servicio abierto el servicio de Google que ofrece una retroalimentación y sistema de calificación similar a YouTube. Los desarrolladores tendrán un entorno abierto y sin obstáculos para hacer su contenido disponible.

CAPÍTULO II

2 Estudio de Factibilidad

2.1 Mercado

Existen varias definiciones de mercado, sin embargo todas ellas llegan a un mismo sentido. A continuación se presentan las siguientes.(AngelFire, 2013).

- ✚ Es el lugar en el cual convergen las fuerzas de la oferta y la demanda para establecer un precio único.
- ✚ Es el área en que confluyen las fuerzas de la oferta y la demanda para realizar la transacción de bienes y servicios a un precio determinado.
- ✚ Es el lugar en el que se compra y se vende mercancías a un precio definido.

El mercado de empresas que desarrollan aplicaciones móviles ha registrado un crecimiento importante tanto en la oferta como en la demanda, sin embargo, es difícil introducir al mercado un nuevo producto debido a la gran competencia que existe.

Propósito de estudio de mercado

- ✚ Determinar que existe un mercado objetivo y una necesidad sobre un producto.
- ✚ Utilizar métodos tanto administrativos como tecnológicos que sean apropiados para la creación del producto.
- ✚ Comprobar que es posible desarrollar un aplicativo móvil
- ✚ Demostrar que el proyecto es económicamente rentable

2.2 Definición del producto

Aplicativo web

La página web de administración permite al usuario, administrar la información de un sitio que pertenece a una determinada categoría, la información que gestione un usuario, no podrá ser alterada por otros usuarios.

Aplicativo móvil

El aplicativo móvil en formato APK⁶. Estará disponible en la tienda Google play y podrá ser instalado en un teléfono o Tablet que disponga de un SSOO Android.

⁶Application PacKage File: Es un paquete para el sistema operativo Android.

2.3 Selección de la muestra

En el proceso cualitativo, es un grupo de personas, eventos, sucesos, comunidades y entre otros, sobre el cual se habrán de recolectar los datos, sin que necesariamente sea representativo del universo o población que se estudia.

En este caso tomamos el número de habitantes de la ciudad de Ibarra como población, especialmente la zona urbana y rural.

La población es 181.175, fuente página oficial de Municipio de Ibarra.

2.3.1 Determinación del tamaño de la muestra

Fórmula para calcular el tamaño de la muestra.(Rabolini, 2013).

Para calcular el tamaño de la muestra suele utilizarse la siguiente fórmula:

$$n = \frac{N\sigma^2Z^2}{(N-1)e^2 + \sigma^2Z^2}$$

Dónde:

n = Tamaño de la muestra.

Z = Desviación estándar, para un intervalo de confianza de 95.55, es 1.96.

e = Límite aceptable de error muestral que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador.

N = Tamaño de la población.

σ = Desviación estándar de la población que, generalmente cuando no se tiene su valor, suele utilizarse un valor constante de 0,5.

La fórmula del tamaño de la muestra se obtiene de la fórmula para calcular la estimación del intervalo de confianza para la media, la cual es:

$$\bar{X} - Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}} \leq \mu \leq \bar{X} + Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}}$$

De donde el error es:

$$e = Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}}$$

De esta fórmula del error de la estimación del intervalo de confianza para la media se despeja la n, para lo cual se sigue el siguiente proceso:

Elevando al cuadrado a ambos miembros de la fórmula se obtiene:

$$(e)^2 = \left(Z \frac{\sigma}{\sqrt{n}} \sqrt{\frac{N-n}{N-1}} \right)^2$$

$$e^2 = Z^2 \frac{\sigma^2 N - n}{n N - 1}$$

Multiplicando fracciones:

$$e^2 = \frac{Z^2 \sigma^2 (N - n)}{n(N - 1)}$$

Eliminando denominadores:

$$e^2 n(N - 1) = Z^2 \sigma^2 (N - n)$$

Eliminando paréntesis:

$$e^2 n N - e^2 n = Z^2 \sigma^2 N - Z^2 \sigma^2 n$$

Transponiendo n a la izquierda:

$$e^2 n N - e^2 n + Z^2 \sigma^2 n = Z^2 \sigma^2 N$$

Factor común de n:

$$n(e^2 N - e^2 + Z^2 \sigma^2) = Z^2 \sigma^2 N$$

Despejando n:

$$n = \frac{Z^2 \sigma^2 N}{e^2 N - e^2 + Z^2 \sigma^2}$$

Ordenando se obtiene la fórmula para calcular el tamaño de la muestra:

$$n = \frac{Z^2 \sigma^2 N}{e^2 (N - 1) + Z^2 \sigma^2}$$

✚ Ejemplo Ilustrativo

Se tiene $N = 181.175$, para el 95% de confianza $Z = 1,96$, y como no se tiene los demás valores se tomará $\sigma = 0,5$, y $e = 0,05$.

Reemplazando valores de la fórmula se tiene:

$$n = \frac{N \sigma^2 Z^2}{e^2 (N - 1) + \sigma^2 Z^2}$$

$$n = \frac{181.175 \cdot 0,5^2 \cdot 1,96^2}{0,05^2(181.175 - 1) + 0,5^2 \cdot 1,96^2}$$

$$n = \frac{181.175 \cdot 0,5^2 \cdot 1,96^2}{0,05^2(181.175 - 1) + 0,5^2 \cdot 1,96^2} = 383$$

2.4 Factores de la encuesta

La encuesta como método para medir las opciones de un determinado conglomerado social resulta de valor científico para el análisis de información.

Decisiones Preliminares

La encuesta ha sido elaborada con la intención de obtener la información que se desea, se busca el sentido común del entrevistado.

Decisiones sobre el contenido del cuestionario

El cuestionario ha sido elaborado con la intención de que el entrevistado pueda contestarlo, con un vocabulario sencillo. Esta elaborado para que el entrevistado responda en la mayor brevedad posible.

2.5 Análisis e interpretación de resultados

De las 383 encuestas realizadas en el sector urbano y rural de la ciudad de Ibarra, se obtuvieron los siguientes resultados.

Pregunta 1 ¿Cuál es la marca de su celular?

Alternativa	F	%
Nokia	108	28,2
Samsung	92	24,0
LG	39	10,2
BlackBerry	29	7,6
Ericsson	26	6,8
IPhone	24	6,3
Otro	20	5,2
Alcatel	17	4,4
Motorola	15	3,9
HTC	13	3,4
TOTAL	383	100

Tabla 15: Frecuencia correspondiente a la pregunta 1

Fuente:Propia

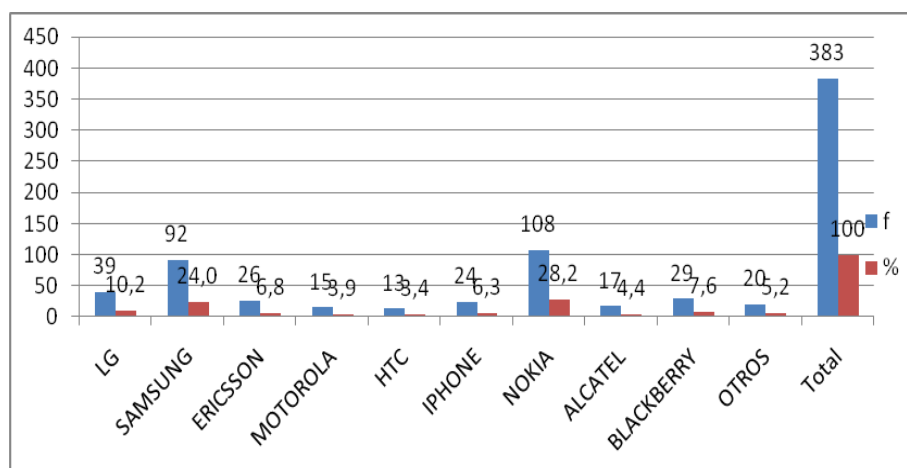


Ilustración 18: Gráfico estadísticos correspondiente a la pregunta 1

Fuente: Propia

La mayoría de los pobladores encuestados en la ciudad de Ibarra, el 28,2%, tienen celulares de marca Nokia, seguidos por Samsung con 24%, con un porcentaje menor LG con un 10,2%, BlackBerry con un porcentaje de 7,6, seguido por Ericsson con un 6,8%. Y con un porcentaje menos significativo el 5,2% de los encuestados poseen otra marca de celular que no es muy comercial; el iPhone posee el 6,3% de la población, seguido con un 4,4% por la marca Alcatel, Motorola con un 3,9% y en último lugar con porcentaje más bajo de 3,4% se encuentra la marca HTC.

Pregunta 2 ¿Cada cuánto tiempo cambia su celular?

Alternativa	F	%
1 año en adelante	299	78,1
6 a 12 meses	66	17,2
1 a 6 meses	18	4,7
TOTAL	383	100

Tabla 16: Frecuencia correspondiente a la pregunta 2

Fuente: Propia

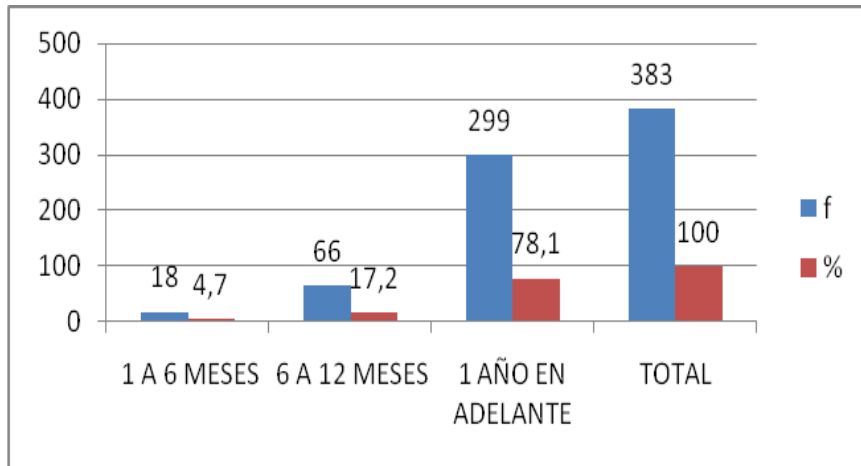


Ilustración 19: Gráfico estadísticos correspondiente a la pregunta 2

Fuente: Propia

La mayoría de los encuestados el 78,1% cambia de celular cada año o más; mientras que con un porcentaje menor el 17,2% cambia su celular cada 6 o 12 meses y tan solo una mínima cantidad de la población el 4,7% cambia su celular de 1 a 6 meses.

Pregunta 3 ¿Su celular tiene acceso a Internet?

Alternativa	F	%
NO	158	41,3
Zonas Wi-fi	137	35,8
Total	88	23
Suma Total	383	100

Tabla 17: Frecuencia correspondiente a la pregunta 3

Fuente: Propia

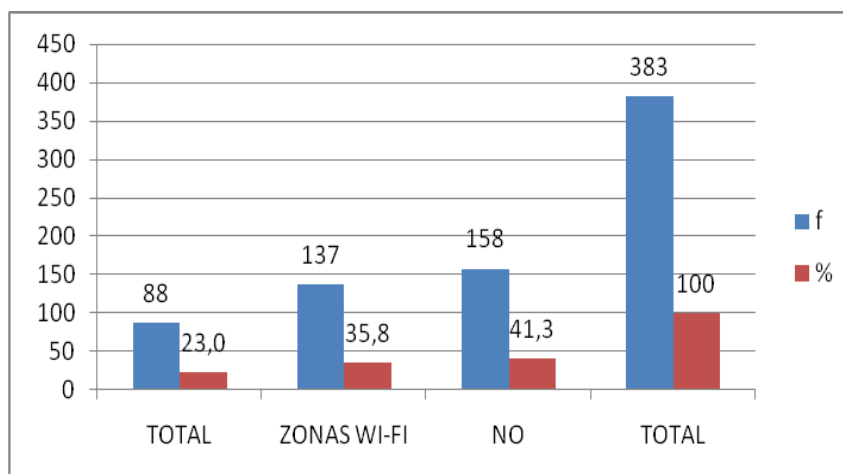


Ilustración 20: Gráfico estadísticos correspondiente a la pregunta 3

Fuente: Propia

La mayoría de la población el 41,3 % no tiene acceso a internet o no posee un paquete de datos en su teléfono, mientras que el 35,8% tiene internet en su celular en las zonas **Wi-fi** y en una menor cantidad de pobladores el 23% tienen un acceso a internet total.

Pregunta 4 ¿Su celular tiene un sistema Operativo **Android**?

Alternativa	F	%
NO	197	51,4
SI	98	25,6
Desconoce	88	23
Total	383	100,0

Tabla 18: Frecuencia correspondiente a la pregunta 4

Fuente: Propia

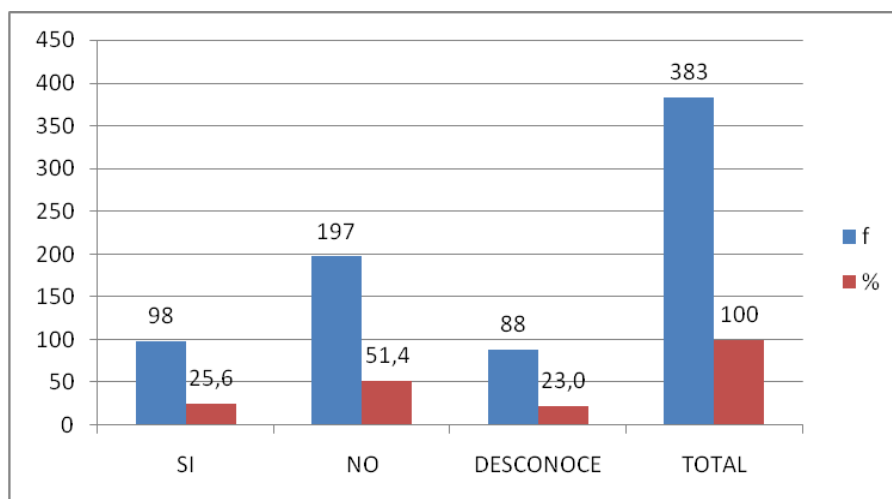


Ilustración 21: Gráfico estadísticos correspondiente a la pregunta 4

Fuente: Propia

La mayoría de la población el 51,4% no tiene un celular con un Sistema Operativo *Android*, cerca de un cuarto de la población encuestada el 25,6% sí cuenta con un Sistema Operativo *Android* en su celular y un menor porcentaje de la población el 23% desconoce si su celular consta de un Sistema Operativo *Android*.

Pregunta 5 ¿Conoce de alguna aplicación móvil, para la promoción de sitios turísticos y atractivos de la ciudad de Ibarra, desarrollada en Ecuador?

Alternativa	F	%
NO	192	50
Desconoce	157	41
SI	34	9
Total	383	100,0

Tabla 19: Frecuencia correspondiente a la pregunta 5

Fuente: Propia

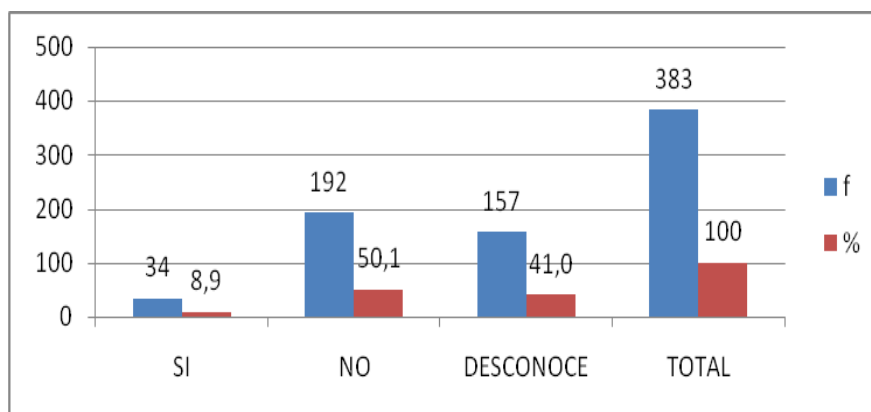


Ilustración 22: Gráfico estadísticos correspondiente a la pregunta 5

Fuente: Propia

La mayoría de personas el 50,1% no conoce de alguna aplicación móvil, para la promoción de sitios turísticos y atractivos de la ciudad de Ibarra, desarrollada en Ecuador; en menor porcentaje se encuentra la población que desconoce del tema con un porcentaje del 41,0%, mientras que un 8,9% de la población sí conoce de una aplicación móvil, para la promoción de sitios turísticos y atractivos de la ciudad de Ibarra.

Pregunta 6 ¿Considerando el precio, como se deberían promocionar los sitios turísticos y atractivos de una ciudad?

Alternativa	F	%
Televisión	121	32
Aplicación móvil	118	31
páginas web	110	29
prensa escrita	20	5
radio emisoras	14	4
TOTAL	383	100

Tabla 20: Frecuencia correspondiente a la pregunta 6

Fuente: Propia

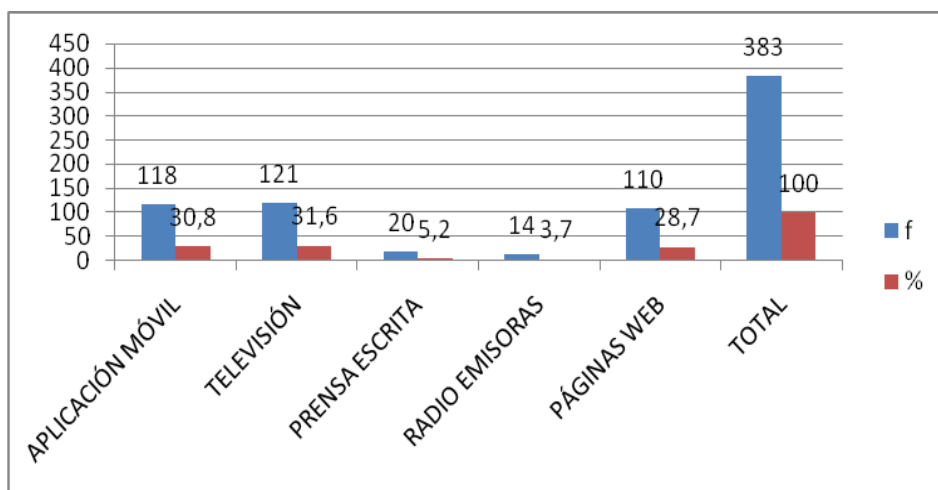


Ilustración 23: Gráfico estadísticos correspondiente a la pregunta 6

Fuente: Propia

Considerando el precio, la mayoría de personas el 31,6% consideran de la televisión, seguido con el 30,8% las aplicaciones móviles, con el 38,7% consideran a los sitios web, mientras que en porcentajes inferiores con 5,2% se encuentran la prensa escrita y radio emisoras con 3,7%.

Pregunta 7 ¿Le gustaría probar una aplicación móvil, para la promoción de sitios turísticos de Ibarra?

Alternativa	F	%
SI	325	85
NO	58	15
Total	383	100,0

Tabla 21: Frecuencia correspondiente a la pregunta 7

Fuente: Propia

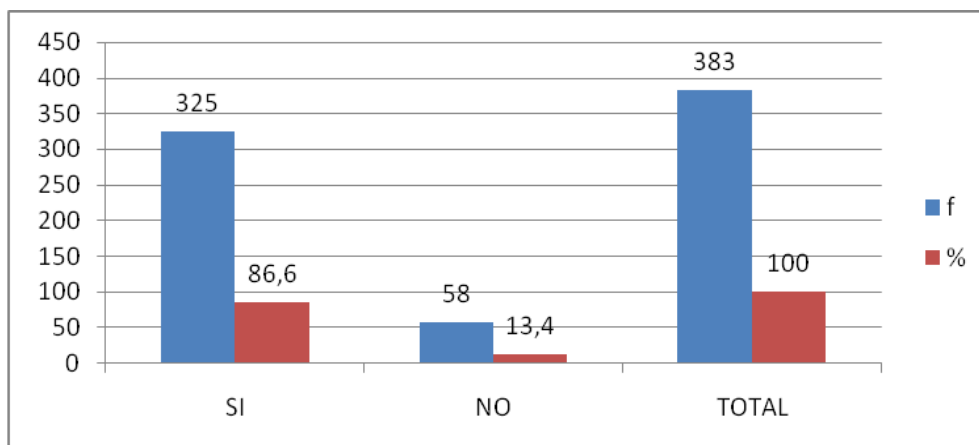


Ilustración 24: Gráfico estadísticos correspondiente a la pregunta 7

Fuente: Propia

A la mayoría de los encuestados de la ciudad de Ibarra, el 86,6% le gustaría probar una aplicación móvil, para la promoción de sitios turísticos de Ibarra y solo a un 13,4% no le gustaría probar una aplicación móvil, para la promoción de sitios turísticos de Ibarra.

2.6 Conclusiones del estudio realizados

De las encuestas realizadas se procede a estimar que las personas sí disponen de un teléfono móvil, con un sistema operativo *Android*, pues *Android* funciona en una gama de dispositivos móviles como *tablets* de las siguientes marcas: *LG*, *Motorola*, *Ericsson*, *HTC*, *Samsung*, lo que aún tiene más alcance.

Además, nadie puede negar que los telefonos como *BlackBerry*, *iPhone* y *nokia* haya marcado un antes y un después en el sector de la telefonía móvil. Sin embargo, si nos fijamos en la tabla 13 (Pág. 34) observamos que tras su introducción en el mercado y su posterior espectacular crecimiento, estos dispositivos **se encuentra actualmente en su etapa de madurez**, la cual se caracteriza por la existencia de un alto número de competidores y por una estabilización y estancamiento de las ventas. En estas circunstancias, una caída de la demanda de dichas marcas es algo más que previsible, pues los servicios que prestan son obsoletos.

CAPÍTULO III

3 Desarrollo de la solución

3.1 Fase de exploración

La aplicación a desarrollarse tiene como propósito ofrecer a los usuarios un gestor web de contenidos, para gestionar información de sitios turísticos, pues en la actualidad existen pocos portales que integran un gestor.

De acuerdo a las historias de usuario iniciales se procede a rediseñar las nuevas historias, que serán la base de los requerimientos del usuario para el equipo de desarrollo XP.

3.1.1 Historias de usuario

1. Acceso y control al sistema

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Acceso y control al sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Jairo Morales	
Descripción: El usuario tendrá acceso a un formulario <i>LOGIN</i> , el cual comprobará, si los datos introducidos corresponden al usuario real; previo ingreso hacia el sistema. La pantalla tendrá las siguientes características. <ul style="list-style-type: none">- Un formulario <i>LOGIN</i>.- Campos de texto con una breve descripción, de la información a ingresarse.- Un campo <i>PASSWORD</i> que mostrará asteriscos en lugar de caracteres. Eventos al presionar el botón Entrar. <ul style="list-style-type: none">- Se verificará que todos los campos de texto estén llenos, caso contrario el sistema mostrará un mensaje de alerta, correspondiente al campo requerido, sin afectar los campos llenos.- El usuario tendrá la opción de dibujar una figura correctamente, que corresponderá al código <i>captcha</i>, para habilitar el botón login y acceder hacia el sistema; Así restringiremos el ingreso de robots. Acciones en el botón Cerrar. <ul style="list-style-type: none">- Cerrará el formulario.	
Observaciones: Deberá ingresarse usuarios iniciales a nivel de base de datos	

Tabla 22: Historia Nro. 1

Fuente: Propia

2. Gestión de usuarios

Historia de Usuario	
Número: 2	Usuario: Cliente
Nombre historia: Gestión de usuarios.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Jairo Morales	
<p>Descripción: El Gestor dispondrá de formularios para ingresar, editar y eliminar la información de una entidad usuario. Los campos que tendrá una entidad usuario son: Nombres, apellidos, mail, usuario, clave, estado, contador de registros sitio, contador de registros gastronomía, contador de registros fotos, contador de registros festivo, contador de registros video, contador de registros historia, serán los que corresponda a un usuario. La variable contador controlará el ingreso de información hacia la base de datos, pues este valor corresponderá al número de registro que el usuario ingrese sobre un sitio turístico.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Cabecera: Tendrá su respectivo nombre, como también una imagen pequeña (X), que al presionarla proceda a cerrar el formulario. - Cuerpo: Tendrá dos columnas, una para los nombres descriptivos de la información, y la siguiente para los campos de texto mencionados. - Pie: Constará de dos botones de color verde y ovalado, uno para almacenar la información, y otro para cancelar. <p>Características generales:</p> <ul style="list-style-type: none"> - Se utilizará colores sencillos para el diseño y maquetado. - Los componentes de formulario estarán centrados. - El usuario podrá ver una breve descripción sobre los campos de texto. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. - El campo correo electrónico tendrá que se ser correcto. 	

Tabla 23: Historia Nro. 2

Fuente: Propia

3. Asignación de roles

Historia de Usuario	
Número: 3	Usuario: Cliente
Nombre historia: Asignación de roles	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1,3	Iteración asignada: 1
Programador responsable: Jairo Morales	
<p>Descripción: El gestor dispondrá de formularios para ingresar, editar y eliminar los campos de una entidad rol. El usuario administrador de roles accederá a un formulario compuesto por un menú usuario, cabecera, cuerpo y pie, para ingresar la información de un rol con los siguientes campos: Nombre y descripción.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú usuario: Procederá a listar de los usuarios. - Cabecera: Tendrá su respectivo nombre, como también una imagen pequeña (X), que al presionarla cerrará el formulario. - Cuerpo: Contendrá dos columnas, una para los nombres descriptivos de la información, y la siguiente para los campos de texto mencionados. - Pie: Constará de dos botones de color verde y ovalado, uno para almacenar la información, y otro para cancelar. <p>Características generales:</p> <ul style="list-style-type: none"> - Un usuario podrá tener varios roles. - El menú usuario desplegará una lista completa de todos los usuarios disponibles. - Los campos de texto tendrán una breve descripción de la información. - Los campos de texto en un formulario tendrán tener una longitud adecuada para la información que será introducida en ellos, si el límite de caracteres establecidos es inferior o superior, debe imprimir un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación, Información ingresada, error, Inténtelo de nuevo. 	
<p>Observaciones: Tendrá que llenarse los campos de su formulario por completo y con exactitud. Si algo no corresponde, escriba <i>NA</i> en los campos de texto requerido. Esto quiere decir "No aplicable".</p>	

Tabla 24: Historia Nro. 3

Fuente: Propia

4. Asignación de responsabilidades

Historia de Usuario	
Número: 4	Usuario: Cliente
Nombre historia: Asignación de responsabilidades	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2,3	Iteración asignada: 1
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad responsabilidad. Los campos a gestionar son: Nombre, descripción. Estructura de los formularios ingresar y editar.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú usuarios: Desplegará una lista de usuarios creados. - Menú roles: Desplegará los roles disponibles del usuario seleccionado en el Menú usuarios. - Cabecera: Tendrá su respectivo nombre, como también una imagen pequeña (X), que al presionarla proceda a cerrar el formulario. - Cuerpo: Tendrá dos columnas, una para los nombres descriptivos de la información, y la siguiente para los campos de texto mencionados. - Pie: Constará de dos botones de color verde y ovalado, uno para almacenar la información, y otro para cancelar. <p>Características generales:</p> <ul style="list-style-type: none"> - Se Utilizará colores sencillos para el diseño y maquetado. - El menú usuarios desplegará la lista completa de todos los usuarios. - El menú roles debe desplegará la lista completa de todos roles que tenga el usuarios. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducida en ellos, si el límite de caracteres establecidos es inferior o superior, debe imprimir un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, error, inténtelo de nuevo. 	
<p>Observaciones: El campo fecha no podrá ser ingresado por el usuario, ya que es de tipo <i>TIMESTAMP</i>, y debe funcionar a modo de auditoria.</p>	

Tabla 25: Historia Nro. 4

Fuente: Propia

5. Asignación de permisos hacia pantallas

Historia de Usuario	
Número: 5	Usuario: Cliente
Nombre historia: Asignación de permisos hacia pantallas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2,3	Iteración asignada: 1
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad pantalla.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú usuario: Desplegará los usuarios creados y disponibles. - Menú roles: Desplegará los roles disponibles del usuario seleccionado en el menú usuario. - Menú responsabilidades: Desplegará las responsabilidades disponibles del rol seleccionado en el menú roles. - Cabecera: Constará de su respectivo nombre, como también una imagen pequeña (X), que al presionarla proceda a cerrar el formulario. - Cuerpo: Imprimirá todas las pantallas disponibles, además un botón Guardar y editar. <p>Características generales:</p> <ul style="list-style-type: none"> - Utilizar colores degradados en los estilos de los formularios. - Se Utilizará colores sencillos para el diseño y maquetado. - Los componentes de formulario estarán centrados. - El cliente podrá visualizar un reporte completo de las pantallas disponibles, para poder seleccionarlas, y asignarlas hacia un usuario creado. - Almacenará todas las pantallas seleccionadas, para un respectivo usuario, rol y responsabilidad. - Editará las pantallas seleccionadas 	
<p>Observaciones: Las pantallas de administración tendrán que ser ingresadas previamente por el usuario, mediante un script SQL, en la base de datos, al momento de crear la base de datos.</p>	

Tabla 26: Historia Nro. 5

Fuente: Propia

6. Gestión de provincias

Historia de Usuario	
Número: 6	Usuario: Cliente
Nombre historia: Gestión de provincias.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad provincia.</p> <p>Los campos a gestionar son: Nombre, descripción, capital, población, región natural.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Cabecera: Tendrá su respectivo nombre, como también una imagen pequeña (X), que al presionarla proceda a cerrar el formulario. - Cuerpo: Contendrá dos columnas, una para los nombres descriptivos de la información, y la siguiente para los campos de texto mencionados. - Pie: Constará de dos botones de color verde y ovalado, uno para almacenar la información, y otro para cancelar. <p>Características generales:</p> <ul style="list-style-type: none"> - Se Utilizará colores sencillos para el diseño y maquetado. - Se Centrarán los componentes del formulario. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducida en ellos, si el límite de caracteres establecidos es inferior o superior, debe imprimir un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, Se imprimirá un mensaje de texto en donde especifique el estado de la operación, Información ingresada, error, Inténtelo de nuevo. - Al ingresar letras en campos, que especifiquen solo números, se imprimirá un mensaje de texto, correspondiente. Ejemplo: "Ingresar solo números". 	
<p>Observaciones:</p> <ul style="list-style-type: none"> - Llene los campos de su formulario por completo y con exactitud. Si algo no corresponde, escriba "No aplicable". 	

Tabla 27: Historia Nro. 6

Fuente: Propia

7. Gestión de ciudades

Historia de Usuario	
Número: 7	Usuario: Cliente
Nombre historia: Gestión de ciudades.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1,2	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad ciudad.</p> <p>Campos a gestionar: Nombre, descripción y población.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincias: Desplegará las provincias disponibles. - Cabecera: Tendrá su respectivo nombre, como también una imagen pequeña (X), que al presionarla proceda a cerrar el formulario. - Cuerpo: Dos columnas, una para los nombres descriptivos de la información, y la siguiente para los campos de texto mencionados. - Pie: Constará de dos botones de color verde y ovalado, uno para almacenar la información, y otro para cancelar. <p>Características generales.</p> <ul style="list-style-type: none"> - Se Utilizará colores sencillos para el diseño y maquetado. - Se Centrarán los componentes de formulario. - Utilizar menús actuales. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducida en ellos, si el límite de caracteres establecidos es inferior o superior, debe imprimir un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, se imprimirá un mensaje de texto en donde especifique el estado de la operación, Información ingresada, error, Inténtelo de nuevo. 	
<p>Observaciones: Llene los campos de su formulario por completo y con exactitud. Si algo no corresponde, escriba "No aplicable".</p>	

Tabla 28: Historia Nro. 7

Fuente: Propia

8. Gestión de categorías

Historia de Usuario	
Número: 8	Usuario: Cliente
Nombre historia: Gestión de categorías.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1,1	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El usuario administrador de categorías accederá hacia un formulario compuesto por una cabecera, cuerpo y pie. Los campos a gestionar son: Nombre y descripción. Los nombres de las categorías pueden corresponderán a sitios y atractivos turísticos.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Cabecera: Tendrá su respectivo nombre, como también una imagen pequeña (X), que al presionarla proceda a cerrar el formulario. - Cuerpo: Dos columnas, una para los nombres descriptivos de la información, y la siguiente para los campos de texto mencionados. - Pie: Constará de dos botones de color verde y ovalado, uno para almacenar la información, y otro para cancelar. <p>Características generales:</p> <ul style="list-style-type: none"> - Se Utilizará colores sencillos para el diseño y maquetado. - Se Centrarán los componentes de formulario. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducida en ellos, si el límite de caracteres establecidos es inferior o superior, debe imprimir un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, Se imprimirá un mensaje de texto en donde especifique el estado de la operación, Información ingresada, error, Inténtelo de nuevo. - Al ingresar letras en campos, que especifiquen solo números, se imprimirá un mensaje de texto, correspondiente. Ejemplo: "Ingresar solo números". 	
<p>Observaciones: Las categorías pueden corresponder a un hotel, hostería, restaurantes o parques temáticos.</p>	

Tabla 29: Historia Nro. 8

Fuente: Propia

9. Gestión de sitios

Historia de Usuario	
Número: 9	Usuario: Cliente
Nombre historia: Gestión de sitios	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2,2	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El usuario administrador de sitios accederá hacia a un formulario para ingresar la información de un sitio que corresponda a una categoría seleccionada: Hostería, monumentos, centros educativos, sitios turísticos, destino turísticos y entre otros.</p> <p>Los campos del sitio a ingresar son: Nombre, descripción, página web, mail, nombre página facebook, nombre cuenta twitter, dirección, teléfono, latitud, longitud.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincia: Listará las provincias. - Menú ciudades: Listará las ciudades de una provincia seleccionada. - Menú categoría: Listará las categorías disponibles. - Visor de mapas: Permitirá elegir vistas de mapas en línea. - Botones: Guardar y cerrar, con su respectivo nombre. <p>Características generales:</p> <ul style="list-style-type: none"> - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. - La posición seleccionada en el mapa, será almacenada. <p>Observaciones: Su ubicación Geo-referenciada se almacenará, utilizando el código de proyección EPSG: 4326, pues es un sistema de <i>coordenadas geográficas (WGS84)</i> que nos permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia).</p>	

Tabla 30: Historia Nro. 9

Fuente: Propia

10. Gestión de historias

Historia de Usuario	
Número: 10	Usuario: Cliente
Nombre historia: Gestión de historias	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad historia o evento histórico.</p> <p>El usuario administrador de historias accederá hacia a un formulario para ingresar la información de una historia que corresponda a un sitio seleccionado.</p> <p>Los campos a ingresar son: Nombre y descripción.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincia: Listará las provincias. - Menú ciudades: Listará las ciudades de una provincia seleccionada. - Menú sitios: Listará los sitios disponibles, que corresponden a un usuario único. <p>Características Generales:</p> <ul style="list-style-type: none"> - El formulario tendrá tener un nombre. - Un botón guardar y otro cerrar. - los componentes de formulario serán centrados. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. 	

Tabla 31: Historia Nro. 10

Fuente: Propia

11. Gestión de URL'S hacia YouTube

Historia de Usuario	
Número: 11	Usuario: Cliente
Nombre historia: Gestión de URL'S hacia YouTube	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1,2	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad video.</p> <p>El usuario administrador de <i>URL'S</i> hacia <i>YouTube</i> accederá a un formulario para ingresar la información de direcciones que corresponda a un sitio seleccionado. Los campos a ingresar son: Nombre, descripción y <i>URL</i> video.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincia: Listará las provincias. - Menú ciudades: Listará las ciudades de una provincia seleccionada. - Menú sitios: Listará los sitios disponibles, que corresponden a un usuario único. <p>Características generales:</p> <ul style="list-style-type: none"> - Los formularios tendrá tener un nombre. - Un botón guardar y otro cerrar. - los componentes de formulario serán centrados. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. 	

Tabla 32: Historia Nro. 11

Fuente: Propia

12. Gestión de imágenes

Historia de Usuario	
Número: 12	Usuario: Cliente
Nombre historia: Gestión de imágenes.	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1,1	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad foto. El usuario administrador de imágenes accederá a un formulario para ingresar los siguientes campos: Nombre, descripción e imagen.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincia: Listará las provincias. - Menú ciudades: Listará las ciudades de una provincia seleccionada. - Menú sitios: Listará los sitios disponibles, que corresponden a un usuario único. <p>Características generales.</p> <ul style="list-style-type: none"> - Los formularios tendrá tener un nombre. - Un botón guardar y otro cerrar. - los componentes de formulario serán centrados. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> - El tamaño de la imagen deberá ser menor a 200 KB - Dimensión de (100 a 500) PX ancho (100 a 500) PX alto - Formato PNG y JPEG. 	

Tabla 33: Historia Nro. 12

Fuente: Propia

13. Gestión de festivos

Historia de Usuario	
Número: 13	Usuario: Cliente
Nombre historia: Gestión de festivos	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1,1	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad festivo. El usuario administrador de imágenes accederá a un formulario para ingresar los siguientes campos: Nombre, descripción, foto, fecha inicial y fecha final.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincia: Listará todas las provincias disponibles. - Menú ciudades: Listará las ciudades de una provincia seleccionada. - Menú sitios: Listará los sitios disponibles, que corresponden a un usuario único. <p>Características generales:</p> <ul style="list-style-type: none"> - Los formularios tendrá tener un nombre. - Un botón guardar y otro cerrar. - los componentes de formulario serán centrados. - El formulario festivo tendrá calendarios desplegados, para poder seleccionar una fecha inicia y final. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. 	
<p>Observaciones:</p> <ul style="list-style-type: none"> - El tamaño de la imagen deberá ser menor a 200 KB - Dimensión de (100 a 500) PX ancho (100 a 500) PX alto - Formato PNG y JPEG. 	

Tabla 34: Historia Nro. 13

Fuente: Propia

14. Gestión de gastronomías

Historia de Usuario	
Número: 14	Usuario: Cliente
Nombre historia: Gestión de gastronomías	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1,1	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El gestor presentará formularios para ingresar, editar y eliminar los campos de una entidad gastronomía.</p> <p>El usuario administrador de gastronomías accederá a un formulario para ingresar los siguientes campos: Nombre, descripción e imagen.</p> <p>Estructura y funcionalidades de los formularios:</p> <ul style="list-style-type: none"> - Menú provincia: Listará las provincias. - Menú ciudades: Listará las ciudades de una provincia seleccionada. - Menú sitios: Listará los sitios disponibles, que corresponden a un usuario único. <p>Características generales:</p> <ul style="list-style-type: none"> - Los formularios tendrá tener un nombre. - Un botón guardar y otro cerrar. - los componentes de formulario serán centrados. - Los campos de texto dispondrán de una breve descripción de la información que será introducida. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. <p>Observaciones:</p> <ul style="list-style-type: none"> - El tamaño de la imagen deberá ser menor a 200 KB - Dimensión de (100 a 500) PX ancho (100 a 500) PX alto - Formato PNG y JPEG. 	

Tabla 35: Historia Nro. 14

Fuente: Propia

15. Gestión de rutas

Historia de Usuario	
Número: 15	Usuario: Cliente
Nombre historia: Gestión de rutas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1,4	Iteración asignada: 2
Programador responsable: Jairo Morales	
<p>Descripción: El usuario accederá a un formulario, para ingresar la información, con los siguientes campos: nombre, descripción y una ruta respectiva.</p> <p>Visor de mapas: Permitirá seleccionar el tipo proveedor de mapas como: <i>Google, Mapquest u Open Streetmaps</i>, además graficar y almacenar una ruta, respecto a un sistema geo-referenciado.</p> <p>Características generales:</p> <ul style="list-style-type: none"> - El usuario podrá graficar una ruta. - El usuario elegirá su proveedor de mapas en línea. - Se Insertará una breve descripción de la información a introducirse en los campos de texto. - Los campos de texto en un formulario tendrán una longitud adecuada para la información que será introducirá en ellos, si el límite de caracteres establecidos es inferior o superior, imprimirá un mensaje de alerta moderado, con la siguiente información. Ejemplo: "Por favor ingrese entre 6 y 15 caracteres". - Se comprobará si la dirección de correo electrónico está bien escrita o no. - Se comprobará que se haya suministrado todos los campos obligatorios, caso contrario el sistemas mostrará un mensaje de alerta: "Este campo es requerido", sin afectar los campos llenos. - Si todas las validaciones fueron aprobadas, imprimirá un mensaje de texto en donde especifique el estado de la operación: Información ingresada, Error, Inténtelo de nuevo. - La posición seleccionada en el mapa, será almacenada. - Se procederá a cerrar el respectivo formulario. - 	
<p>Observaciones: Su ruta geo-referenciada se almacenará, utilizando el código de proyección <i>EPSG: 4326</i>, pues es un sistema de <i>coordenadas geográficas (WGS84)</i> que nos permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia).</p>	

Tabla 36: Historia Nro. 15

Fuente: Propia

16. Reporte Plano

Historia de Usuario	
Número: 16	Usuario: Cliente
Nombre historia: Reporte plano de Sitios.	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Jairo Morales	
<p>Descripción: Todos los usuario sin importar el tipo de privilegio, Accederán hacia una interfaz web, que le permitirá seleccionar una provincia y ciudad; posteriormente ingresar un tipo de categoría, la cual se autocompletará sin necesidad de escribirla por completo.</p> <p>Estructura y funcionalidades de los formularios:</p> <p>Menú provincias: Listará todas las provincias. Menú ciudades: Listará las ciudades de una provincia seleccionada. Campo de texto: Ingreso de las categorías disponibles, sobre la cual se efectuará la búsqueda.</p> <p>Al presionar sobre un botón texto, se imprimirá un reporte de todos los sitios encontrados respecto a los parámetros seleccionados e ingresados. Se imprimirá la siguiente información de un sitio: Nombre, web, mail, teléfono, facebook, twitter, dirección.</p>	
<p>Observaciones: Cada sitio tendrá enlaces hacia historias, enlaces video YouTube, imágenes, festivales, gastronomías, rutas y localización.</p>	

Tabla 37: Historia Nro. 16

Fuente: Propia

17. Reporte geo-referenciado

Historia de Usuario	
Número: 17	Usuario: Cliente
Nombre historia: Reporte geo-referenciado.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Jairo Morales	
<p>Descripción: Los usuarios de la red accederán hacia una pantalla, para buscar información sobre sitios y destinos turísticos.</p> <p>Estructura y funcionalidades de los formularios:</p> <p>Menú provincias: Listará todas las provincias. Menú ciudades: Listará las ciudades de una provincia seleccionada. Campo de texto: Ingreso de las categorías disponibles, sobre la cual se efectuará la búsqueda.</p> <p>Visor de mapas: Imprimirá marcadores de los sitios encontrados, sobre mapas de <i>Google</i>, <i>Mapquest</i> y <i>Open Streetmaps</i>.</p> <p>Al presionar sobre un botón mapa, permitirá buscar la información ingresada por teclado, la búsqueda debe efectuarse dentro de la provincia y ciudad previamente seleccionada, los marcadores sobre el mapa permitirán presentar la siguiente información de un sitio: Nombre, latitud, longitud, web, mail, <i>twitter</i>, <i>facebook</i>, dirección y teléfono.</p>	
<p>Observaciones: Deberá obligatoriamente presionar sobre los marcadores para consultar información adicional sobre un sitio.</p>	

Tabla 38: Historia Nro. 17

Fuente: Propia

18. Enlaces hacia historias

Historia de Usuario	
Número: 18	Usuario: Cliente
Nombre historia: Enlaces hacia historias	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Jairo Morales	
<p>Descripción: Los usuario que efectúen una búsqueda, tendrán la posibilidad de acceder hacia reportes de las historias ingresadas.</p> <p>El reporte tendrá la información de: Provincia, ciudad, sitio, categorías como también el nombre y descripción de historia.</p>	

Tabla 39: Historia Nro. 18

Fuente: Propia

19. Enlaces hacia YouTube

Historia de Usuario	
Número: 19	Usuario: Cliente
Nombre historia: Enlaces hacia YouTube	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Jairo Morales	
<p>Descripción: Los usuario que efectúen una búsqueda, tendrá la posibilidad de acceder hacia reportes de los videos y visualizar los videos ingresados en <i>YouTube</i>.</p> <p>El reporte tendrá la información de: Provincia, ciudad, sitio, categorías como también el nombre y descripción y <i>URL</i> de video.</p>	

Tabla 40: Historia Nro. 19

Fuente: Propia

20. Enlaces hacia imágenes

Historia de Usuario	
Número: 20	Usuario: Cliente
Nombre historia: Enlaces hacia imágenes	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1,2	Iteración asignada: 3
Programador responsable: Jairo Morales	
Descripción: Los usuario que efectúen una búsqueda, tendrán la posibilidad de acceder hacia reportes de las imágenes ingresadas. El reporte tendrá la información de: Provincia, ciudad, sitio, categorías. Las imágenes aparecerán en un carrusel dinámico, con opciones atrás y adelante.	

Tabla 41: Historia Nro. 20

Fuente: Propia

21. Enlaces hacia festivos

Historia de Usuario	
Número: 21	Usuario: Cliente
Nombre historia: Enlaces hacia festivos	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 0,3	Iteración asignada: 3
Programador responsable: Jairo Morales	
Descripción: Los usuario que efectúen una búsqueda, tendrán la posibilidad de acceder hacia reportes de los días festivos ingresadas. El reporte tendrá la información de: Provincia, ciudad, sitio, categorías como también el nombre, descripción, fecha inicia y final de los días festivos. Las imágenes aparecerán en un carrusel dinámico, con opciones atrás y adelante.	

Tabla 42: Historia Nro. 21

Fuente: Propia

22. Enlaces hacia gastronomías

Historia de Usuario	
Número: 22	Usuario: Cliente
Nombre historia: Enlaces hacia gastronomías	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Jairo Morales	
<p>Descripción: Los usuario que efectúen una búsqueda, tendrán la posibilidad de acceder hacia reportes de las gastronomías ingresadas.</p> <p>El reporte tendrá la información de: Provincia, ciudad, sitio y categorías. Las imágenes aparecerán en un carrusel dinámico, con opciones atrás y adelante.</p>	

Tabla 43: Historia Nro. 22

Fuente: Propia

23. Enlaces hacia rutas

Historia de Usuario	
Número: 23	Usuario: Cliente
Nombre historia: Enlaces hacia rutas	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1,2	Iteración asignada: 3
Programador responsable: Jairo Morales	
<p>Descripción: Los usuario que efectúen una búsqueda, tendrán la posibilidad de acceder hacia reportes de las rutas ingresadas.</p> <p>El reporte tendrá la información de: Provincia, ciudad, sitio y categorías. El usuario dispondrá de un listado de las rutas encontradas, un marcador de color verde indicara el inicio de la ruta y un marcador rojo el final.</p>	

Tabla 44: Historia Nro. 23

Fuente: Propia

24. Reportes plano de sitios (App móvil)

Historia de Usuario	
Número: 24	Usuario: Cliente
Nombre historia: Reporte plano de sitios (App móvil)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Jairo Morales	
Descripción: <p>Todos los usuarios que dispongan de un dispositivo con un SSOO <i>Android</i>, podrán instalar el Aplicativo disponible en la tienda Google Play.</p> <p>Una vez instalada la aplicación móvil podrán acceder a los reportes; es necesaria una conexión a internet.</p> <p>Menú provincias: Listará un menú con todas las provincias.</p> <p>Menú ciudades: Listará las ciudades de una provincia seleccionada.</p> <p>Menú categorías: Listará todas las categorías disponibles.</p> <p>Al presionar sobre un botón buscar, se accederá a un servicio web que permitirá visualizar la información de un sitio con los siguientes campos: Nombre, web, mail, teléfono, página facebook, twitter y dirección.</p>	

Tabla 45: Historia Nro. 24

Fuente: Propia

25. Reporte de historias y videos (App móvil)

Historia de Usuario	
Número: 25	Usuario: Cliente
Nombre historia: Reporte de historias y videos (App móvil)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1,2	Iteración asignada: 3
Programador responsable: Jairo Morales	
Descripción: <p>Los usuarios podrán acceder a reportes asociados con un sitio, previamente seleccionando de los menús respectivos:</p> <p>Menú provincias: Listará un menú con todas las provincias.</p> <p>Menú ciudades: Listará las ciudades de una provincia seleccionada.</p> <p>Menú categorías: Listará todas las categorías disponibles.</p> <p>La información plana asociada a mostrar es la siguiente: Historias y Videos.</p>	

Tabla 46: Historia Nro. 25

Fuente: Propia

26. Reporte de festivos, fotos y gastronomías (App móvil)

Historia de Usuario	
Número: 26	Usuario: Cliente
Nombre historia: Reporte de festivos, fotos y gastronomías (App móvil).	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 1,2	Iteración asignada: 3
Programador responsable: Jairo Morales	
Descripción: Los usuarios podrán acceder a reportes asociados con un sitio, previamente seleccionando de los menús respectivos: Menú provincias: Listará un menú con todas las provincias. Menú ciudades: Listará las ciudades de una provincia seleccionada. Menú categorías: Listará todas las categorías disponibles. La información plana y visual asociada es la siguiente: Festivos, Fotos y Gastronomía.	

Tabla 47: Historia Nro. 26

Fuente: Propia

3.1.2 Modelo Vista Controlador

De cara a afrontar el desarrollo de la Aplicación Web, se establece un modelo o arquitectura que permita estructurar las capas de manera que puedan interactuar y alcanzar un mayor rendimiento.

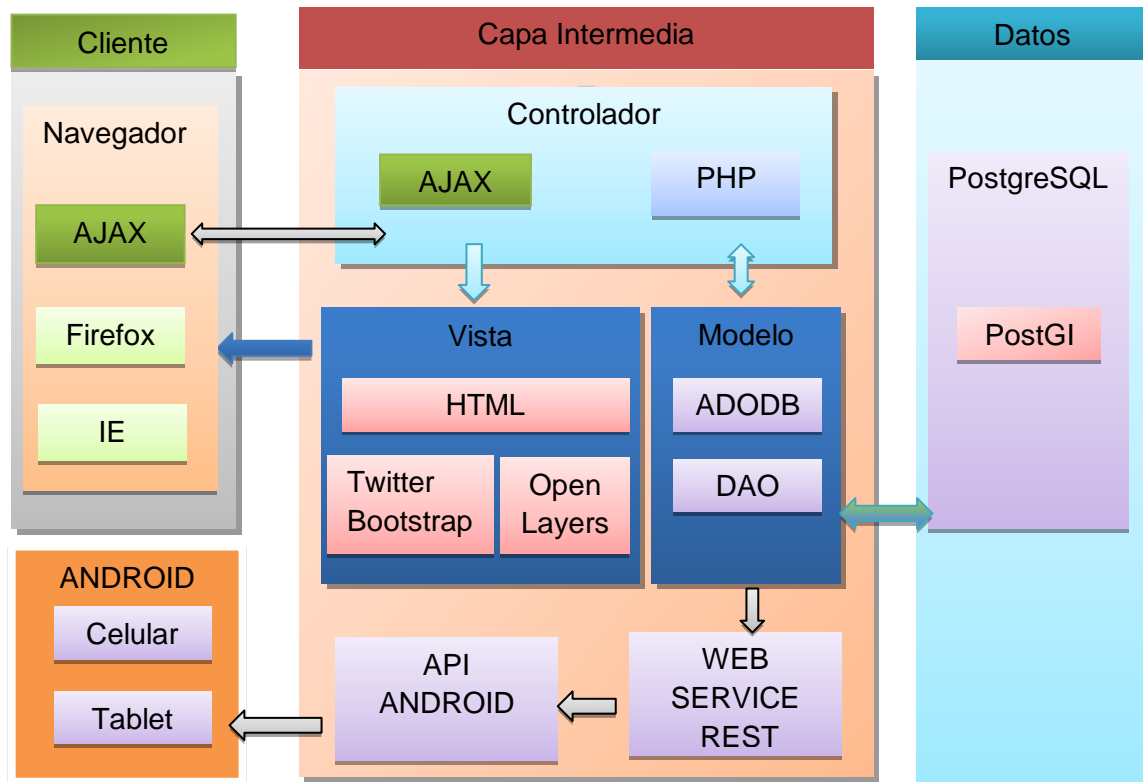


Ilustración 25: Arquitectura del sistema

Fuente: Propia

✚ El controlador

El controlador es el motor de aplicación, pues gestiona todas las peticiones en la capa intermedia que se realizan desde el cliente, además invoca al resto de componentes de la aplicación,

Por ejemplo en el caso de que una petición requiera enviar como respuesta al cliente determinada información existente en la base de datos, el controlador solicitará los datos necesarios al modelo, y una vez recibidos, se los proporcionará a la vista para que esta les aplique el formato de presentación correspondiente y les envíe la respuesta al cliente.

🚦 La vista

Como su nombre lo indica, la vista es la encargada de generar respuestas, habitualmente archivos *PHP*, *HTML* ó *XHTML*, que son enviadas al cliente, cuando esta respuesta tiene que incluir datos proporcionados por el controlador, las etiquetas *HTML* deberán ser dinámicas y atractivas, pues el usuario debe estar a gusto con el diseño y maquetado de la interfaz web.

En la vista es donde *AJAX* juega un papel muy importante, tanto las páginas *PHP* o *HTML* pueden incluir código script de cliente que se comuniquen en segundo plano con el servidor.

En este caso una vez que se haya generado las respuestas hacia el cliente, las peticiones realizadas al servidor desde el código *AJAX* serán dirigidas al controlador, de este modo cuando *PHP* recibe la petición desde una aplicación *AJAX* cliente, recuperará los datos necesarios a través de modelo, en vez de encaminar la petición a la vista para que vuelva a generar la respuesta, lo que supondría una recarga de la página, aplicará a los datos el formato correspondiente (texto plano *XML*, *JSON*) y los enviará directamente a la página que hizo la petición.

🚦 El Modelo

En la arquitectura MVC, la lógica de negocio de la aplicación, incluyendo el acceso a los datos y su manipulación, está encapsulado dentro del modelo, en una aplicación *PHP*, el modelo puede ser implementado mediante la herramienta *ADODB5*, para gestionar las conexiones de una manera más segura y rápida utilizando el concepto de *DAO*⁷.

3.1.3 Arquitectura funcional del sistema

Todas las herramientas que se presentan en la arquitectura, son basadas en *Open Source* y tienen licencia de libre distribución.

Es importante mencionar que el aplicativo web, será el encargado de generar toda la información turística hacia el aplicativo móvil alojado en la tienda virtual de Google Play, y que puede ser descargado por el usuario sin ningún tipo de restricción.

Una vez instalado el APP móvil sobre un dispositivo físico y una conexión a la red, se procederá a consumir los servicios web *REST* generados desde *PHP*, utilizando el estándar *JSON*, que es el encargado de almacenar y transferir la información.

⁷ Data Access Object: Patrón de diseño para acceder a la información de una base de datos.

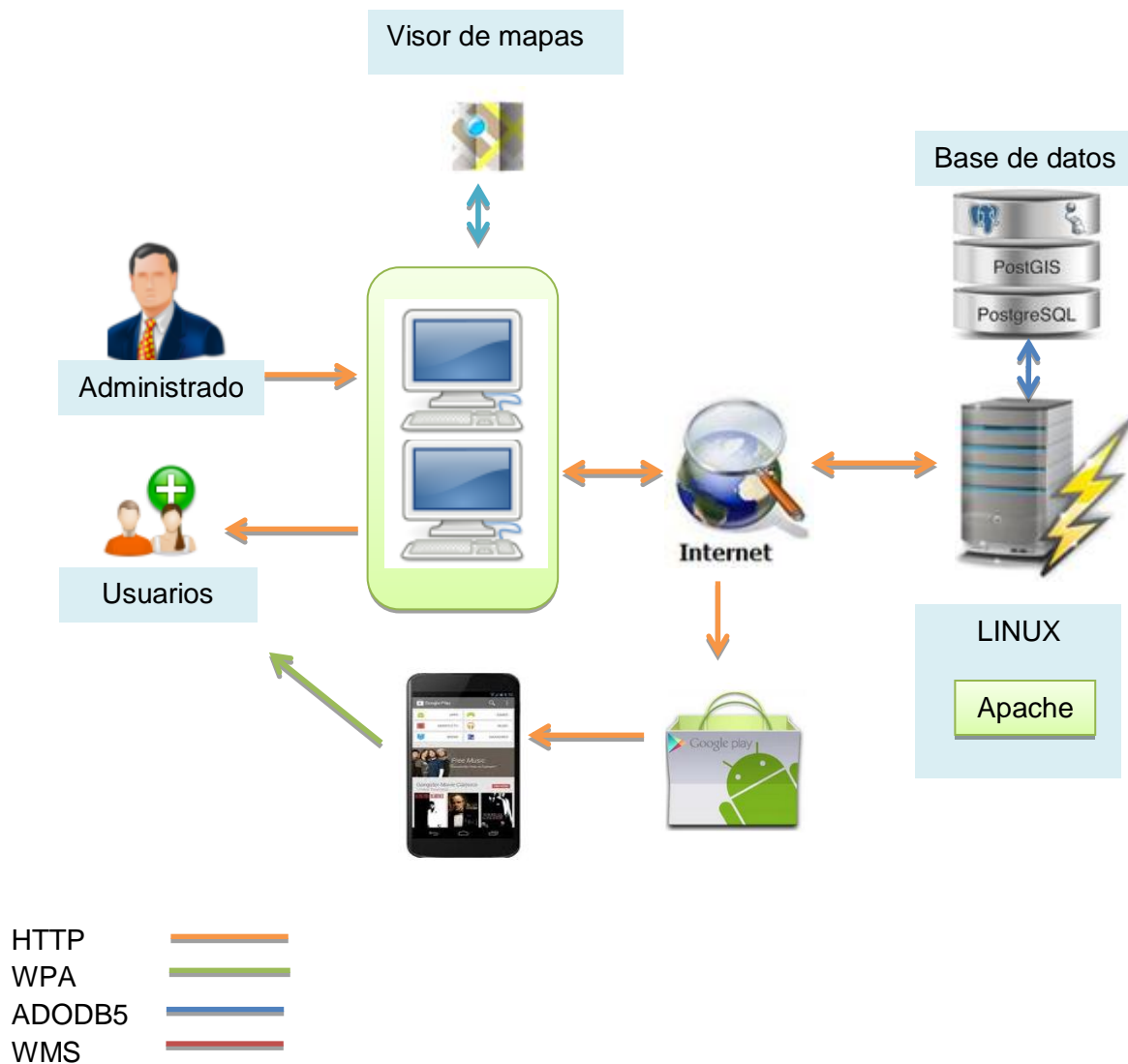


Ilustración 26: Arquitectura funcional del sistema

Fuente: Propia

3.2 Fase de planeación

3.2.1 Cronograma de actividades

En base a las historias de usuario, se procedió a ordenarlas y en consecuencia se obtuvo los módulos del sistema, además se establece un cronograma de actividades; los cuales deberán realizarse durante tres iteraciones.

Módulo	N.	Historias de usuario	Fecha Estimada	Esfuerzo de desarrollo		
				Semanas Ideales	Días Ideales	Horas Ideales
1	1	Acceso y control al sistema	03/11/12 07/11/12	1	5	40
	2	Gestión de usuarios	09/11/12 18/11/12	2	10	71
	3	Asignación de roles	22/11/12 29/11/12	1,3	8	62
	4	Asignación de responsabilidades	03/12/12 15/12/12	2,3	13	83
	5	Asignación de permisos Hacia pantallas	17/12/12 29/12/12	2,3	13	83
2	6	Gestión de provincias	02/1/13 06/1/13	1	5	40
	7	Gestión de ciudades	08/1/13 14/1/13	1,2	7	56
	8	Gestión de categorías	18/1/13 23/1/13	1,1	6	48
	9	Gestión de sitios	25/1/13 05/2/13	2,2	12	90
	10	Gestión de historias	08/2/13 17/2/13	2	10	71
	11	Gestión de URL'S hacia YouTube	21/2/13 27/2/13	1,2	7	56
	12	Gestión de imágenes	02/3/13 07/3/13	1,1	6	48
	13	Gestión de festivos	10/3/13 15/3/13	1,1	6	48
	14	Gestión de gastronomías	16/3/13 21/3/13	1,1	6	48
	15	Gestión de rutas	22/3/13 30/3/13	1,4	9	67
3	16	Reporte plano de Sitios	03/4/13 07/4/13	1	5	45
	17	Reporte geo-referenciado	09/4/13 18/4/13	2	10	71
	18	Enlaces hacia historias	20/4/13 24/4/13	1	5	40
	19	Enlaces hacia YouTube	25/4/13 29/4/13	1	5	40
	20	Enlaces hacia imágenes	03/5/13 09/5/13	1,2	7	56
	21	Enlaces hacia festivos	12/5/13 14/5/13	0,3	3	30
	22	Enlaces hacia Gastronomías	15/5/13 19/5/13	1	5	40
	23	Enlaces hacia rutas	20/5/13 26/5/13	1,2	7	56
	24	Reporte plano de sitios (App móvil)	05/6/13 14/6/13	2	10	71
	25	Reporte de historias y videos (App móvil)	15/6/13 19/6/13	1,2	7	50
	26	Reporte de festivos, fotos y gastronomías (App móvil)	21/6/13 28/6/13	1,2	7	45

Tabla 48: Cronograma de actividades

Fuente: Propia

3.2.2 Módulos de sistema

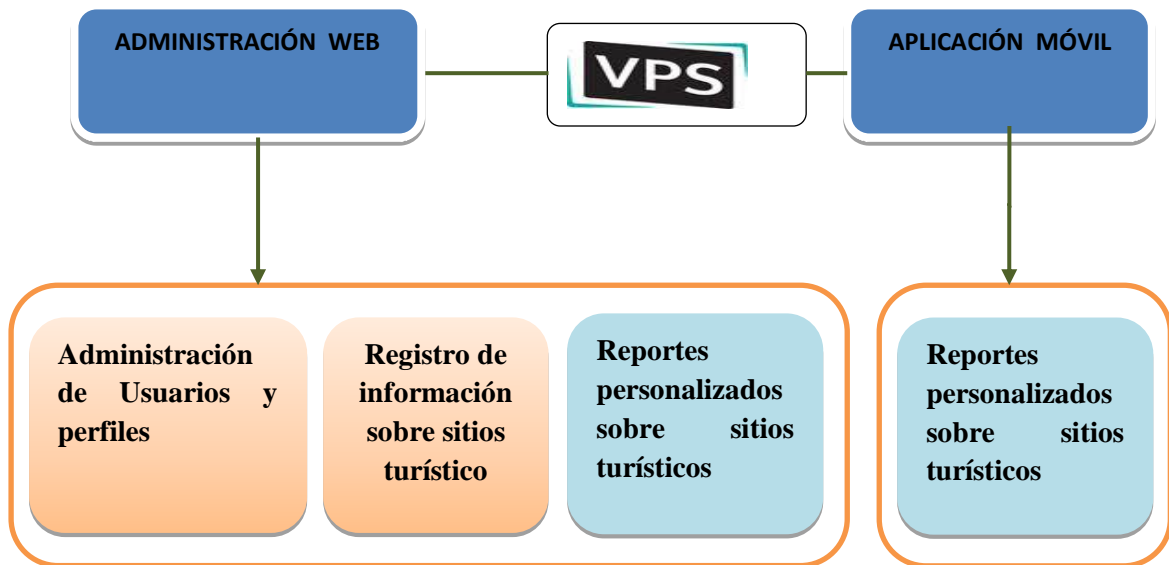


Ilustración 27: Módulos del sistema

Fuente: Propia

Descripción de los módulos

- **Administración de usuarios y perfiles**

El sistema web de Turismo se integrara dentro del modelo de autenticación de la plataforma.

De esta forma, los usuarios que necesiten usar la aplicación web deberán autenticarse en el sistema utilizando un usuario que disponga de suficientes permisos para poder realizar las operaciones deseadas.

La aplicación brindará acceso a una pantalla inicial (formulario) donde el usuario administrador deberá consignar el *login/password* a un cliente, además solo el administrador brindara los permisos hacia las pantallas de administración disponibles en el sistema de manera conveniente.

- **Registro de información sobre sitios turísticos**

Será el encargado de almacenar la información generada desde la aplicación web y que constituirá la base de los sitios turísticos. El módulo recibirá toda la información plana y geográfica y la almacenará en la base de datos del VPS (Servidor privado Virtual), para su posterior utilización.

Este módulo brindara una interfaz gráfica que permitirá a los usuarios acceder hacia varias pantallas y formularios para publicar la información de interés.

- **Reportes personalizados sobre sitios turísticos**

Presentará información vital de los atractivos y sitios turísticos ingresados por el usuario, en forma de reportes relativa a la entidad solicitada por el usuario, por ejemplo un sitio turístico y sus características.

Para la visualización de la información, el sistema utilizará los valores definidos por los usuarios.

3.3 Iteración I

3.3.1 Cronograma de actividades

ITE.	Historias de usuario	Fecha	Duración en Semanas	Duración en horas
1	Historia 1 Acceso y control al sistema	03/11/12 07/11/12	1	40
1	Historia 2 Gestión de usuarios	09/11/12 18/11/12	2	71
1	Historia 3 Asignación de roles	22/11/12 29/11/12	1,3	62
	Historia 4 Asignación de Responsabilidades	03/12/12 15/12/12	2,3	83
1	Historia 5 Asignación de permisos hacia Pantallas	17/12/12 29/12/12	2,3	83
				Total: 339

Tabla 49: Cronograma de actividades, Iteración I

Fuente: Propia

3.3.2 Tareas por historia Iteración I

- Acceso al sistema

Historia 1: Acceso al sistema.		
Número	Nombre	Tiempo estimado
1	Configuración de un entorno de desarrollo web.	8 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
3	Diseño de interfaz gráfica <i>login</i> .	8 horas
4	<i>MAPEO OBJETO-RELACIONAL</i> entidad usuario.	8 horas
5	Creación de archivos <i>PHP</i> .	8 horas
		40 Horas

Tabla 50: Tareas de acceso al sistema

Fuente: Propia

1. Configuración de un entorno de desarrollo web

Número tarea: 1	Número historia: 1
Nombre tarea: Configuración de un entorno de desarrollo web.	
Tipo de tarea : General	Tiempo estimado: 8 horas
Fecha inicio: 03/11/12	Fecha fin: 03/11/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de instalación y configuración de un entorno de desarrollo web, utilizando software libre. Componentes a implementare: Bases de datos: PostgreSQL 9.1, pgAdmin III. Plataforma de desarrollo: Netbeans 7.2.1. Servidor de páginas web: Apache. Componentes: Twitter bootstrap master 2.0 Editores de texto: Notepad++, SciTE Source Code Editor.	

Tabla 51: Acceso al sistema - Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 1
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administrador de base de datos	Tiempo estimado: 8 horas
Fecha inicio: 04/11/12	Fecha fin: 04/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PGADMINIII</i>.</p> <p><i>createBd.sql:</i> Permite crear la base de datos turismo. <i>createUserTable.sql:</i> Permite crear la tabla usuario, con sus respectivas claves primarias y tipos de datos. <i>insertUser.sql:</i> Permite ingresar datos de los usuarios manualmente, pues en el momento no se dispone de formularios para administrar usuarios, la contraseña que ingresamos utiliza el método de encriptación <i>MD5</i>.</p>	

Tabla 52: Acceso al sistema - Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica login

Número tarea: 3	Número historia: 1
Nombre tarea: Diseño de interfaz gráfica <i>login</i> .	
Tipo de tarea : Diseño y maquetado	Tiempo estimado: 8 horas
Fecha inicio: 05/11/12	Fecha fin: 05/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para el formulario login, mediante la construcción de hojas de estilo CSS y archivos JS, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>.</p> <p>Él envió de las peticiones hacia el servidor, se la realizo definiendo el objeto <i>AJAX</i>, el cual nos permite enviar nuestra información hacia el servidor de una manera asíncrona.</p> <ul style="list-style-type: none"> - Se diseñó las clases en <i>JAVASCRIPT</i>, para que respondan a los eventos de usuario, - Construcción manual del objeto <i>AJAX</i>, para poder instanciarlos y poder reutilizarlos en las funciones que requiera él envió de información del cliente/servidor de forma asíncrona. 	

Tabla 53: Acceso al sistema - Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad usuario

Número tarea: 4	Número historia: 1
Nombre tarea: MAPEO OBJETO-RELACIONAL entidad usuario.	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio: 06/11/12	Fecha fin: 06/11/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de mapeo de la entidad usuario, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i> , para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. <ul style="list-style-type: none"> - Implementación de la Librería <i>ADODB5</i>, para gestionar las conexiones hacia nuestra bases de datos de una manera más rápida y segura. - Conversión de nuestra tabla usuario en un objeto, utilizando una clase <i>PHP</i>, para poder instanciarlo y acceder a sus respectivos atributos y métodos. 	

Tabla 54: Acceso al sistema - Tarea Nro. 4

Fuente: Propia

5. Creación de archivos PHP

Número tarea: 5	Número historia: 1
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio: 07/11/12	Fecha fin: 07/11/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i> , que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado. funciones.php: Verifica si el usuario disponga de una clave <i>MD5</i> , y si está registrado en el sistema, para proceder a crear las variables de tipo sesión. ingreso.php: Acceso hacia la interfaz de administración login.php: Verifica sí el correo y contraseña de un usuario estén registrado en la bases de datos. Salir.php: Eliminamos las variables de tipo sesión creadas por el usuario. sinPermisos.php: Retorna un mensaje indicando que el usuario no tiene permisos. sistema.php: En caso de que el usuario no este registrado, controla que no ingrese hacia el sistema.	

Tabla 55: Acceso al sistema - Tarea Nro. 5

Fuente: Propia

- **Gestión de usuarios**

Historia 2: Gestión de usuarios.		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando lenguaje <i>SQL</i> .	11 horas
2	Diseño de interfaz gráfica usuario (Formulario ingresar, editar y eliminar)	20 horas
3	MAPEO <i>OBJETO-RELACIONAL</i> entidad usuario	20 horas
4	Creación de archivos <i>PHP</i>	10 horas
5	Control de acceso hacia pantallas	10 horas
		71 horas

Tabla 56: Tareas de gestión de usuarios

Fuente: Propia

1. Creación de scripts utilizando lenguaje *SQL*

Número tarea: 1	Número historia: 2
Nombre tarea: Creación de scripts utilizando lenguaje <i>SQL</i> .	
Tipo de tarea : Administración de Bases de datos	Tiempo estimados: 11 horas
Fecha inicio: 09/11/12	Fecha fin: 10/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de script <i>SQL</i>, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>PGADMIN III</i>. Fue necesaria la reutilización de la tabla usuario, a la cual se le incorporo los siguientes campos: nombre, apellido, mail, usuario, clave, estado y contadores para cada pantalla. Se creó los scripts para poder gestionar la tabla usuario.</p> <p><i>createUser.sql:</i> Permite crear la tabla usuario, con sus respectivas claves primarias y tipos de datos. <i>insertUser.sql:</i> Permite ingresar datos de los usuarios manualmente, pues en el momento no se dispone de formularios para administrar usuarios, la contraseña que ingresamos utiliza el tipo de encriptación <i>MD5</i>.</p>	

Tabla 57: Gestión de usuarios – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica usuario (Formulario ingresar, editar y eliminar)

Número tarea: 2	Número historia: 2
Nombre tarea: Diseño de interfaz gráfica usuario (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 10/11/12	Fecha fin: 12/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de usuarios, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i>, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>.</p> <p>Formulario Ingresar usuario: Permite ingresar los campos de un usuario. Formulario Editar usuario: Permite editar los campos de un usuario seleccionado. Reporte de usuarios: Presentan una consulta de los usuarios creados.</p> <p>Funciones <i>JAVASCRIPT</i>, que actúan a los eventos respectivos de cada formulario, en cada función que requiera <i>peticiones hacia el servidor</i>. Se definió el objeto <i>AJAX</i>, el cual nos permitirá enviar nuestra información hacia el servidor de manera <i>asíncrona</i>.</p> <p>validarCamposUsuario(): Valida si los campos tengan el límite de caracteres establecidos; esta función nos permite trabajar con los formularios ingresar y editar. insertUsuario(): Recibe como parámetro los campos de los usuarios, y los envía hacia el servidor. updaUsuario(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editUsuario(): Recibe como parámetro la clave primaria de un usuario, y lo envía hacia el servidor. delUsuario(): Recibe como parámetro las claves primarias de todos los usuarios seleccionados, y la envía hacia el servidor. paginUsuario(): Permite paginar el listado de usuarios.</p>	

Tabla 58: Gestión de usuarios – Tarea Nro. 2

Fuente: Propia

3. Mapeo Objeto-Relacional entidad usuario

Número tarea: 3	Número historia: 2
Nombre tarea: MAPEO <i>OBJETO-RELACIONAL</i> entidad usuario.	
Tipo de tarea : Programación	Tiempo estimado: 20 horas
Fecha inicio: 13/11/12	Fecha fin: 15/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad usuario, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla usuario, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>usuario.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>usuario()</i>: Constructor que nos permite instanciar el objeto usuario. <i>insertar()</i>: Permite ingresar un usuarios hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla usuario, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las claves primaria de una entidad usuario y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 59: Gestión de usuarios – Tarea Nro. 3

Fuente: Propia

4. Creación de archivos *PHP*

Número tarea: 4	Número historia: 2
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 16/11/12	Fecha fin: 17/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado. <i>usuarioCons.php</i>: Instanciamos el objeto usuario para poder imprimir sus campos. <i>usuarioFormInsert.php</i>: Formulario encargado de enviar las variables hacia el servidor, para insertar los campos respectivos. <i>usuarioFormUpdate</i>: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. <i>usuarioFuncion</i>: Define el objeto usuario y accede a todos sus métodos.</p>	

Tabla 60: Gestión de usuarios – Tarea Nro. 4

Fuente: Propia

5. Control de acceso hacia pantallas

Número tarea: 5	Número historia: 2
Nombre tarea: Control de acceso hacia pantallas.	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 17/11/12	Fecha fin: 18/11/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe la creación de variables sesión, que controlarán el acceso hacia las siguientes pantallas: Gestión usuarios, gestión roles, asignación de responsabilidades, gestión de pantallas, gestión de provincias, gestión de ciudades, gestión de categorías, gestión de sitios, gestión de historias, gestión de <i>URL'S</i> hacia <i>YOUTUBE</i> , gestión de imágenes, gestión de festivales, gestión de gastronomías y gestión de rutas.	

Tabla 61: Gestión de usuarios – Tarea Nro. 5

Fuente: Propia

- **Asignación de roles**

Historia 3: Asignación de roles.		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	10 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	10 horas
3	Diseño de interfaz gráfica rol (Formulario ingresar, editar y eliminar)	18 horas
4	<i>MAPEO OBJETO-RELACIONAL</i> entidad rol.	18 horas
5	Creación de archivos <i>PHP</i> .	6 horas
		62 horas

Tabla 62: Tareas de asignación de roles

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 3
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 10 horas
Fecha inicio: 22/11/12	Fecha fin: 22/11/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre la tabla usuario y rol. - Un usuario puede tener varios roles, por lo que fue necesaria incorporar la clave primaria de usuario, hacia la tabla rol. 	

Tabla 63: Asignación de roles – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 3
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 10 horas
Fecha inicio: 23/11/12	Fecha fin: 23/11/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de un script SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . <i>creaRol.sql:</i> Crea la tabla rol. <i>primRol.sql:</i> Crea claves primarias. <i>foreRol.sql:</i> Crea las claves foráneas. <i>insrRol.sql:</i> Permite ingresar datos de prueba hacia la tabla rol.	

Tabla 64: Asignación de roles – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica rol (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 3
Nombre tarea: Diseño de interfaz gráfica rol (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 18 horas
Fecha inicio: 24/11/12	Fecha fin: 25/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de roles, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i>, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>.</p> <p>Formulario Ingresar rol: Permite ingresar los campos de un rol. Formulario Editar rol: Permite editar los campos de un rol seleccionado. Reporte de roles: presentan una consulta de los roles creados.</p> <p>Creación de menú usuarios, fusionando código <i>PHP</i>, <i>JAVASCRIPT</i> y <i>HTML</i>.</p> <p>Funciones <i>JAVASCRIPT</i>, que actúan a los eventos respectivos de cada formulario, en cada función que requiera <i>peticiones hacia el servidor</i>. Se definió el objeto <i>AJAX</i>, el cual nos permitirá enviar nuestra información hacia el servidor de manera <i>asíncrona</i>.</p> <p>validarCamposRol(): Valida los campos de formulario Ingresar y Editar. insertRol(): Recibe como parámetro los campos del rol, y los envía hacia el servidor. updaRol(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editRol(): Recibe como parámetro la clave primaria de un rol, y lo envía hacia el servidor. delRol(): Recibe como parámetro las claves primarias y foráneas de un rol, y la envía hacia el servidor. consRol(): Permite seleccionar un usuario a la vez, para asignarle los roles respectivos. paginRol(): Permite paginar el listado de roles. selectUsuarioRol_pantalla(): Retorna una matriz en formato <i>JSON</i>, con todos los roles de un usuario seleccionado en el menú usuarios.</p>	

Tabla 65: Asignación de roles – Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad rol

Número tarea: 4	Número historia: 3
Nombre tarea: MAPEO OBJETO-RELACIONAL entidad rol.	
Tipo de tarea : Programación	Tiempo estimado: 18 horas
Fecha inicio: 26/11/12	Fecha fin: 28/11/16
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad rol, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla rol, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>rol.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>rol()</i>: Constructor que nos permite instanciar el objeto rol. <i>insertar()</i>: Permite ingresar un rol hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla rol, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las clave primaria de una entidad rol y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 66: Asignación de roles – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos *PHP*

Número tarea: 5	Número historia: 3
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 29/11/12	Fecha fin: 29/11/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p><i>rolCons.php</i>: Instanciamos el objeto rol para poder acceder e imprimir sus campos. <i>rolFormInsert.php</i>: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. <i>rolFormUpdate</i>: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. <i>rolFuncion</i>: Incorporamos funciones para ingresar, editar y eliminar. <i>selectUsuarioRol</i>: Retorna los roles de un usuario en formato <i>JSON</i>, utilizando literales de objeto tipo número.</p>	

Tabla 67: Asignación de roles – Tarea Nro. 5

Fuente: Propia

- **Asignación de responsabilidades**

Historia 4: Asignación de responsabilidades.		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	10 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	10 horas
3	Diseño de interfaz gráfica responsabilidad (Formulario ingresar, editar y eliminar)	30 horas
4	MAPEO <i>OBJETO-RELACIONAL</i> entidad responsabilidad.	10 horas
5	Creación de archivos <i>PHP</i> .	23 horas
		83 horas

Tabla 68: Tareas de asignación de responsabilidades

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 4
Nombre tarea: Diseño de bases de datos	
Tipo de tarea: Administración de base de datos.	Tiempo estimado: 10 horas
Fecha inicio: 17/12/12	Fecha fin: 18/12/12
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre las tabla usuario, rol y responsabilidad. - Un usuario puede tener varias responsabilidades de un rol, por lo que fue necesaria incorporar la clave primaria de usuario y rol, hacia la tabla de responsabilidades. 	

Tabla 69: Asignación de responsabilidades – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 4
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Puntos estimados: 10 horas
Fecha inicio: 19/12/12	Fecha fin: 20/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i>.</p> <p><i>creaResp.sql:</i> Crea la tabla responsabilidad. <i>primResp.sql:</i> Crea claves primarias. <i>foreResp.sql:</i> Crea claves de foráneas. <i>inseResp.sql:</i> Permite ingresar datos de prueba hacia la tabla responsabilidad.</p>	

Tabla 70: Asignación de responsabilidades – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica responsabilidad (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 4
Nombre tarea: Diseño de interfaz gráfica responsabilidad (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 30 horas
Fecha inicio: 21/12/12	Fecha fin: 23/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de responsabilidades, mediante la construcción de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería TWITTER BOOTSTRAP MASTER, para mejorar el diseño de nuestros componentes HTML.</p> <p><i>Formulario Ingresar responsabilidad:</i> Permite ingresar los campos de una responsabilidad. <i>Formulario Editar responsabilidad:</i> Permite editar los campos de una responsabilidad seleccionada. Reporte de responsabilidades: presentan una consulta de las responsabilidades creadas. Creación de menú usuarios y roles, fusionando código PHP, JAVASCRIPT y HTML.</p>	

Funciones *JAVASCRIPT*, que actúan a los eventos respectivos de cada formulario, en cada función que requiera peticiones hacia el servidor. Se definió el objeto *AJAX*, el cual nos permitirá enviar nuestra información hacia el servidor de manera asíncrona.

validarCamposResponsabilidad(): Valida los campos de formulario ingresar y editar .

insertResponsabilidad(): Recibe como parámetro los campos de una *responsabilidad*, y los envía hacia el servidor.

updaResponsabilidad(): Recibe como parámetros los campos a editarse, y los enviar hacia el servidor.

editResponsabilidad(): Recibe como parámetro las clave primaria y foráneas de una *responsabilidad*, y la envía hacia el servidor.

deResponsabilidad(): Recibe como parámetro las claves primarias y foráneas de una *responsabilidad*, y la envía hacia el servidor.

selectUsuarioRol_pantalla(): Retorna una matriz en formato *JSON*, con todos los roles de un usuario seleccionado en el menú usuarios.

consRolResponsabilidad_pantalla(): Retorna una matriz en formato *JSON*, con todos las responsabilidades de un usuario seleccionado en el menú roles.

paginResponsabilidad(): Permite paginar el listado de responsabilidades.

Tabla 71: Asignación de responsabilidades – Tarea Nro. 3

Fuente: Propia

4. MAPEO OBJETO-RELACIONAL entidad responsabilidad

Número tarea: 4	Número historia: 4
Nombre tarea: MAPEO OBJETO-RELACIONAL entidad responsabilidad.	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 24/12/12	Fecha fin: 26/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad responsabilidad, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla responsabilidad, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: responsabilidad(): Constructor que nos permite instanciar el objeto responsabilidad. insertar(): Permite ingresar un usuarios hacia la base de datos, utilizando el comando <i>INSERT</i>. actualiza(): Permite editar los campos de la tabla usuario, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. eliminar(): Recibe como parámetros, las clave primarias de una entidad rol y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 72: Asignación de responsabilidades – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos PHP

Número tarea: 5	Número historia: 4
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 23 horas
Fecha inicio: 27/12/12	Fecha fin: 29/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p>responsabilidadCons.php: Instanciamos el objeto responsabilidad para poder imprimir sus campos.</p> <p>responsabilidadFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos.</p> <p>responsabilidadFormUpdate: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos.</p> <p>responsabilidadFuncion: Incorporamos funciones para ingresar, editar y eliminar.</p> <p>selectUsuarioRol: Retorna los roles de un usuario en formato <i>JSON</i>, utilizando literales de objeto tipo número.</p> <p>selectRolResponsabilidad: Retorna las responsabilidades de un usuario en formato <i>JSON</i>, utilizando literales de objeto tipo número.</p>	

Tabla 73: Asignación de responsabilidades – Tarea Nro. 5

Fuente: Propia

- **Asignación de permisos hacia pantallas**

Historia 5: Asignación de permisos hacia pantallas.		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	16 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	10 horas
3	Diseño de interfaz gráfica pantalla	25 horas
4	<i>MAPEO OBJETO-RELACIONAL</i> entidad pantalla.	16 horas
5	Creación de archivos <i>PHP</i> .	16 horas
		83 horas

Tabla 74: Tareas de asignación de permisos hacia pantallas

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 5
Nombre tarea: Diseño de bases de datos	
Tipo de tarea: Administración de base de datos.	Tiempo estimado: 16 horas
Fecha inicio: 17/12/12	Fecha fin: 20/12/12
Programador responsable: Jairo Morales	
Descripción: <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre la tabla usuario, rol, responsabilidad y pantalla. - Un usuario puede tener acceso hacia varias pantallas de administración ya definidas: Gestión usuarios, gestión roles, asignación de responsabilidades, asignación de permisos hacia pantallas, gestión de provincias, gestión de ciudades, gestión de categorías, gestión de sitios, gestión de historias , gestión de <i>URL'S</i> hacia <i>YOUTUBE</i>, gestión de imágenes , gestión de festivales, gestión de gastronomías, gestión de rutas. 	

Tabla 75: Asignación de permisos hacia pantallas – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 5
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Puntos estimados: 10 horas
Fecha inicio: 20/12/12	Fecha fin: 22/12/12
Programador responsable: Jairo Morales	
Descripción: <p>La presente tarea describe el proceso de creación de scripts <i>SQL</i>, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i>.</p> <p><i>creaPan.sql:</i> Crea la tabla pantalla. <i>primPan.sql:</i> Crea claves primarias. <i>forePan.sql:</i> Crea claves de foráneas. <i>insPan.sql:</i> Permite ingresar datos de prueba hacia la tabla pantalla.</p>	

Tabla 76: Asignación de permisos hacia pantallas – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica pantalla

Número tarea: 3	Número historia: 5
Nombre tarea: Diseño de interfaz gráfica pantalla	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 25 horas
Fecha inicio: 23/12/12	Fecha fin: 25/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la administración de pantallas, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i>, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>. Se construyó un panel principal con todas las pantallas definidas, además se incluyó una etiqueta <i>check</i>, para dar altas y bajas hacia las pantallas.</p> <p>Funciones <i>JAVASCRIPT</i>: consPantalla(): Imprime un listado de las pantallas disponibles: Gestión usuarios, gestión roles, gestión de responsabilidades, gestión de pantallas, gestión de provincias, gestión de ciudades, gestión de categorías, gestión de sitios, gestión de historias , gestión de <i>URL</i> 'S hacia <i>YOUTUBE</i>, gestión de imágenes, gestión de festivos, gestión de gastronomías, gestión de rutas.</p> <p>activarChkBox(): Activa las cajas de texto, para poder seleccionarlas. consultarPantallasAsignadas(): Recibe como parámetros los datos de un usuario e imprime las pantallas ya asignadas. marcarChkBox(): Marca todos los <i>CHECK</i> hijos. insertarPantallas(): Envía las variables hacia el servidor. eliminarPantalla(): Elimina la pantalla seleccionada. ComprobarSeleCheckHijos(): Desmarca el <i>CHECK</i> padre, siempre y cuando todos los <i>check</i> hijos no estén marcados. DesactivarChkBox(): Desactiva los <i>CHECK</i>.</p>	

Tabla 77: Asignación de permisos hacia pantallas – Tarea Nro. 3

Fuente: Propia

4. MAPEO OBJETO-RELACIONAL entidad pantalla

Número tarea: 4	Número historia: 5
Nombre tarea: MAPEO OBJETO-RELACIONAL entidad pantalla.	
Tipo de tarea : Programación	Tiempo estimado: 16 horas
Fecha inicio: 26/12/12	Fecha fin: 27/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad pantalla, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente.</p> <p>Convertimos nuestra tabla pantalla, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>pantalla()</i>: Constructor que nos permite instanciar el objeto pantalla. <i>insertar()</i>: Permite ingresar las claves primarias de un usuario, rol , responsabilidad , además las pantallas seleccionadas, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla pantalla, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las clave primarias de una entidad pantalla y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 78: Asignación de permisos hacia pantallas – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos PHP

Número tarea: 5	Número historia: 5
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 16 horas
Fecha inicio: 28/12/12	Fecha fin: 29/12/12
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p><i>pantallaCons.php</i>: Imprime las pantallas disponibles sobre el panel creado. <i>pantallaDel.php</i>: Elimina las pantallas que no estén seleccionadas <i>pantallaInsert</i>: Inserta las pantallas seleccionadas. <i>pantallaSelect</i>: Selecciona las pantallas disponibles asignadas a un usuario.</p>	

Tabla 79: Asignación de permisos hacia pantallas – Tarea Nro. 5

Fuente: Propia

3.3.3 Pruebas Iteración I

1. Especificación de Prueba: Acceso y control al sistema

Historia de usuario 1

Versión <1.0>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
08/11/12	1.0	Revisión	Jairo Morales
08/11/12	1.1	Modificación	Jairo Morales

Tabla 80: Iteración I, Historial de revisiones Nro. 1

Fuente: Propia

En esta historia se verifica, que la introducción de los datos hacia el sistema, cumplan las especificaciones establecidas en la Historia 1, previo acceso hacia las pantallas de administración web. Si los campos introducidos no corresponden, se imprimen alertas definidas, que informan la situación actual de la operación.

En caso de no estar registrado, deberá registrarse en la página web del sitio, en cuestión de horas un especialista atenderá su petición y habilitará el respectivo usuario.

Registro de información correcta

- Descripción

Los valores introducidos por el usuario deben cumplir, con las reglas establecidas, para acceder hacia el sistema, por el momento no dispondrá de pantallas de administración, para poder gestionar la información turística.

El campo correo electrónico debe de ser real

- Condiciones de ejecución

El usuario deberá estar previamente creado y habilitado, para poder registrarse y a acceder hacia el sistema.

- Entrada

El usuario introducirá su email y *password*.

- Resultado esperado

Tras la introducción de datos, si el usuario procesado ha ingresado sus datos correctos, accede hacia el sistema.

- Evaluación de la prueba

Prueba satisfactoria.

Registro de información incorrecta

- Descripción

Si los datos ingresados por el usuario creado y habilitado no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

Al intentar introducir un valor que no tenga el formato del email, nos generará una alerta moderada.

- Condiciones de ejecución

El usuario deberá estar previamente creado y habilitado, para poder registrarse y acceder hacia el sistema.

- Entrada

El usuario introducirá su email y *password*.

- Resultado esperado

El formulario *login*, imprime las alertas respectivas en caso de que el usuario ingrese, campos que no concuerdan.

- Evaluación de la prueba

Prueba satisfactoria.

2. Especificación de Prueba: Gestión de usuarios

Historia de usuario 2

Versión <1.1>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
19/11/12	1.1	Revisión	Jairo Morales
20/11/12	1.2	Modificación	Jairo Morales

Tabla 81: Iteración I, Historial de revisiones Nro. 2

Fuente: Propia

En esta historia se verifica, que el administrador de usuario disponga con los permisos necesarios, para poder crear usuarios.

El administrador de la aplicación, tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla usuario.

Los datos que se ingresen sobre un usuario a crear deberán cumplir con las especificaciones establecidas. Si los campos introducidos no corresponden, se imprimirá alertas definidas, que informan la situación de la operación.

Registrar la información correcta de un usuario

- Descripción

El administrador de la aplicación creará y habilitará un usuario, de acuerdo a las peticiones recibidas de los clientes.

Una vez creado el usuario, se reenviara el correo electrónico, con su contraseña de acceso respectiva.

- Condiciones de ejecución

El usuario deberá tener los permisos, para administrar usuarios.

- Entrada

El usuario administrador debe llenar los campos obligatorios en el formulario, caso contrario ingresar no aplicable.

Las contraseñas deben de coincidir.

El correo electrónico debe ser el mismo, del cual se recibió la petición.

El estado del usuario a crear debe habilitarse.

Nombres [5-32] caracteres, apellidos [5-32] caracteres, mail [5-32] caracteres, usuario (alias) [5-20] caracteres, clave [5-15] caracteres, estado activado o desactivado.

- Resultado esperado

Tras la creación de un usuario. Si el proceso ha sido correcto, el usuario creado y habilitado, podrá registrarse y acceder hacia el sistema.

Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de un usuario

- Descripción

Si los datos ingresados por el usuario administrador no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

- Condiciones de ejecución

El usuario deberá tener los permisos, para administrar usuarios.

- Entrada

El usuario administrador debe llenar los campos obligatorios en el formulario, caso contrario ingresar “no aplicable”.

Las contraseñas deben de coincidir.

El correo electrónico debe ser el mismo, del cual se recibió la petición.

El estado del usuario a crear debe habilitarse.

Los campos deberán tener un número de caracteres preestablecidos dentro de un rango.

Nombres [5-32] caracteres, apellidos [5-32] caracteres, mail [5-32] caracteres, usuario (alias) [5-20] caracteres, clave [5-15] caracteres, estado activado o desactivado.

- Resultado esperado

El formulario ingresar usuario, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

Evaluación de la prueba

Prueba satisfactoria.

3. Especificación de Prueba: Asignación de roles

Historia de usuario 3

Versión <1.2>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
23/11/12	1.2	Revisión	Jairo Morales
24/11/12	1.3	Modificación	Jairo Morales

Tabla 82: Iteración I, Historial de revisiones Nro. 3

Fuente: Propia

El control de acceso basado en roles, es una función de seguridad para controlar el acceso de usuarios a tareas que normalmente están restringidas al administrador. Mediante la aplicación de atributos de seguridad a procesos y usuario.

En esta historia se verifica, que el administrador de roles, seleccione un usuario desde un menú y pueda asignarle uno o varios roles.

El administrador de roles, tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla rol.

Registrar la información correcta de un rol

- Descripción

El administrador de roles, asignará a la vez un solo rol a un usuario.

- Condiciones de ejecución

El usuario deberá tener los permisos, para administrar roles.

- Entrada

Del menú usuarios, se debe seleccionar un solo usuario.

El nombre y descripción de rol, deberán tener un número de caracteres preestablecidos dentro de un rango.

Nombre [5-32] caracteres, descripción [5-64] caracteres.

- Resultado esperado

Tras la selección de un usuario y la creación de un rol. Si el proceso ha sido correcto, el usuario seleccionado sí dispondrá de un rol.

Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de un rol

- Descripción

Si los datos ingresados por el administrador de roles no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

- Condiciones de ejecución

El usuario deberá tener los permisos, para administrar roles.

- Entrada

Del menú usuarios, se debe seleccionar un solo usuario.

El nombre de rol y descripción, deberán tener un número de caracteres preestablecidos dentro de un rango.

- Resultado esperado

El formulario ingresar rol, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

Evaluación de la prueba

Prueba satisfactoria.

4. Especificación de Prueba: Asignación de responsabilidades

Historia de usuario 4

Versión <1.3>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
16/12/12	1.3	Revisión	Jairo Morales
17/12/12	1.4	Modificación	Jairo Morales

Tabla 83: Iteración I, Historial de revisiones Nro. 4

Fuente: Propia

El control de acceso a responsabilidades, es una función de seguridad para controlar el acceso de usuarios a funcionalidades específicas del sistema, que normalmente están restringidas hacia un único usuario.

En esta historia se verifica, que el administrador de responsabilidades, seleccione un usuario y rol, desde un menú y pueda asignarle responsabilidades.

El administrador de responsabilidades, tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla responsabilidad.

Registrar la información correcta de una responsabilidad

- Descripción

El administrador de responsabilidades, asignará a la vez una responsabilidad hacia un usuario y rol seleccionado.

Condiciones de ejecución

El usuario deberá tener los permisos, para administrar responsabilidades.

- Entrada

Del menú usuarios, se debe seleccionar un solo usuario.

Del menú roles, se debe seleccionar un solo rol, correspondiente al usuario seleccionado en el menú usuarios.

El nombre y descripción de responsabilidad, deberán tener un número de caracteres preestablecidos dentro de en un rango.

Nombre [5-32] caracteres, descripción [5-64] caracteres.

- Resultado esperado

Tras la selección de un usuario y rol, se procedió a asignarle una responsabilidad; Si el proceso ha sido correcto, el usuario seleccionado dispondrá de una responsabilidad.

- Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de una responsabilidad

- Descripción

Si los datos ingresados por el administrador de responsabilidades no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

- Condiciones de ejecución

El usuario deberá tener los permisos, para administrar responsabilidades.

- Entrada

Del menú usuarios, se debe seleccionar un solo usuario.

Del menú roles, se debe seleccionar un solo rol, correspondiente al usuario seleccionado en el menú usuarios.

El nombre y descripción de responsabilidad, deberán tener un número de caracteres preestablecidos dentro de un rango.

Nombre [5-32] caracteres, descripción [5-64] caracteres.

- Resultado esperado

El formulario ingresar responsabilidad, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

Evaluación de la prueba

Prueba satisfactoria.

5. Especificación de Prueba: Asignación de permisos hacia pantallas.

Historia de usuario 5

Versión <1.4>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
01/1/13	1.4	Revisión	Jairo Morales
01/1/13	1.5	Modificación	Jairo Morales

Tabla 84: Iteración I, Historial de revisiones Nro. 5

Fuente: Propia

El control de asignación hacia pantallas, brinda una mayor seguridad al sistema, ya que el usuario administrador podrá seleccionar una o más pantallas hacia un respectivo cliente.

Un cliente podrá disponer de todas las pantallas de administración de sitios si así lo requiere, a excepción de las pantallas de administración de usuarios, por razones de seguridad.

Selección correcta de pantallas

- Descripción

El usuario tiene la comodidad de asignar un usuario, solo marcado los componentes *CHECK*.

- Condiciones de ejecución

El usuario deberá tener los permisos, para administrar pantallas.

- Entrada

Del menú usuarios, se debe seleccionar un solo usuario.

Del menú roles, se debe seleccionar un solo rol, correspondiente al usuario seleccionado en el menú usuarios.

Del menú responsabilidades seleccionar al menos una.

- Resultado esperado

Las pantallas asignadas sí almacenan la información de pantallas asignadas hacia un usuario.

Evaluación de la prueba

Prueba satisfactoria.

3.3.4 Resultado de la 1ª Iteración de construcción

Se tuvo retrasos en los tiempos establecidos, ya que el periodo de desarrollo se extendió hacia 1 y dos días, los cuales fueron controlados a su debido tiempo, acelerando el desarrollo en tareas posteriores.

✓ **Plan de entrega inicial.**

Historia 1: Control y Acceso a sistema.

Historia 2: Gestión de usuarios.

Historia 3: Asignación de Roles a un usuario.

Historia 4: Asignación de Responsabilidades a un usuario.

Historia 5: Asignación de permisos hacia pantallas

Versión del sistema 1.5.

✓ Modificación de requerimientos.

- 1.- Acceso al sistema (Se mantiene).
- 2.- Gestión de usuarios (Se mantiene).
- 3.- Asignación de roles (Se mantiene).
- 4.- Asignación de responsabilidades (Se mantiene).
- 5.- Asignación de permisos hacia pantallas (Se mantiene).

3.3.5 Demo de la versión, Iteración I

Historia 1: Acceso al sistema (5 días)

Login: El usuario debe ingresar un email valido y una contraseña, después proceder a graficar la forma de la figura generada por el sistema, para acceder hacia el sistema.
Si el usuario no tiene el alta respectiva, no podrá acceder a ninguna pantalla de administración.

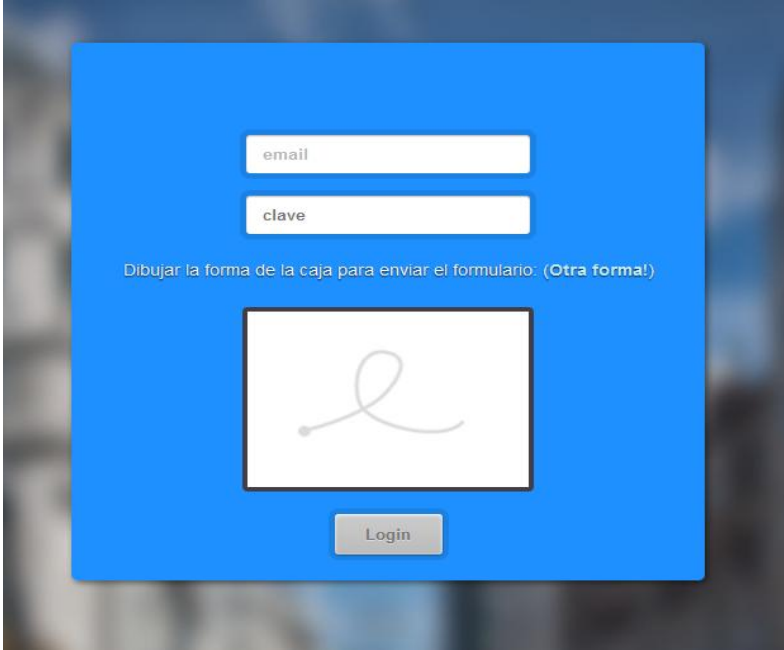
The image shows a login interface on a blue background. At the top, there are two white input fields: the first is labeled 'email' and the second is labeled 'clave'. Below these fields is a text instruction: 'Dibujar la forma de la caja para enviar el formulario. (Otra forma!)'. Underneath this instruction is a white square box containing a simple, hand-drawn cursive shape that resembles a lowercase 'e'. At the bottom of the interface is a grey button with the text 'Login'.

Ilustración 28: Formulario para ingresar al sistema

Fuente: Propia

Historia 2: Gestión de Usuarios (10 días)

Ingresar usuario: El usuario deberá presionar sobre el botón agregar usuario, el cual habilitará un formulario para ingresar los campos respectivos.

Nombres [5-32] caracteres, apellidos [5-32] caracteres, mail [5-32] caracteres, usuario (alias) [5-20] caracteres, clave [5-15] caracteres, estado activado o desactivado.

Número de registros a ingresar, de acorde a los que desee el administrador.

The screenshot shows the 'Agregar Usuario' form. On the left, a table lists users with columns for 'Nombre', 'Acciones', and 'Link'. The user 'Jairo' is listed with an edit icon and a 'Roles' link. The main form contains the following fields:

Nombres:	Mishell Guadalupe	# Sitios:	2
Apellidos:	Aux Morales	# Historia:	2
Mail:	mishel@hotmail.com	# Videos:	2
Usuario:	naser	# Fotos:	2
Clave:	****	# Festivos:	2
Repita clave:	****	# Gastronomías:	2
Estado:	<input checked="" type="radio"/> Activar <input type="radio"/> Desactivar	# Rutas:	2

Buttons: Guardar, Cerrar

Ilustración 29: Formulario para ingresar una entidad usuario

Fuente: Propia

Editar usuario: El usuario deberá presionar sobre la imagen editar, la cual habilitará un formulario para editar los campos respectivos.

Nombres [5-32] caracteres, apellidos [5-32] caracteres, mail [5-32] caracteres, usuario (alias) [5-20] caracteres, clave [5-15] caracteres, estado activado o desactivado.

Número de registros a ingresar, de acorde a los que desee el administrador.

Eliminar usuario: Presionar sobre la imagen eliminar, el cual procederá a borrar el respectivo usuario.

The screenshot shows the 'Editar Usuario' form. On the left, the same table as in Illustration 29 is visible. The main form contains the following fields:

Nombre:	Jairo Alberto	# Sitios:	10000
Apellidos:	Morales de la Torre	# Historia:	50000
Mail:	dmlajairo@gmail.com	# Videos:	50000
Usuario:	Jairo	# Fotos:	50000
Clave:	jairo123	# Festivos:	50000
Repita clave:	jairo123	# Gastronomías:	50000
# Estado:	<input checked="" type="radio"/> Activar <input type="radio"/> Desactivar	# Rutas:	50000

Buttons: Guardar, Cerrar

Ilustración 30: Formulario para editar una entidad usuarios

Fuente: Propia

Historia 3: Asignación de roles (8 días)

Ingresar roles: El usuario deberá presionar sobre el botón agregar roles, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres para los roles: Nombre [5-32] caracteres, descripción [5-64] caracteres.

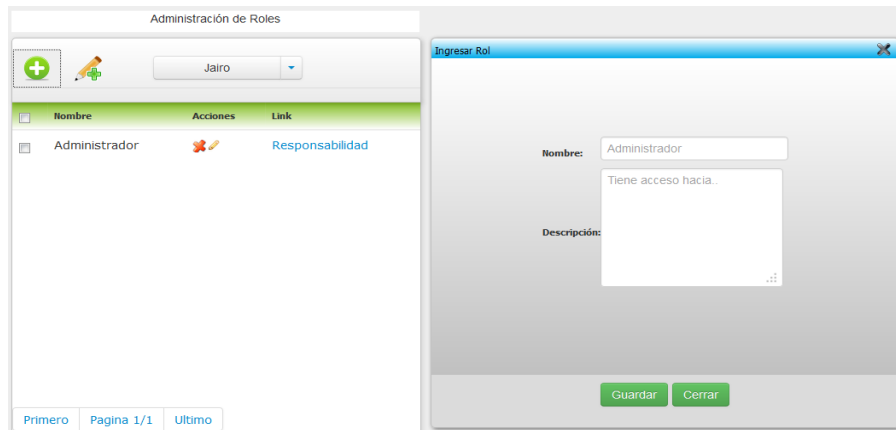


Ilustración 31: Formulario para ingresar una entidad rol

Fuente: Propia

Editar roles: El usuario deberá presionar sobre la imagen editar roles, el cual habilitará un formulario para editar los campos respectivos.

Número de caracteres para los roles: Nombre [5-32] caracteres, descripción [5-64] caracteres.

Eliminar un rol: El usuario deberá presionar sobre la imagen eliminar, para borrar un rol.

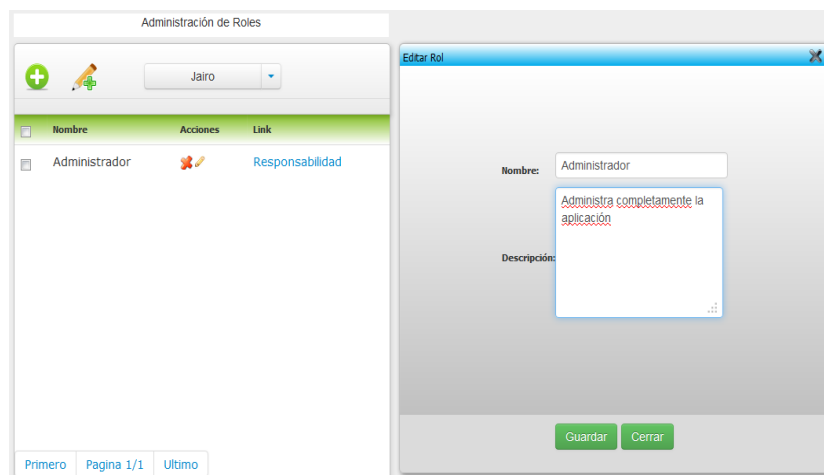


Ilustración 32: Formulario para editar una entidad rol

Fuente: Propia

Historia 4: Asignación de Responsabilidades (13 días)

Ingresar responsabilidades: El usuario deberá presionar sobre el botón agregar responsabilidades, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres: Nombre [5-32] caracteres, descripción [5-64] caracteres.

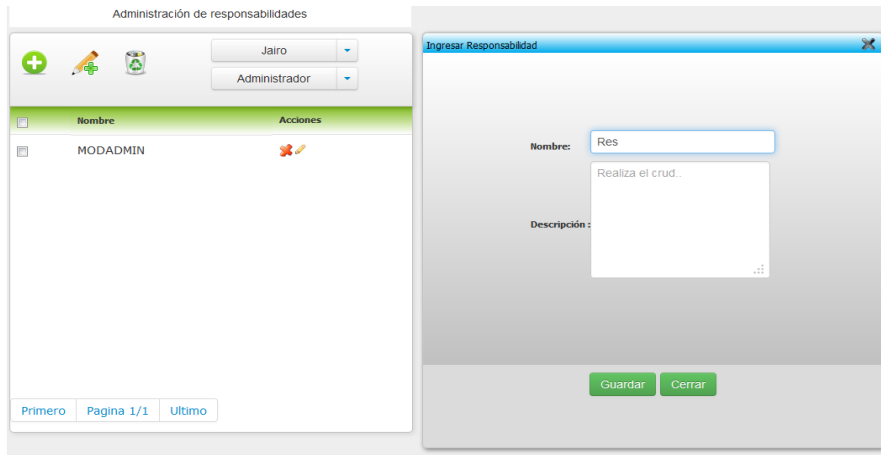


Ilustración 33: Formulario para ingresar una entidad responsabilidad

Fuente: Propia

Editar responsabilidades: El usuario deberá presionar sobre la imagen editar responsabilidades, el cual habilitará un formulario para editar los campos respectivos.

Número de caracteres: Nombre [5-32] caracteres, descripción [5-64] caracteres.

Eliminar una responsabilidad: Él usuario deberá presionar sobre la imagen eliminar, para borrar una responsabilidad.

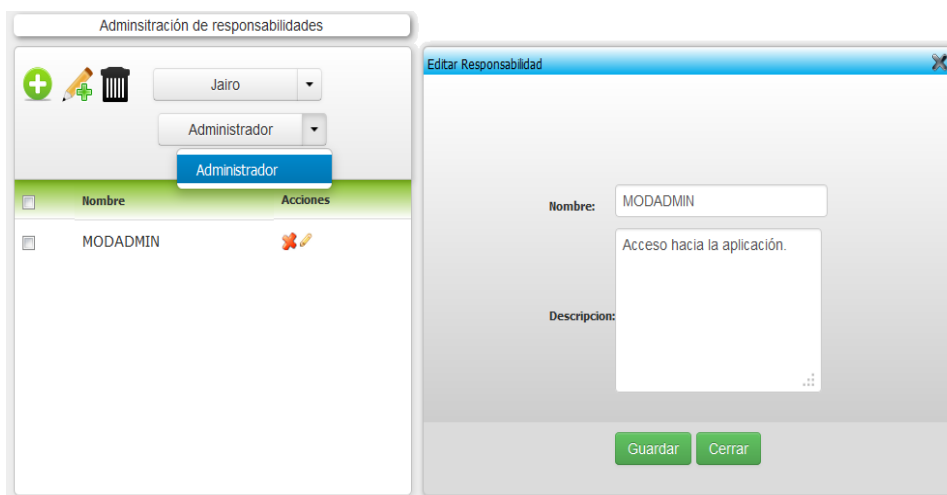


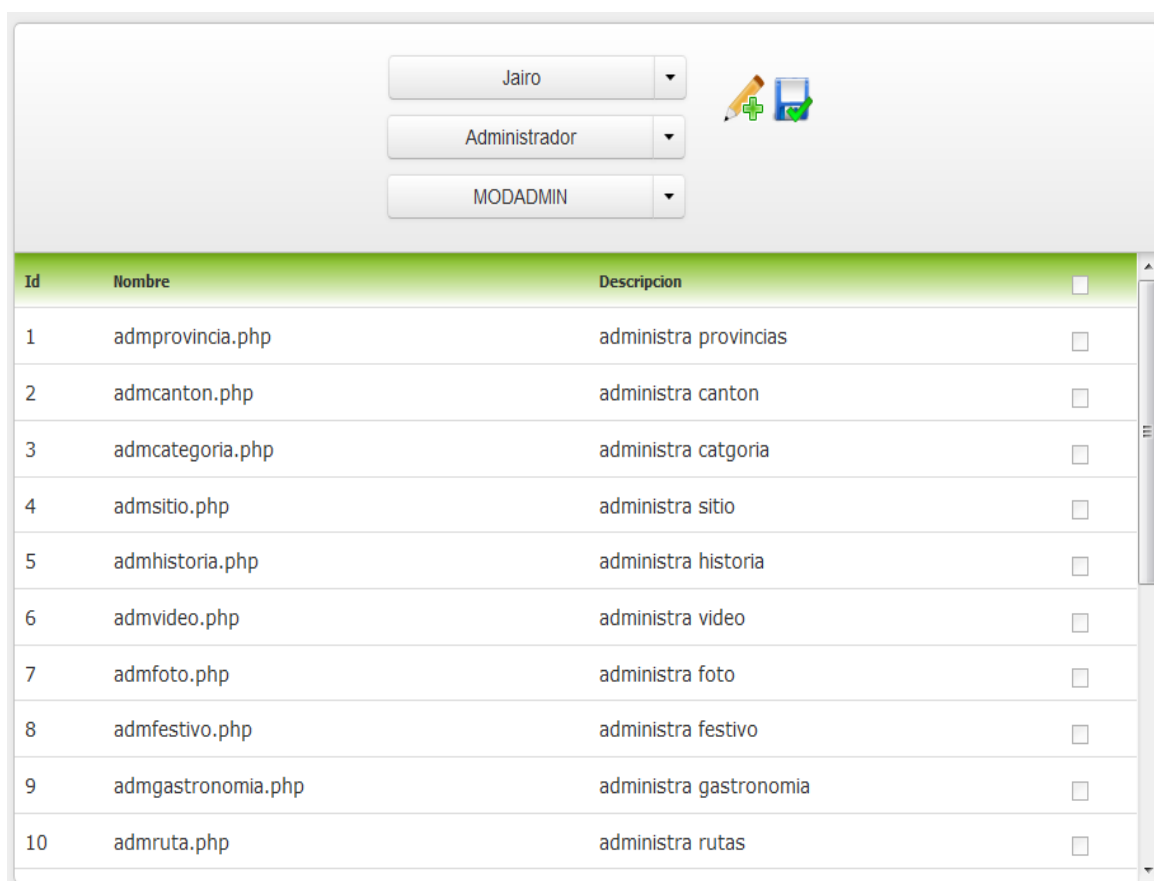
Ilustración 34: Formulario para editar una entidad responsabilidad

Fuente: Propia

Historia 5: Asignación de permisos hacia pantallas (13 días)

Agregar permisos hacia pantallas: Reporte de todas las pantallas de administración: Provincia, ciudad, categoría, sitio, historia, foto, video, gastronomía, festivo, ruta, usuario, rol, responsabilidad y pantalla.

El usuario administrador deberá seleccionar un usuario, rol y responsabilidad, para asignarle los accesos respectivos y almacenarlos en la base de datos.



Id	Nombre	Descripcion	<input type="checkbox"/>
1	admprovincia.php	administra provincias	<input type="checkbox"/>
2	admcanton.php	administra canton	<input type="checkbox"/>
3	admcategoria.php	administra categoria	<input type="checkbox"/>
4	admsitio.php	administra sitio	<input type="checkbox"/>
5	admhistoria.php	administra historia	<input type="checkbox"/>
6	admvideo.php	administra video	<input type="checkbox"/>
7	admfoto.php	administra foto	<input type="checkbox"/>
8	admfestivo.php	administra festivo	<input type="checkbox"/>
9	admgastronomia.php	administra gastronomia	<input type="checkbox"/>
10	admruta.php	administra rutas	<input type="checkbox"/>

Ilustración 35: Formulario para gestionar los permisos al usuario

Fuente: Propia

3.3.6 Pruebas de aceptación

Historial 1: Acceso al sistema

- Acceso hacia el sistema seguro, pues se cuenta con una contraseña encriptado tipo MD5, además variables de tipo sesión.
- El formulario *login* cuenta con la validación de los campos respectivos.
- Una vez que el usuario ingresa su correo y contraseña, procede hacia el sistema
- Sí un usuario no está registrado en la base de datos, no tendrá un acceso.

Historia 2.- Gestión de usuarios

- Formulario para ingresar un usuario, sí ingresa un usuario con los respectivos campos.
- Formulario para editar la información de un usuario, sí permite editar la información respectiva de un usuario.

Historia 3.- Asignación de roles

- Formulario para asignar uno o varios roles a un usuario, permite seleccionar un usuario, desde un menú denominado usuarios y posteriormente asignarle un rol.
- Formulario para editar la información de un rol, sí permite editar la información respectiva de un rol seleccionado.

Historia 4.- Asignación de responsabilidades

- Formulario para asignar responsabilidades, permite seleccionar un usuario y rol, posteriormente asignarle una responsabilidad respectiva.
- Formulario para editar la información de una responsabilidad, sí permite editar la información respectiva de una responsabilidad.

Historia 5.- Asignación de permisos hacia pantallas

- El panel de pantallas permite seleccionar un usuario, rol y responsabilidad respectiva. Para posteriormente asignarle una pantalla de administración.

3.3.7 Incidencias

- Se rediseño la tabla usuario, pues inicialmente constaba de pocos campos.
- Los formularios cambiaron de aspecto frecuentemente, ya que fueron apareciendo nuevos estándares de diseño.
- El modelado de la base de datos sufrió cambios esporádicos, en el trascurso de diseño.

- A no tener las pantallas de administración disponibles, se simulo utilizando archivos que solo imprimen “hola mundo”.

3.4 Iteración II

3.4.1 Cronograma de actividades

ITE.	Historias de usuario	Fecha	Duración en Semanas	Duración en horas
2	Historia 6 Gestión de provincias	02/1/13 06/1/13	1	40
2	Historia 7 Gestión de ciudades	08/1/13 14/1/13	1,2	56
2	Historia 8 Gestión de categorías	18/1/13 23/1/13	1,1	48
2	Historia 9 Gestión de sitios	25/1/13 05/2/13	2,2	90
2	Historia 10 Gestión de historias	08/2/13 17/2/13	2	71
2	Historia 11 Gestión de URL'S hacia <i>YouTube</i>	21/2/13 27/2/13	1,2	56
2	Historia 12 Gestión de imágenes	02/3/13 07/3/13	1,1	48
2	Historia 13 Gestión de festivos	10/3/13 15/3/13	1,1	48
	Historia 14 Gestión de gastronomías	16/3/13 21/3/13	1,1	48
2	Historia 15 Gestión de rutas	22/3/13 30/3/13	1,4	67
				Total: 572 h

Tabla 85: Cronograma de actividades Iteración II

Fuente: Propia

3.4.2 Tareas por historia iteración II

- Gestión de provincias

Historia 6: Gestión de provincias.		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
2	Diseño de interfaz gráfica provincia (Formulario ingresar, editar y eliminar)	12 horas
3	Mapeo Objeto-Relacional entidad provincia.	8 horas
4	Creación de archivos <i>PHP</i> .	12 horas
		40 horas

Tabla 86: Tareas de gestión de provincias

Fuente: Propia

1. Creación de scripts utilizando lenguaje *SQL*

Número tarea: 1	Número historia: 6
Nombre tarea: Creación de scripts utilizando lenguaje <i>SQL</i> .	
Tipo de tarea : Administrador de base de datos	Tiempo estimado: 8 horas
Fecha inicio: 02/1/13	Fecha fin: 02/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de scripts <i>SQL</i>, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i>.</p> <p><i>creaProv.sql:</i> Permite crear la tabla provincia. <i>Primprov.sql:</i> Creación de claves primarias. <i>foreProv.sql:</i> Creación de claves foráneas. <i>inseProv.sql:</i> Inserción de datos hacia la tabla provincia, pues en el momento no se dispone de formularios para administrar provincias.</p>	

Tabla 87: Gestión de provincias – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica provincia (Formulario ingresar, editar y eliminar)

Número tarea: 2	Número historia: 6
Nombre tarea: Diseño de interfaz gráfica provincia (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 12 horas
Fecha inicio: 03/1/13	Fecha fin: 03/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de provincias, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i>, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>.</p> <p>Él envío de las peticiones hacia el servidor, se lo realiza definiendo el objeto <i>AJAX</i>, el cual nos permitirá enviar nuestra información hacia el servidor de manera asíncrona.</p> <p>Formulario Ingresar provincia: Permite ingresar los campos de una provincia. Formulario Editar provincia: Permite editar los campos de una provincia seleccionada. Reporte de provincia: Podemos seleccionar uno o más provincias y eliminarlas.</p> <p>Funciones JavaScript, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor, fue necesario instanciar una comunicación <i>AJAX</i> asíncrona con el servidor.</p> <p>validarCamposProvincia(): Valida los campos de formulario ingresar y editar. insertProvincia(): Recibe como parámetro los campos de una provincia, y los envía hacia el servidor. updaProvinci a(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editProvincia(): Recibe como parámetro la clave primaria de una provincia, y lo envía hacia el servidor. delProvincia(): Recibe como parámetro las claves primarias de todas las provincias seleccionadas, y la envía hacia el servidor. paginProvincia(): Paginar el listado de provincias.</p>	

Tabla 88: Gestión de provincias – Tarea Nro. 2

Fuente: Propia

3. Mapeo Objeto-Relacional entidad provincias

Número tarea: 3	Número historia: 6
Nombre tarea: Mapeo Objeto-Relacional entidad provincia.	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio: 04/1/13	Fecha fin: 04/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad provincia, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla provincia, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>provincia.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>provincia()</i>: Constructor que nos permite instanciar el objeto provincia. <i>insertar()</i>: Permite ingresar una provincia hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla provincia, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetro, la clave primarias de una entidad provincia y la elimina de la base de datos, utilizando el comando <i>DELETE</i></p>	

Tabla 89: Gestión de provincias – Tarea Nro. 3

Fuente: Propia

4. Creación de archivos PHP

Número tarea: 4	Número historia: 6
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 12 horas
Fecha inicio: 05/1/13	Fecha fin: 06/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado. <i>provinciaCons.php</i>: Instanciamos el objeto provincia para poder imprimir sus campos. <i>provinciaFormInsert.php</i>: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. <i>provinciaFormUpdate</i>: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. <i>provinciaFuncion</i>: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 90: Gestión de provincias – Tarea Nro. 4

Fuente: Propia

- **Gestión de ciudades**

Historia 7: Gestión de ciudades		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	6 horas
2	Creación de scripts utilizando lenguaje SQL.	10 horas
3	Diseño de interfaz gráfica ciudad (Formulario ingresar, editar y eliminar)	20 horas
4	Mapeo Objeto-Relacional entidad ciudad.	10 horas
5	Creación de archivos <i>PHP</i> .	10 horas
		56 Horas

Tabla 91: Tareas de gestión de ciudades

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 7
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 08/1/13	Fecha fin: 08/1/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia y cantón. - Una provincia puede tener varias ciudades, por lo que fue necesaria incorporar la clave primaria de provincia, hacia la tabla de ciudad. 	

Tabla 92: Gestión de ciudades – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 7
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 10 horas
Fecha inicio: 10/1/13	Fecha fin: 10/1/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . creaCiud.sql: Permite crear la tabla ciudad. primCiud.sql: Creación de claves primarias. foreCiud.sql: Creación de claves foráneas. inseProv.sql: Inserción de datos hacia la tabla ciudad, pues en el momento no se dispone de formularios para administrar ciudades.	

Tabla 93: Gestión de ciudades – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica ciudad (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 7
Nombre tarea: Diseño de interfaz gráfica ciudad (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 11/1/13	Fecha fin: 12/1/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de ciudades, mediante la construcción de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería TWITTER BOOTSTRAP MASTER, para mejorar el diseño de nuestros componentes HTML. Formulario Ingresar ciudad: Permite ingresar los campos de texto respectivos. Formulario Editar ciudad: Permite editar los campos de texto seleccionados. Reporte de ciudades: Permite imprimir las ciudades de una provincia. Creación de menú provincia, fusionando código PHP, JAVASCRIPT y HTML. Funciones JAVASCRIPT, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor, fue necesario instanciar una comunicación AJAX asíncrona con el servidor. validarCamposCanton(): Valida los campos de formulario ingresar y editar. insertCanton(): Recibe como parámetro los campos del cantón, y los envía hacia el servidor. updaCanton(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editCanton(): Recibe como parámetro la clave primaria de un cantón, y lo envía hacia el servidor. delCanton(): Recibe como parámetro las claves primarias y foráneas de un cantón, y los envía hacia el servidor. consCanton(): Permite seleccionar una provincia a la vez, para asignarle los cantones respectivos. paginCanton (): Permite paginar el listado de cantones.	

Tabla 94: Gestión de ciudades – Tarea Nro. 3

Fuente: Propia

4.- Mapeo Objeto-Relacional entidad ciudad

Número tarea: 4	Número historia: 7
Nombre tarea: Mapeo Objeto-Relacional entidad ciudad.	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 13/1/13	Fecha fin: 13/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad ciudad, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente.</p> <p>Métodos implementados: ciudad():Constructor que nos permite instanciar el objeto rol. insertar():Permite ingresar una ciudad hacia la base de datos, utilizando el comando <i>INSERT</i>. actualiza():Permite editar los campos de la tabla ciudad, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. eliminar():Recibe como parámetros, las claves primarias de una entidad ciudad y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 95: Gestión de ciudades – Tarea Nro. 4

Fuente: Propia

5.- Creación de archivos PHP

Número tarea: 5	Número historia: 7
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 14/1/13	Fecha fin: 14/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p>ciudadCons.php: Instanciamos el objeto cantón para poder imprimir sus campos. ciudadFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. ciudadFormUpdate.php: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. ciudadFuncion.php: Incorporamos funciones para ingresar, editar y eliminar. selectProvinciaCiudad.php: Retorna un listado de las provincias disponibles en formato <i>JSON</i>.</p>	

Tabla 96: Gestión de ciudades – Tarea Nro. 5

Fuente: Propia

- **Gestión de categorías**

Historia 8: Gestión de categorías		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
2	Diseño de interfaz gráfica categoría (Formulario ingresar, editar y eliminar)	20 horas
3	Mapeo Objeto-Relacional entidad categoría.	10 horas
4	Creación de archivos <i>PHP</i> .	10 horas
		48 horas

Tabla 97: Tareas de gestión de categorías

Fuente: Propia

1. Creación de scripts utilizando lenguaje *SQL*

Número tarea: 1	Número historia: 8
Nombre tarea: Creación de scripts utilizando lenguaje <i>SQL</i> .	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 8 horas
Fecha inicio: 18/1/13	Fecha fin: 18/1/13
Programador responsable: Jairo Morales	
<p>Descripción: Creación de scripts <i>SQL</i>, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i>.</p> <p><i>creaCate.sql</i>: Permite crear la tabla categoría. <i>primCate.sql</i>: Creación de claves primarias. <i>foreCate.sql</i>: Creación de claves foráneas. <i>inseCate.sql</i>: Inserción de datos hacia la tabla categoría, pues en el momento no se dispone de formularios para administrar categorías.</p>	

Tabla 98: Gestión de categorías – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica categoría (Formulario ingresar, editar y eliminar)

Número tarea: 2	Número historia: 8
Nombre tarea: Diseño de interfaz gráfica categoría (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 19/1/13	Fecha fin: 20/1/13
Programador responsable: Jairo Morales	
<p>Descripción: Construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i>, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>. Formulario Ingresar categoría: Permite ingresar los campos de texto respectivos. Formulario Editar categoría: Permite editar los campos de texto seleccionados. Funciones JavaScript, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor. validarCamposCategoria(): Valida los campos de formulario ingresar y editar. insertCategoria(): Recibe como parámetro los campos de categoría, y los envía hacia el servidor. updaCategoria(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editCategoria(): Recibe como parámetro la clave primaria de un categoría , y lo envía hacia el servidor. delCategoria(): Recibe como parámetro las claves primarias y foráneas de un categoría , y los envía hacia el servidor. consCategoria(): Permite consultar las categorías. paginCategoria(): Permite paginar el listado de categorías.</p>	

Tabla 99: Gestión de categorías – Tarea Nro. 2

Fuente: Propia

3. Mapeo Objeto-Relacional entidad categoría

Número tarea: 3	Número historia: 8
Nombre tarea: Mapeo Objeto-Relacional entidad categoría.	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 21/1/13	Fecha fin: 21/1/13
Programador responsable: Jairo Morales	
<p>Descripción: Mapeo de la entidad usuario, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla categoría, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>categoría.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos. categoría(): Constructor que nos permite instanciar el objeto categoría. insertar(): Permite ingresar una categoría hacia la base de datos, utilizando el comando <i>INSERT</i>. actualiza(): Permite editar los campos de la tabla categoría, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. eliminar(): Recibe como parámetro, la clave primaria de una entidad categoría y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 100: Gestión de categorías – Tarea Nro. 3

Fuente: Propia

4. Creación de archivos PHP

Número tarea: 4	Número historia: 8
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 22/1/13	Fecha fin: 23/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p>categoríaCons.php: Instanciamos el objeto categoría para poder imprimir sus campos.</p> <p>categoríaFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos.</p> <p>categoríaFormUpdate.php: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos.</p> <p>categoríaFuncion.php: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 101: Gestión de categorías – Tarea Nro. 4

Fuente: Propia

- **Gestión de sitios**

Historia 9: Gestión de sitios		
Número	Nombre	Tiempo estimado
1	Configuración de un ambiente <i>GIS</i>	12 horas
2	Diseño de base de datos	6 horas
3	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
4	Diseño de interfaz gráfica sitio (Formulario ingresar, editar y eliminar)	25 horas
5	Configuración de un visor de mapas	25 horas
6	Mapeo Objeto-Relacional entidad sitio.	8 horas
7	Creación de archivos <i>PHP</i> .	6 horas
		Total: 90 horas

Tabla 102: Tareas de Gestión de sitios

Fuente: Propia

1. Configuración de un ambiente GIS

Número tarea: 1	Número historia: 9
Nombre tarea: Configuración de un ambiente GIS	
Tipo de tarea: Administración de base de datos geografía.	Tiempo estimado: 12 horas
Fecha inicio: 25/1/13	Fecha fin: 26/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de instalación de componentes, para gestionar información geográfica.</p> <ul style="list-style-type: none"> - <i>POSTGIS=1.5.2</i> - <i>GEOS=3.2.2-CAPI-1.6.2</i> - <i>PROJ=Rel. 4.6.1, 21 August 2008</i> - <i>LIBXML=2.7.6 USE_STATS"</i> - Importación de librerías <i>Open Layes</i> 	

Tabla 103: Gestión de sitios – Tarea Nro. 1

Fuente: Propia

2- Diseño de base de datos

Número tarea: 2	Número historia: 9
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 27/1/13	Fecha fin: 27/1/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño de bases de datos</p> <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio. - Un usuario puede seleccionar una provincia, ciudad, categoría y asignarle un sitio. - Se migró todas las claves primarias hacia la tabla sitio. 	

Tabla 104: Gestión de sitios – Tarea Nro. 2

Fuente: Propia

3. Creación de scripts utilizando lenguaje SQL

Número tarea: 3	Número historia: 9
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea: Administración de Bases de datos geográfica.	Tiempo estimado: 8 horas
Fecha inicio: 28/1/13	Fecha fin: 28/1/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdminIII</i> . creaSitio.sql: Creación de la tabla provincia, ciudad, categoría y sitio. primSitio.sql: Claves primarias de las tablas sitio. foreSitio.sql: Claves foráneas de las tala sitio. inseSitio.sql: Generación de datos para la tabla sitio. geomSitio.sql: Creación del campo <i>geometry</i> , que almacenara la latitud y longitud de un sitio, previa transformación de sistema de referencias 4326.	

Tabla 105: Gestión de sitios – Tarea Nro. 3

Fuente: Propia

4. Diseño de interfaz gráfica sitio (Formulario ingresar, editar y eliminar)

Número tarea: 4	Número historia: 9
Nombre tarea: Diseño de interfaz gráfica sitio (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 25 horas
Fecha inicio: 29/1/13	Fecha fin: 31/1/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de sitios, mediante la construcción de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería TWITTER BOOTSTRAP MASTER, para mejorar el diseño de nuestros componentes HTML. Formulario Ingresar sitio: Permite ingresar los campos de un sitio. Formulario Editar sitio: Permite editar los sitios. Creación de menú provincia y ciudad fusionando código PHP, JAVASCRIPT y HTML. Creación e Integración del menú categoría hacia Formulario ingresar sitio. Funciones JAVASCRIPT, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor, fue necesario instanciar una comunicación AJAX asíncrona con el servidor. validarCamposSitio(): Valida los campos de formulario ingresar y editar. insertSitio(): Recibe como parámetro los campos del sitio, y los envía hacia el servidor. updaSitio(): Recibe como parámetros los campos a editarse, y los enviar hacia el servidor. editSitio(): Recibe como parámetro la clave primaria de un sitio, y lo envía hacia el servidor. delSitio(): Recibe como parámetro las claves primarias y foráneas de un sitio, y la envía hacia el servidor. consSitio(): Permite seleccionar un sitio a la vez, para ingresarlo. paginSitio(): Pagar el reporte sitio.	

Tabla 106: Gestión de sitios – Tarea Nro. 4

Fuente: Propia

5. Configuración de un visor de mapas

Número tarea: 5	Número historia: 9
Nombre tarea: Configuración de un visor de mapas.	
Tipo de tarea: Programación <i>G/S</i> .	Tiempo estimado: 25 horas
Fecha inicio: 01/2/13	Fecha fin: 03/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe la configuración de un visor mapas, para poder visualizar partes de mapas interactivos en nuestro navegador web, con las opciones habilitadas que nos permitirá agregar un marcador que corresponderá a un sitio.</p> <p>Funciones JavaScript implementadas: init(): Definimos un objeto <i>OpenLayers.Map</i>, el cual permitirá imprimir partes de un mapa en nuestro navegador. borrarFeatures(): Borrarnos todas las capas creadas por el usuario. obtenerPunto(): Obtenemos la latitud y longitud de nuestro mapa, y realizamos la respectiva transformación hacia el sistema de referencia 4326, que es el más utilizado por la comunidad de desarrolladores <i>G/S</i>. agregarCapaEditar(): Recibe como parámetros la latitud y longitud de nuestro sitio y la añade en forma de marcador hacia nuestro mapa.</p>	

Tabla 107: Gestión de sitios – Tarea Nro. 5

Fuente: Propia

6. Mapeo Objeto-Relacional entidad sitio

Número tarea: 6	Número historia: 9
Nombre tarea: Mapeo Objeto-Relacional entidad <i>sitio</i> .	
Tipo de tarea: Programación <i>G/S</i> .	Tiempo estimado: 8 horas
Fecha inicio: 04/2/13	Fecha fin: 04/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad sitio, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente.</p> <p>Convertimos nuestra tabla sitio, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>sitio.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: sitio(): Constructor que nos permite instanciar el objeto sitio. insertar(): Permite ingresar un sitio hacia la base de datos, utilizando el comando <i>INSERT</i> y la función <i>ST_GeomFromText</i>. actualiza(): Permite editar los campos de la tabla sitio, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. eliminar(): Recibe como parámetros, la claves primarias de un entidad sitio y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 108: Gestión de sitios – Tarea Nro. 6

Fuente: Propia

7. Creación de archivos *PHP*

Número tarea: 7	Número historia: 9
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 05/2/13	Fecha fin: 05/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p>sitioCons.php: Instanciamos el objeto sitio para poder acceder e imprimir sus campos.</p> <p>sitioFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos.</p> <p>sitioFormUpdate: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos.</p> <p>sitioFuncion: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 109: Gestión de sitios – Tarea Nro. 7

Fuente: Propia

- **Gestión de historias**

Historia 10: Gestión de historias		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	10 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	6 horas
3	Diseño de interfaz gráfica historias (Formulario ingresar, editar y eliminar)	35 horas
4	Mapeo Objeto-Relacional entidad historia.	12 horas
5	Creación de archivos <i>PHP</i> .	8 horas
		Total: 71 horas

Tabla 110: Tareas de Gestión de historias

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 10
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 10 horas
Fecha inicio: 08/2/13	Fecha fin: 08/2/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none">- Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio e historia.- Un usuario puede seleccionar una provincia, ciudad, sitio y asignarle una o varias historias.- Se migró todas las claves primarias hacia la tabla historia.	

Tabla 111: Gestión de historias – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 10
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 6 horas
Fecha inicio: 09/2/13	Fecha fin: 09/2/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . <i>creaHist.sql:</i> Permite crear la tabla historia. <i>primHist.sql:</i> Creación de claves primarias. <i>foreHist.sql:</i> Creación de claves foráneas. <i>inserHist.sql:</i> Inserción de datos hacia la tabla historia, pues en el momento no se dispone de formularios para administrar historias.	

Tabla 112: Gestión de historias – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica historias (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 10
Nombre tarea: Diseño de interfaz gráfica historias (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 35 horas
Fecha inicio: 10/2/13	Fecha fin: 13/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de historias, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i>, además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i>, para mejorar el diseño de nuestros componentes <i>HTML</i>.</p> <p>Formulario Ingresar historia: Permite ingresar los campos de texto respectivos. Formulario Editar historia: Permite editar los campos de texto seleccionados. Creación de menú provincia, ciudad y sitio fusionando código <i>PHP</i>, <i>JAVASCRIPT</i> y <i>HTML</i>.</p> <p>Funciones <i>JAVASCRIPT</i>, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor, fue necesario instanciar una comunicación <i>AJAX</i> asíncrona con el servidor.</p> <p>validarCamposHistoria(): Valida los campos de formulario ingresar y editar. insertHistoria(): Recibe como parámetro los campos de una historia, y los envía hacia el servidor. updaHistoria(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editHistoria(): Recibe como parámetro la clave primaria de una historia, y lo envía hacia el servidor. deHistoria(): Recibe como parámetro las claves primarias y foráneas de una historia, y los envía hacia el servidor. consHistoria(): Permite consultar las historias. paginHistoria(): Permite paginar el listado de historias.</p>	

Tabla 113: Gestión de historias – Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad historia

Número tarea: 4	Número historia: 10
Nombre tarea: Mapeo Objeto-Relacional entidad <i>historia</i> .	
Tipo de tarea : Programación	Tiempo estimado: 12 horas
Fecha inicio:14/2/13	Fecha fin:15/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad historia, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla <i>historia</i>, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>historia.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>historia ()</i>: Constructor que nos permite instanciar el objeto <i>historia</i>. <i>insertar()</i>: Permite ingresar una <i>historia</i> hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla <i>historia</i>, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las claves primaria de una entidad <i>historia</i> y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 114: Gestión de historias – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos *PHP*

Número tarea: 5	Número historia: 10
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio:16/2/13	Fecha fin:17/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p><i>historiaCons.php</i>: Instanciamos el objeto <i>historia</i> para poder imprimir sus campos. <i>historiaFormInsert.php</i>: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. <i>historiaFormUpdate.php</i>: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. <i>historiaFuncion.php</i>: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 115: Gestión de historias – Tarea Nro. 5

Fuente: Propia

- **Gestión de URL 'S hacia YouTube**

Historia 11: Gestión de URL 'S hacia YouTube.		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	7 horas
2	Creación de scripts utilizando lenguaje SQL.	8 horas
3	Diseño de interfaz gráfica historias (Formulario ingresar, editar y eliminar	25 horas
4	Mapeo Objeto-Relacional entidad video.	8 horas
5	Creación de archivos PHP.	8 horas
		Total: 56 horas

Tabla 116: Tareas de gestión de URL'S hacia YouTube

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 11
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 7 horas
Fecha inicio: 21/2/13	Fecha fin: 21/2/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio y video. - Un usuario puede seleccionar una provincia, ciudad, sitio y asignarle uno o varios videos. - Se migró todas las claves primarias hacia la tabla video. 	

Tabla 117: Gestión de URL 'S hacia YouTube – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 11
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 8 horas
Fecha inicio: 22/2/13	Fecha fin: 22/2/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts <i>SQL</i> , para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . <i>creVideo.sql</i> : Permite crear la tabla video. <i>primVideo.sql</i> : Creación de claves primarias. <i>foreVideo.sql</i> : Creación de claves foráneas. <i>insertVideo.sql</i> : Inserción de datos hacia la tabla video, pues en el momento no se dispone de formularios para administrar videos.	

Tabla 118: Gestión de URL'S hacia YouTube – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica video (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 11
Nombre tarea: Diseño de interfaz gráfica video (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 25 horas
Fecha inicio: 23/2/13	Fecha fin: 25/2/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de videos, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i> , además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para mejorar el diseño de nuestros componentes <i>HTML</i> . <i>Formulario Ingresar video</i> : Permite ingresar los campos de texto respectivos. <i>Formulario Editar video</i> : Permite editar los campos de texto seleccionados. Creación de menú provincia, ciudad y sitio fusionando código <i>PHP</i> , <i>JavaScript</i> y <i>HTML</i> . Funciones <i>JAVASCRIPT</i> , que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor. <i>validarCamposVideo()</i> : Valida los campos de formulario ingresar y editar. <i>insertVideo()</i> : Recibe como parámetro los campos de un video, y los envía hacia el servidor. <i>updaVideo()</i> : Recibe como parámetros los campos a editarse, y los envía hacia el servidor. <i>editVideo()</i> : Recibe como parámetro la clave primaria de un video, y lo envía hacia el servidor. <i>deVideo()</i> : Recibe como parámetro las claves primarias y foráneas de un video, y los envía hacia el servidor. <i>consVideo()</i> : Permite consultar los videos. <i>paginVideo()</i> : Permite <i>paginar el listado de videos</i> .	

Tabla 119: Gestión de URL'S hacia YouTube – Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad video

Número tarea: 4	Número historia: 11
Nombre tarea: Mapeo Objeto-Relacional entidad <i>historia</i> .	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio: 26/2/13	Fecha fin: 26/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad usuario, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla <i>video</i>, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>video.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos. Métodos implementados: video(): Constructor que nos permite instanciar el objeto <i>video</i>. insertar(): Permite ingresar un <i>video</i> hacia la base de datos, utilizando el comando <i>INSERT</i>. actualiza(): Permite editar los campos de la tabla <i>video</i>, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. eliminar(): Recibe como parámetros, las claves primarias de una entidad <i>video</i> y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 120: Gestión de URL'S hacia YouTube – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos *PHP*

Número tarea: 5	Número historia: 11
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio: 27/2/13	Fecha fin: 27/2/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado. videoCons.php: Instanciamos el objeto <i>video</i> para poder imprimir sus campos. videoFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. videoFormUpdate.php: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. videoFuncion.php: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 121: Gestión de URL'S hacia YouTube – Tarea Nro. 5

Fuente: Propia

- **Gestión de imágenes**

Historia 12: Gestión de imágenes		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	8 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
3	Diseño de interfaz gráfica imagen (Formulario ingresar, editar y eliminar)	20 horas
4	Mapeo Objeto-Relacional entidad imagen.	6 horas
5	Creación de archivos <i>PHP</i> .	6 horas
		Total: 48 horas

Tabla 122: Tareas de Gestión imágenes

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 12
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 8 horas
Fecha inicio: 02/3/13	Fecha fin: 02/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio e imagen. - Un usuario puede seleccionar una provincia, ciudad, categoría, sitio y asignarle uno o varias imágenes. - Se migró todas las claves primarias hacia la tabla imagen. 	

Tabla 123: Gestión de imágenes – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 12
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 8 horas
Fecha inicio: 03/3/13	Fecha fin: 03/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . <i>crealmg.sql:</i> Permite crear la tabla imagen. <i>primimg.sql:</i> Creación de claves primarias. <i>foreimg.sql:</i> Creación de claves foráneas. <i>inselimg.sql:</i> Inserción de datos hacia la tabla imagen, pues en el momento no se dispone de formularios para administrar imágenes.	

Tabla 124: Gestión de imágenes – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica imagen (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 12
Nombre tarea: Diseño de interfaz gráfica imagen (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 04/3/13	Fecha fin: 05/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de imágenes, mediante la construcción de hojas de estilo CSS y archivos <i>JAVASCRIPT</i> . <i>Formulario Ingresar video:</i> Permite ingresar los campos de texto respectivos. <i>Formulario Editar video:</i> Permite editar los campos de texto seleccionados. Creación de menú provincia, ciudad y sitio fusionando código <i>PHP</i> , JavaScript y <i>HTML</i> . Funciones JavaScript, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor. <i>validarCamposImagen():</i> Valida los campos de formulario ingresar y editar. <i>insertImagen():</i> Recibe como parámetro los campos de una imagen, y los envía hacia el servidor. <i>updalmagen():</i> Recibe como parámetros los campos a editarse, y los enviar hacia el servidor. <i>editImagen():</i> Recibe como parámetro la clave primaria de una imagen, y lo envía hacia el servidor. <i>dellmagen():</i> Recibe como parámetros las claves primarias de una entidad imagen, y la elimina de la base de datos. <i>conslmagen():</i> Permite consultar las imágenes. <i>paginImagen():</i> Permite paginar el listado de imágenes.	

Tabla 125: Gestión de imágenes – Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad imagen

Número tarea: 4	Número historia: 12
Nombre tarea: <i>Mapeo Objeto-Relacional entidad imagen.</i>	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 06/3/13	Fecha fin: 06/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad imagen, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla <i>imagen</i>, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>imagen.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>imagen ()</i>: Constructor que nos permite instanciar el objeto <i>imagen</i>. <i>insertar()</i>: Permite ingresar una <i>imagen</i> hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla <i>imagen</i>, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las claves primarias de una entidad <i>imagen</i> y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 126: Gestión de imágenes – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos *PHP*

Número tarea: 5	Número historia: 12
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 07/3/13	Fecha fin: 07/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p><i>imagenCons.php</i>: Instanciamos el objeto <i>video</i> para poder imprimir sus campos. <i>imagenFormInsert.php</i>: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. <i>imagenFormUpdate.php</i>: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. <i>imagenFuncion.php</i>: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 127: Gestión de imágenes – Tarea Nro. 5

Fuente: Propia

- **Gestión de festivos**

Historia 13: Gestión de festivos		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	8 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
3	Diseño de interfaz gráfica festivo (Formulario ingresar, editar y eliminar)	20 horas
4	Mapeo Objeto-Relacional entidad festivo.	6 horas
5	Creación de archivos <i>PHP</i> .	6 horas
		Total: 48 horas

Tabla 128: Tareas de Gestión de festivos

Fuente: Propia.

1. Diseño de bases de datos

Número tarea: 1	Número historia: 13
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 8 horas
Fecha inicio: 10/3/13	Fecha fin: 10/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio y festivo. - Un usuario puede seleccionar una provincia, ciudad, categoría, sitio y asignarle uno o varios días festivos. - Se migró todas las claves primarias hacia la tabla festivo. 	

Tabla 129: Gestión de festivos – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 13
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 8 horas
Fecha inicio: 11/3/13	Fecha fin: 11/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . creaFest.sql: Permite crear la tabla festivo. primFest.sql: Creación de claves primarias. foreFest.sql: Creación de claves foráneas. inseFest.sql: Inserción de datos hacia la tabla festivo, pues en el momento no se dispone de formularios para administrar festivos.	

Tabla 130: Gestión de festivos – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica festivo (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 13
Nombre tarea: Diseño de interfaz gráfica festivo (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 12/3/13	Fecha fin: 13/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de días festivos y eventos, mediante la construcción de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería TWITTER BOOTSTRAP MASTER, para mejorar el diseño de nuestros componentes HTML. Formulario Ingresar festivo: Permite ingresar los campos de texto respectivos. Formulario Editar festivo: Permite editar los campos de texto seleccionados. Creación de menú provincia, ciudad, sitio fusionando código PHP y HTML. Diseño de Calendarios desplegados para la selección de fechas. Funciones JAVASCRIPT: validarCamposFestivo(): Valida los campos de formulario ingresar y editar. insertFestivo(): Recibe como parámetro los campos de un festivo, y los envía hacia el servidor. updaFestivo(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editFestivo(): Recibe como parámetro la clave primaria de un festivo, y lo envía hacia el servidor. delFestivo(): Recibe como parámetro las claves primarias y foráneas de un festivo, y los envía hacia el servidor. consFestivo(): Permite consultar los festivos. paginFestivo(): Permite paginar el listado de festivos.	

Tabla 131: Gestión de festivos – Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad Festivo

Número tarea: 4	Número historia: 13
Nombre tarea: Mapeo Objeto-Relacional entidad <i>festivo</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 14/3/13	Fecha fin: 14/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad festivo, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla <i>festivo</i>, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>festivo.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>festivo()</i>: Constructor que nos permite instanciar el objeto <i>festivo</i>. <i>insertar()</i>: Permite ingresar un <i>festivo</i> hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla <i>festivo</i>, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las claves primarias de una entidad <i>festivo</i> y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 132: Gestión de festivos – Tarea Nro. 4.

Fuente: Propia.

5. Creación de archivos *PHP*

Número tarea: 5	Número historia: 13
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 15/3/13	Fecha fin: 15/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p><i>festivoCons.php</i>: Instanciamos el objeto <i>festivo</i> para poder imprimir sus campos. <i>festivoFormInsert.php</i>: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. <i>festivoFormUpdate.php</i>: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. <i>festivoFuncion.php</i>: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 133: Gestión de festivos – Tarea Nro. 5

Fuente: Propia

- **Gestión de gastronomías**

Historia 14: Gestión de gastronomías		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	8 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	8 horas
3	Diseño de interfaz gráfica gastronomía (Formulario ingresar, editar y eliminar)	20 horas
4	Mapeo Objeto-Relacional entidad gastronomía.	6 horas
5	Creación de archivos <i>PHP</i> .	6 horas
		Total: 48 horas

Tabla 134: Tareas de gestión de gastronomías

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 14
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 8 horas
Fecha inicio: 16/3/13	Fecha fin: 16/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño de bases de datos</p> <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio y gastronomía. - Un usuario puede seleccionar una provincia, ciudad, categoría, sitio y asignarle uno o varias gastronomías. - Se migró todas las claves primarias hacia la tabla gastronomía. 	

Tabla 135: Gestión de gastronomías – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 14
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea : Administración de Bases de datos	Tiempo estimado: 8 horas
Fecha inicio: 17/3/13	Fecha fin: 17/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i> . creaGast.sql: Permite crear la tabla gastronomía. primGast.sql: Creación de claves primarias. foreGast.sql: Creación de claves foráneas. inseGast.sql: Inserción de datos hacia la tabla gastronomía, pues en el momento no se dispone de formularios para administrar gastronomías.	

Tabla 136: Gestión de gastronomías – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica gastronomía (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 14
Nombre tarea: Diseño de interfaz gráfica gastronomía (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 18/3/13	Fecha fin: 19/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de gastronomías, mediante la construcción de hojas de estilo CSS y archivos JAVASCRIPT, además utilizamos la librería TWITTER BOOTSTRAP MASTER, para mejorar el diseño de nuestros componentes HTML. Formulario Ingresar gastronomía: Permite ingresar los campos de texto respectivos. Formulario Editar gastronomía: Permite editar los campos de texto seleccionados. Creación de menú provincia, ciudad y sitio fusionando código PHP y HTML. Funciones JavaScript, que actúan a los eventos respectivos de cada formulario, en cada función que requiera un acceso al servidor, fue necesario instanciar una comunicación AJAX asíncrona con el servidor. validarCamposGastronomía(): Valida los campos de formulario ingresar y editar. insertGastronomía(): Recibe como parámetro los campos de una gastronomía, y los envía hacia el servidor. updaGastronomía(): Recibe como parámetros los campos a editarse, y los envía hacia el servidor. editGastronomía(): Recibe como parámetro la clave primaria de una gastronomía, y lo envía hacia el servidor. delGastronomía(): Recibe como parámetro las claves primarias y foráneas de una gastronomía, y los envía hacia el servidor. consGastronomía(): Permite consultar las gastronomías. paginGastronomía(): Permite paginar el listado de gastronomías	

Tabla 137: Gestión de gastronomías – Tarea Nro. 3

Fuente: Propia

4. Mapeo Objeto-Relacional entidad gastronomía

Número tarea: 4	Número historia: 14
Nombre tarea: Mapeo Objeto-Relacional entidad <i>gastronomía</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio:20/3/13	Fecha fin:20/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad <i>gastronomía</i>, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla <i>gastronomía</i>, en su respectivo objeto encapsulado, utilizando una clase <i>PHP</i> denominada <i>gastronomía.php</i>, para poder instanciarla y acceder a sus respectivos atributos y métodos.</p> <p>Métodos implementados: <i>gastronomía ()</i>: Constructor que nos permite instanciar el objeto <i>gastronomía</i>. <i>insertar()</i>: Permite ingresar una <i>gastronomía</i> hacia la base de datos, utilizando el comando <i>INSERT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla <i>gastronomía</i>, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las claves primarias de una entidad <i>gastronomía</i> y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 138: Gestión de gastronomías – Tarea Nro. 4

Fuente: Propia

5. Creación de archivos PHP

Número tarea: 5	Número historia: 14
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio:21/3/13	Fecha fin:21/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p>Gastronomía Cons.php: Instanciamos el objeto <i>gastronomía</i> para poder imprimir sus campos. gastronomíaFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. gastronomíaFormUpdate.php: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. gastronomíaFuncion.php: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 139: Gestión de gastronomías – Tarea Nro. 5

Fuente: Propia

- **Gestión de rutas**

Historia 15: Gestión de rutas		
Número	Nombre	Tiempo estimado
1	Diseño de bases de datos	5 horas
2	Creación de scripts utilizando lenguaje <i>SQL</i> .	5 horas
3	Diseño de interfaz gráfica ruta (Formulario ingresar, editar y eliminar)	17 horas
4	Configuración de un visor de mapas.	25
5	Mapeo Objeto-Relacional entidad ruta.	8 horas
6	Creación de archivos <i>PHP</i> .	7 horas
		Total: 67 horas

Tabla 140: Tarea de gestión de rutas

Fuente: Propia

1. Diseño de bases de datos

Número tarea: 1	Número historia: 15
Nombre tarea: Diseño de bases de datos	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 5 horas
Fecha inicio: 22/3/13	Fecha fin: 22/3/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de bases de datos <ul style="list-style-type: none"> - Diseño diagrama entidad relación entre entidades: Provincia, ciudad, categoría, sitio y ruta. - Un usuario puede seleccionar una provincia, ciudad, sitio y asignarle una ruta. - Se migró todas las claves primarias hacia la tabla rutas. 	

Tabla 141: Gestión de rutas – Tarea Nro. 1

Fuente: Propia

2. Creación de scripts utilizando lenguaje SQL

Número tarea: 2	Número historia: 15
Nombre tarea: Creación de scripts utilizando lenguaje SQL.	
Tipo de tarea: Administración de Bases de datos geográfica.	Tiempo estimado: 5 horas
Fecha inicio: 23/3/13	Fecha fin: 23/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de scripts SQL, para gestionar la base de datos desde una consola, utilizando el gestor de consultas <i>pgAdmin III</i>. creaRuta.sql: Creación de la tabla ruta. primRuta.sql: Claves primarias de las tablas ruta. foreRuta.sql: Claves foráneas de las tala ruta. inseRuta.sql: Generación de datos, para las respectivas tablas creadas. geomRuta.sql: Creación del campo <i>geometry</i>, que almacenara la ruta de un sitio, previa transformación de sistema de referencias 4326.</p>	

Tabla 142: Gestión de rutas – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica ruta (Formulario ingresar, editar y eliminar)

Número tarea: 3	Número historia: 15
Nombre tarea: Diseño de interfaz gráfica ruta (Formulario ingresar, editar y eliminar)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 17 horas
Fecha inicio:24/3/13	Fecha fin:25/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de diseño, para la pantalla administración de rutas, mediante la construcción de hojas de estilo CSS y archivos <i>JAVASCRIPT</i>. Formulario Ingresar ruta: Permite ingresar los campos de un sitio. Formulario Editar ruta: Permite editar los sitios de una provincia y ciudad seleccionada. Creación de menú provincia, ciudad y sitio fusionando código <i>PHP</i> y <i>HTML</i>. Funciones JavaScript: validarCamposRuta(): Valida los campos de formulario ingresar y Editar. insertRuta(): Recibe como parámetro los campos de la ruta, y los envía hacia el servidor. updaRuta(): Recibe como parámetros los campos a editarse, y los enviar hacia el servidor. editRuta(): Recibe como parámetro la clave primaria de una ruta, y lo envía hacia el servidor. delRuta(): Recibe como parámetros las claves primarias y de una entidad ruta, y la elimina de la base de datos. consRuta(): Permite seleccionar una ruta a la vez, para ingresarla. paginRuta(): Paginar el reporte ruta.</p>	

Tabla 143: Gestión de rutas – Tarea Nro. 3

Fuente: Propia

4. Configuración de un visor de mapas

Número tarea: 4	Número historia: 15
Nombre tarea: Configuración de un visor de mapas.	
Tipo de tarea: Programación <i>GIS</i> .	Tiempo estimado: 25 horas
Fecha inicio: 26/3/13	Fecha fin: 27/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de configurar un visor de mapas. Fue necesaria la importación de <i>OPENLAYERS</i> para poder visualizar un mapa interactivo en nuestro navegador web, con las opciones habilitadas que nos permitirá dibujar puntos.</p> <p>Funciones JavaScript implementadas: <i>init()</i>: definimos un objeto <i>OpenLayers.Map</i>, el cual permitirá imprimir partes de un mapa en nuestro navegador. <i>borrarFeatures()</i>: borramos todas las capas creadas por el usuario. <i>obtenerRuta()</i>: obtenemos la ruta de nuestro mapa, y realizamos la respectiva transformación hacia el sistema de referencia 4326, que es el más utilizado por la comunidad de desarrolladores <i>GIS</i>. <i>agregarCapaRuta()</i>: recibe como parámetros los vértices de una ruta y gráfica la ruta en el visor de mapas.</p>	

Tabla 144: Gestión de rutas – Tarea Nro. 4

Fuente: Propia

5. Mapeo Objeto-Relacional entidad ruta

Número tarea: 5	Número historia: 15
Nombre tarea: Mapeo Objeto-Relacional entidad <i>ruta</i> .	
Tipo de tarea: Programación <i>GIS</i> .	Tiempo estimado: 8 horas
Fecha inicio: 28/3/13	Fecha fin: 28/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de mapeo de la entidad ruta, aplicando el paradigma de la <i>PROGRAMACIÓN ORIENTADA A OBJETOS</i>, para poder reutilizar el código fuente y acceder hacia la información de una manera rápida y eficiente. Convertimos nuestra tabla ruta, en su respectivo objeto encapsulado, utilizando una clase PHP denominada <i>ruta.php</i>.</p> <p>Métodos implementados: <i>ruta()</i>: Constructor que nos permite instanciar el objeto ruta. <i>insertar()</i>: Permite ingresar un sitio hacia la base de datos, utilizando el comando <i>INSERT</i> y la función <i>GEOMETRYFROMTEXT</i>. <i>actualiza()</i>: Permite editar los campos de la tabla ruta, a excepción de la clave primaria, utilizando en comando <i>UPDATE</i>. <i>eliminar()</i>: Recibe como parámetros, las claves primarias de una entidad ruta y la elimina de la base de datos, utilizando el comando <i>DELETE</i>.</p>	

Tabla 145: Gestión de rutas – Tarea Nro. 5

Fuente: Propia

6. Creación de archivos *PHP*

Número tarea: 6	Número historia: 15
Nombre tarea: Creación de archivos <i>PHP</i> .	
Tipo de tarea : Programación	Tiempo estimado: 7 horas
Fecha inicio: 29/3/13	Fecha fin: 30/3/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la información del servidor en el lado de cliente, utilizando los formularios ya diseñados, en el área de diseño y maquetado.</p> <p>rutaCons.php: Instanciamos el objeto ruta para poder acceder e imprimir sus campos. rutaFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. rutaFormUpdate: Formulario encargado de enviar las variables hacia el servidor, para editar los campos respectivos. rutaFuncion: Incorporamos funciones para ingresar, editar y eliminar.</p>	

Tabla 146: Gestión de rutas – Tarea Nro. 6

Fuente: Propia

3.4.3 Pruebas Iteración II

1. Especificación de Prueba: Gestión de provincias

Historia de usuario 6

Versión <1.5>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
07/1/2013	1.5	Revisión	Jairo Morales
07/1/2013	1.6	Modificación	Jairo Morales

Tabla 147: Iteración II, Historial de revisiones Nro. 1

Fuente: Propia

En esta historia se verifica, que la introducción de los datos hacia el sistema, cumpla las especificaciones establecidas en la Historia 6, por parte del administrador de provincias.

Si los campos introducidos no corresponden, se imprimen alertas definidas, que informan la situación actual de la operación.

Se asume que el usuario ya está registrado y tiene acceso hacia el sistema, caso contrario deberá registrarse, en cuestión de horas un especialista atenderá su petición y habilitará el respectivo usuario.

Registro de una provincia con datos correcto

Descripción

Los valores introducidos por el administrador deben cumplir, con las reglas establecidas.

Condiciones de ejecución

El administrador de provincias deberá estar previamente creado y habilitado, para poder acceder hacia el sistema, posteriormente procederá a la creación de provincias.

Entrada

Una vez dentro del sistema, deberá presionar sobre el botón agregar provincia.

Los campos a ingresar son: nombre, descripción, capital y población.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, capital [5-32] caracteres, población [5-10] caracteres, región [5-15] caracteres

Resultado esperado

Tras la introducción los datos de una provincias, si los datos procesados han sido correctos, se registra una provincia.

El reporte sí permite eliminar una o varias provincias, en caso de que no exista integridad referencial.

Evaluación de la prueba

Prueba satisfactoria.

Registro de provincia con datos incorrectos

Descripción

Si los datos ingresados de una provincia no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

Al intentar introducir un valor que no tenga el formato del campo adecuado, nos generará una alerta moderada, que especificara el número ideal de caracteres a ingresarse por campo.

Condiciones de ejecución

El usuario deberá estar previamente creado y habilitado, para poder administrar las provincias.

Entrada

Los campos a editar son: nombre, descripción, capital, y población.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, capital [5-32] caracteres, población [5-10] caracteres, región [5-15] caracteres

✚ Resultado esperado

La interfaz de administración de provincias, sí edita los campos e imprime las alertas respectivas, en caso de que el usuario ingrese campos que no concuerdan.

✚ Evaluación de la prueba

Prueba satisfactoria.

2. Especificación de Prueba: Gestión de ciudades

Historia de usuario 7

Versión <1.6>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15/1/2013	1.6	Revisión	Jairo Morales
15/1/2013	1.7	Modificación	Jairo Morales

Tabla 148: Iteración II, Historial de revisiones Nro. 2

Fuente: Propia

En esta historia se verifica, que el administrador de ciudades, realice el registro de una ciudad correctamente.

De esta manera se controlará, a que los campos a cumplan con las especificaciones, él usuario tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla ciudad.

Si los campos introducidos no corresponden, se imprimirá alertas definidas, que informan la situación de la operación.

Registrar la información correcta de una ciudad

✚ Descripción

El administrador de la aplicación creará y editará una ciudad.

Una vez que los datos se registren de forma correcta se procede a ingresarla hacia a base de datos.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar ciudades.

Para que se proceda a la creación de ciudades, debe al menos existir una provincia ya creada, a la cual se le asignarán ciudades.

✚ Entrada

El usuario deberá seleccionar una sola provincia del menú respectivo.

Debe llenar los campos obligatorios en el formulario, caso contrario ingresar “no aplicable”.

Los tipos de datos están brevemente especificados, en los campos de texto a ingresarse.

El nombre, descripción y población, deberán tener un número de caracteres preestablecidos dentro de en un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, población [5-10] caracteres.

✚ Resultado esperado

Tras la creación de una ciudad. Si el proceso ha sido correcto, la ciudad se crea.

El reporte sí permite eliminar una o varias ciudades, en caso de que no exista integridad referencial.

✚ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de una ciudad

✚ Descripción

Si los datos ingresados por el usuario administrador de ciudades no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

Condiciones de ejecución

El usuario deberá tener los permisos, para administrar ciudades.

✚ Entrada

El usuario deberá seleccionar una sola provincia del menú respectivo.

Debe llenar los campos obligatorios en el formulario, caso contrario ingresar “no aplicable”.

Los tipos de datos están brevemente especificados, en los campos de texto a ingresarse.

El nombre, descripción y población, deberán tener un número de caracteres preestablecidos dentro de en un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, población [5-10] caracteres.

✚ Resultado esperado

La interfaz de administración de ciudades, imprime las alertas respectivas en caso de que el usuario ingrese campos que no concuerdan.

✚ Evaluación de la prueba

Prueba satisfactoria.

3. Especificación de Prueba: Gestión de categorías.

Historia de usuario 8

Versión <1.7>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
24/1/2013	1.7	Revisión	Jairo Morales
24/1/2013	1.8	Modificación	Jairo Morales

Tabla 149: Iteración II, Historial de revisiones Nro. 3

Fuente: Propia

En esta historia se verifica, que el administrador de categorías, realice el registro de una categoría correctamente.

De esta manera se controlará, a que los campos a cumplan con las especificaciones, él usuario tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla categoría.

Si los campos introducidos no corresponden, se imprimirá alertas definidas, que informan la situación de la operación.

Registrar la información correcta de una categoría.

✚ Descripción

El administrador de la aplicación creará y editará una categoría.

Una vez que los datos se registren de forma correcta se procede a ingresarla hacia a base de datos.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar categorías.

Para que se proceda a la creación de categorías.

✚ Entrada

Debe llenar los campos obligatorios en el formulario, caso contrario ingresar “no aplicable”.

Los tipos de datos están brevemente especificados, en los campos de texto a ingresarse.

El nombre, descripción, deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32], descripción [5-256]

✚ Resultado esperado

Tras la creación de una categoría. Si el proceso ha sido correcto, la categoría se creara.

El reporte sí permite eliminar una o varias categorías, en caso de que no exista integridad referencial.

✚ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de una categoría.

✚ Descripción

Si los datos ingresados por el usuario administrador de categorías no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar categorías.

✚ Entrada

Debe llenar los campos obligatorios en el formulario, caso contrario ingresar “no aplicable”.

Los tipos de datos están brevemente especificados, en los campos de texto a ingresarse.

El nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32], descripción [5-256]

✚ Resultado esperado

La interfaz de administración de categorías, imprime las alertas respectivas en caso de que el usuario ingrese campos que no concuerdan.

✚ Evaluación de la prueba

Prueba satisfactoria.

4. Especificación de Prueba: Gestión de sitios.

Historia de usuario 9

Versión <1.8>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
06/2/2013	1.8	Revisión	Jairo Morales
07/2/2013	1.9	Modificación	Jairo Morales

Tabla 150: Iteración II, Historial de revisiones Nro. 4

Fuente: Propia

En esta historia se verifica, que el administrador de sitios, seleccione una provincia, ciudad, para asignarle uno o varios sitios.

El administrador de historias, tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla sitios.

Registrar la información correcta de un sitio.

Descripción

El administrador de sitios, asignará a la vez una provincia, ciudad y categoría desde un menú descriptivo.

Condiciones de ejecución

El usuario deberá tener los permisos, para administrar sitios.

Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú categorías, se debe seleccionar una categoría.

El nombre, descripción, página web, cuenta Facebook, twitter, dirección y teléfono, de un sitio, deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres , descripción [5-256] caracteres, página web [5-32] caracteres, mail [5-32] caracteres, facebook [5-32] caracteres, twitter[5-32] caracteres dirección [5-64] caracteres, teléfono [5-64] caracteres.

Resultado esperado

Tras la selección de las entidades se procedió a ingresar un sitio. Si el proceso ha sido correcto, el sitio se crea.

Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de un sitio

+ Descripción

Si los datos ingresados por el administrador de sitios no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

+ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar sitios.

+ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú categorías, se debe seleccionar una categoría.

El nombre, descripción, página web, cuenta facebook, twitter, dirección y teléfono, de un sitio, deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres , descripción [5-256] caracteres, página web [5-32] caracteres, mail [5-32] caracteres, facebook [5-32] caracteres, twitter[5-32] caracteres dirección [5-64] caracteres, teléfono [5-64] caracteres.

+ Resultado esperado

El formulario ingresa y edita un sitio, además imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

+ Evaluación de la prueba

Prueba satisfactoria.

5. Especificación de Prueba: Gestión de historias

Historia de usuario 10

Versión <1.9>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
18/2/2013	1.9	Revisión	Jairo Morales
19/2/2013	2.0	Modificación	Jairo Morales

Tabla 151: Iteración II, Historial de revisiones Nro. 5

Fuente: Propia

En esta historia se verifica, que el administrador de historias, seleccione una provincia, ciudad y sitio para asignarle una o varias historias.

El administrador de historias, tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla historia.

Registrar la información correcta de una historia

Descripción

El administrador de historias, asignará a la vez una provincia, ciudad y sitio desde un menú descriptivo.

Condiciones de ejecución

El usuario deberá tener los permisos, para administrar historias.

Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

El nombre y descripción de la historia, deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-1024] caracteres.

Resultado esperado

Tras la selección de las entidades se procedió a ingresar una historia. Si el proceso ha sido correcto, la historia se crea.

Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de una historia

Descripción

Si los datos ingresados por el administrador de una historia no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

Condiciones de ejecución

El usuario deberá tener los permisos, para administrar historias.

Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

El nombre y descripción, un historia, deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-1024] caracteres.

✚ Resultado esperado

El formulario Ingresar y editar historia, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

✚ Evaluación de la prueba

Prueba satisfactoria.

6. Especificación de Prueba: Gestión de URL'S hacia YouTube

Historia de usuario 11

Versión <2.0>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
01/3/2013	2.0	Revisión	Jairo Morales
01/3/2013	2.1	Modificación	Jairo Morales

Tabla 152: Iteración II, Historial de revisiones Nro. 6

Fuente: Propia

En esta historia se verifica, que el administrador de historias, seleccione una provincia, ciudad y sitio para asignarle uno o varios enlaces hacia *YouTube*.

El administrador de videos, tendrá la facultad de realizar el respectivo *CRUD* (Crear, Leer, Actualizar y Eliminar) de la tabla video.

Registrar la información correcta de un video

✚ Descripción

El administrador de videos, asignará a la vez una provincia, ciudad y sitio desde un menú descriptivo.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar videos.

✚ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

El nombre, descripción y URL del video, deberán tener un número de caracteres preestablecidos dentro de en un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, *URL* [5-64] caracteres.

✚ Resultado esperado

Tras la selección de las entidades se procedió a ingresar un video. Si el proceso ha sido correcto, el video se crea.

✚ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de un video

✚ Descripción

Si los datos ingresados por el administrador de videos no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar videos.

✚ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

El nombre, descripción y URL de un video, deberán tener un número de caracteres preestablecidos dentro de en un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, *URL* [5-64] caracteres.

✚ Resultado esperado

El formulario ingresar y editar video, sí imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

✚ Evaluación de la prueba

Prueba satisfactoria.

7. Especificación de Prueba: Gestión de imágenes

Historia de usuario 12

Versión <2.1>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
08/3/2013	2.1	Revisión	Jairo Morales
09/3/2013	2.2	Modificación	Jairo Morales

Tabla 153: Iteración II, Historial de revisiones Nro. 7

Fuente: Propia

En esta historia se verifica, que el administrador de imágenes, seleccione una provincia, ciudad y sitio para asignarle una o varias imágenes.

El administrador de imágenes, tendrá la facultad de ingresar y editar una imagen.

Registrar la información correcta de una imagen

+ Descripción

El administrador de imágenes, asignará a la vez una provincia, ciudad y sitio desde un menú descriptivo.

+ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar imágenes.

+ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre, descripción e imagen, deberán tener un número de caracteres preestablecidos dentro de en un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso menor a 200 KB, preferencias 500X500 PX.

+ Resultado esperado

Tras la selección de las entidades se procedió a ingresar una imagen. Si el proceso ha sido correcto, la imagen se ingresa.

+ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de una imagen

+ Descripción

Si los datos ingresados por el administrador de imágenes no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

+ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar imágenes.

✚ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre, descripción e imagen, deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso menor a 200 KB, preferencias 500X500 PX.

✚ Resultado esperado

El formulario ingresar y editar imagen, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

✚ Evaluación de la prueba

Prueba satisfactoria.

8. Especificación de Prueba: Gestión de festivos

Historia de usuario 13

Versión <2.2>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
15/3/2013	2.2	Revisión	Jairo Morales
15/3/2013	2.3	Modificación	Jairo Morales

Tabla 154: Iteración II, Historial de revisiones Nro. 8

Fuente: Propia

En esta historia se verifica, que el administrador de festivos, seleccione una provincia, ciudad y sitio para asignarle uno o vario días festivos.

El administrador de festivos, tendrá la facultad de realizar el respectivo CRUD (Crear, Leer, Actualizar y Eliminar) de la tabla festivo.

Registrar la información correcta de un festivo

+ Descripción

El administrador de festivos, asignará a la vez una provincia, ciudad y sitio desde un menú descriptivo.

+ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar festivos.

+ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango, además se deberá seleccionar una fecha inicial y final.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX, la fecha inicial debe ser menor que la fecha mayor.

+ Resultado esperado

Tras la selección de las entidades se procedió a ingresar un festivo. Si el proceso ha sido correcto, el evento se crea.

+ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de un festivo

+ Descripción

Si los datos ingresados por el administrador de festivos no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

+ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar festivos.

+ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango, además se deberá seleccionar una fecha inicial y final.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX, la fecha inicial debe ser menor que la fecha mayor.

 Resultado esperado

El formulario Ingresar y editar festivo, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

 Evaluación de la prueba

Prueba satisfactoria.

9. Especificación de Prueba: Gestión de gastronomías

Historia de usuario 14

Versión <2.3>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
21/3/2013	2.3	Revisión	Jairo Morales
21/3/2013	2.4	Modificación	Jairo Morales

Tabla 155: Iteración II, Historial de revisiones Nro. 9

Fuente: Propia

En esta historia se verifica, que el administrador de gastronomías, seleccione una provincia, ciudad y sitio para asignarle una o varias gastronomías que puede corresponder a toda la cocina en general que se oferte.

El administrador de festivos, tendrá la facultad ingresar y eliminar una imagen.

Registrar la información correcta de una gastronomía.

 Descripción

El administrador de gastronomías, asignará a la vez una provincia, ciudad y sitio desde un menú descriptivo.

 Condiciones de ejecución

El usuario deberá tener los permisos, para administrar gastronomías.

 Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango.

El usuario deberá cargar una imagen que corresponda a tipo de gastronomía.

Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX.

✚ Resultado esperado

Tras la selección de las entidades se procedió a ingresar una gastronomía. Si el proceso ha sido correcto, la gastronomía se crea.

✚ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de una gastronomía.

✚ Descripción

Si los datos ingresados por el administrador de gastronomías no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar gastronomías.

✚ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango.

El usuario deberá cargar una imagen que corresponda a tipo de gastronomía.

Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX.

✚ Resultado esperado

El formulario ingresar y editar gastronomía, imprime las alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

✚ Evaluación de la prueba

Prueba satisfactoria.

10. Especificación de Prueba: Gestión de rutas.

Historia de usuario 15

Versión <2.5>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
01/4/2013	2.5	Revisión	Jairo Morales
02/4/2013	2.6	Modificación	Jairo Morales

Tabla 156: Iteración II, Historial de revisiones Nro. 10

Fuente: Propia

En esta historia se verifica, que el administrador de la aplicación, seleccione una provincia, ciudad, categoría, sitio y pueda asignarles una ruta con su respectivo nombre y descripción.

El administrador de la aplicación, tendrá la facultad de realizar el respectivo *CRUD*(Crear, Leer, Actualizar, Eliminar) de una ruta.

Registrar la información correcta de rutas.

✚ Descripción

El administrador de rutas, asignará a la vez una provincia, ciudad y sitio desde un menú descriptivo.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar las rutas.

✚ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-64] caracteres.

✚ Graficar una correspondiente.

Resultado esperado

Si el proceso ha sido correcto, los campos como la ruta graficada se registraran en las base de datos

✚ Evaluación de la prueba

Prueba satisfactoria.

Registrar la información incorrecta de rutas.

✚ Descripción

Si los datos ingresados por el administrador no concuerdan, se imprimirán alertas definidas, que indiquen el estado de los campos ingresados.

✚ Condiciones de ejecución

El usuario deberá tener los permisos, para administrar rutas.

✚ Entrada

Del menú provincias, se debe seleccionar una provincia

Del menú ciudades, se debe seleccionar una ciudad.

Del menú sitios, se debe seleccionar un sitio.

Los campos: Nombre y descripción deberán tener un número de caracteres preestablecidos dentro de un rango.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-64] caracteres.

Graficar una correspondiente.

✚ Resultado esperado

Los formularios sí ingresan y editan las rutas, además se imprimen alertas respectivas, en caso de que el número de caracteres registrado en un campo sea inferior o superior.

✚ Evaluación de la prueba

Prueba satisfactoria.

3.4.4 Resultado de la 2ª Iteración de construcción

✚ Plan de entrega inicial.

Historia 6: Gestión de provincias

Historia 7: Gestión de ciudades

Historia 8: Gestión de categorías

Historia 9: Gestión de sitios

Historia 10: Gestión de historias

Historia 11: Gestión de *URL'S* hacia *YouTube*

Historia 12: Gestión de imágenes

Historia 13: Gestión de festivales

Historia 14: Gestión de gastronomías

Historia 15: Gestión de rutas

Versión del sistema: 2.6

✚ Modificación de requerimientos.

Consta de 10 historias de usuario:

Historia 6: Gestión de provincias (Se mantiene).

Historia 7: Gestión de ciudades (Se mantiene).

Historia 8: Gestión de categorías (Se mantiene).

Historia 9: Gestión de sitios (Se mantiene).

Historia 10: Gestión de historias (Se mantiene).

Historia 11: Gestión de URL'S hacia YouTube (Se mantiene).

Historia 12: Gestión de imágenes (Se mantiene).

Historia 13: Gestión de festivos (Se mantiene).

Historia 14: Gestión de gastronomías (Se mantiene).

Historia 15: Gestión de rutas (Se mantiene).

3.4.5 Memo de la versión, Iteración II

Historia 6: Gestión de provincias (5 días)

Ingresar provincia: El usuario deberá presionar sobre el botón agregar provincia, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, capital [5-32] caracteres, población [5-10] caracteres, región [5-15] caracteres.

The image shows two parts of a web application interface. On the left is a table with the following data:

Nombre	Acciones	Link
Azuay		Ciudades
Bolivar		Ciudades
Cañar		Ciudades
Carchi		Ciudades
Imbabura		Ciudades

Below the table are navigation buttons: [Primero](#), [Pagina 1/1](#), [Ultimo](#).

On the right is a modal window titled "Ingresar Provincia" with the following fields:

- Nombre:
- Descripcion:
- Capital:
- Poblacion:
- Región natural:

At the bottom of the modal are two buttons: [Guardar](#) and [Cerrar](#).

Ilustración 36: Formulario para ingresar una entidad provincia

Fuente: Propia

Editar provincia: El usuario deberá presionar sobre la imagen editar provincia, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, capital [5-32] caracteres, población [5-10] caracteres, región [5-15] caracteres.

Eliminar provincia: presionar sobre la imagen eliminar provincia.

The screenshot shows the 'Administración de Provincias' interface. On the left, there is a table with columns 'Nombre', 'Acciones', and 'Link'. The table lists provinces: Azuay, Bolívar, Cañar, Carchi, and Imbabura. Each row has an edit icon (wrench and pencil) and a link labeled 'Ciudades'. Below the table are navigation buttons: 'Primero', 'Pagina 1/1', and 'Ultimo'. On the right, the 'Editar Provincia' modal form is open for 'Imbabura'. It contains fields for 'Nombre' (Imbabura), 'Descripción' (La población imbabureña varía en cuanto a costumbres y manera de ser se refiere, por ejemplo con liberales y tradicionales personas en Otavalo, con conservadores y bohemios), 'Capital' (Ibarra), 'Población' (398244), and 'Región' (Sierra). At the bottom of the modal are 'Guardar' and 'Cerrar' buttons.

Ilustración 37: Formulario para editar una entidad provincia

Fuente: Propia

Historia 7: Gestión de ciudades (7 días)

Ingresar ciudad: El usuario deberá presionar sobre el botón agregar ciudad, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, población [5-10] caracteres.

The screenshot shows the 'Administración de Ciudades' interface. On the left, there is a table with columns 'Nombre', 'Acciones', and 'Link'. The table lists cities: Ibarra, Otavalo, and Cotacachi. Each row has an edit icon (wrench and pencil) and a link labeled 'Sitios'. Below the table are navigation buttons: 'Primero', 'Pagina 1/1', and 'Ultimo'. On the right, the 'Ingresar Ciudad' modal form is open for 'Ibarra'. It contains fields for 'Nombre' (Ibarra), 'Descripción' (Es un lugar..), and 'Población' (5433). At the bottom of the modal are 'Guardar' and 'Cerrar' buttons.

Ilustración 38: Formulario para ingresar una entidad ciudad

Fuente: Propia

Editar ciudad: El usuario deberá presionar sobre la imagen editar ciudad, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, población [5-10] caracteres.

Eliminar ciudad: Presionar sobre la imagen eliminar ciudad.

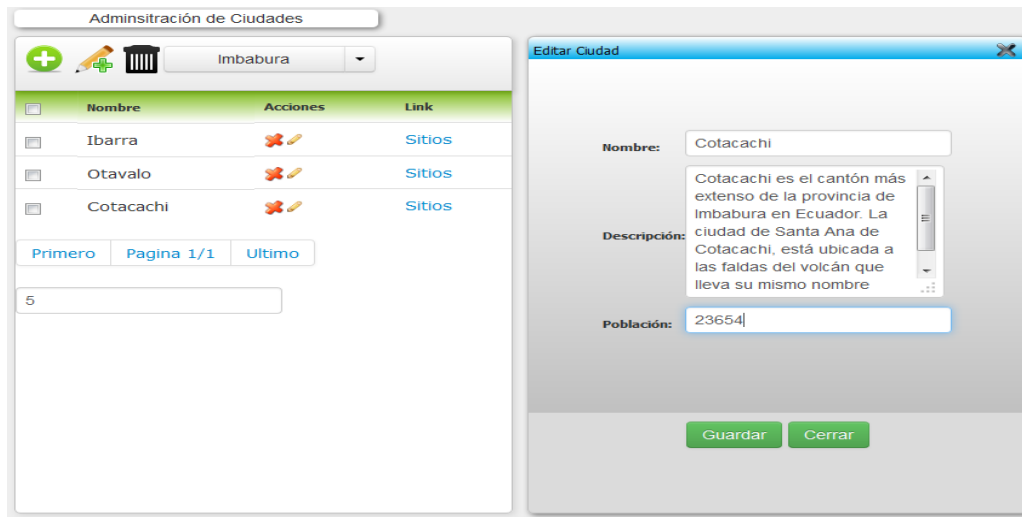


Ilustración 39: Formulario para editar una entidad ciudad

Fuente: Propia

Historia 8: Gestión de categorías (6 días)

Ingresar categoría: El usuario deberá presionar sobre el botón agregar categoría, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32], descripción [5-256]

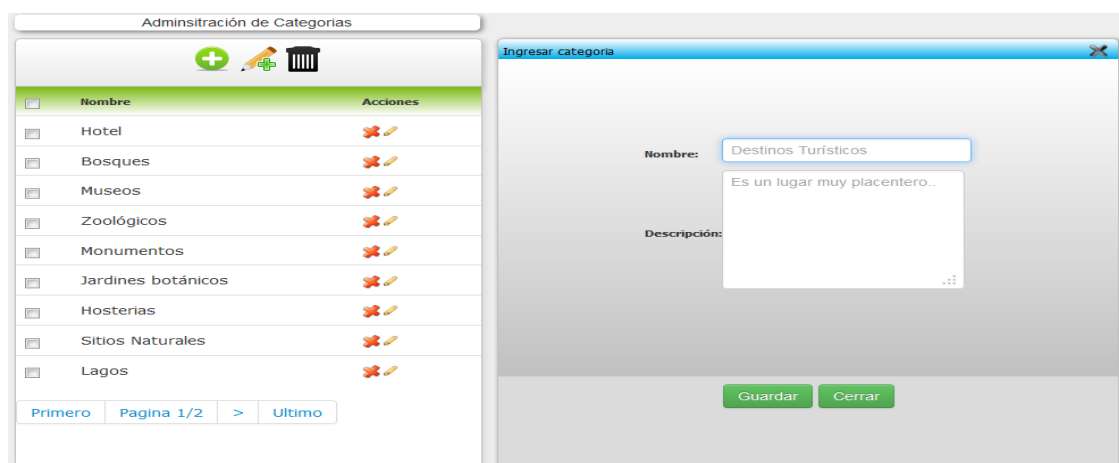


Ilustración 40: Formulario para ingresar una entidad categoría

Fuente: Propia

Editar categoría: El usuario deberá presionar sobre la imagen editar categoría, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32], descripción [5-256]

Eliminar categoría: Presionar sobre la imagen eliminar categoría.

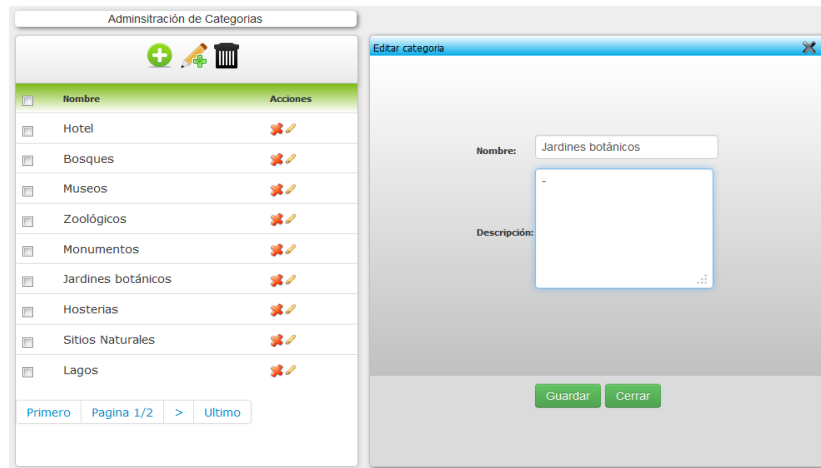


Ilustración 41: Formulario para editar una entidad categoría

Fuente: Propia

Historia 9: Gestión de sitios (12 días)

Ingresar sitio: El usuario deberá presionar sobre el botón agregar sitio, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres , descripción [5-256] caracteres, página web [5-32] caracteres, mail [5-32] caracteres, facebook [5-32] caracteres, twitter[5-32] caracteres dirección [5-64] caracteres, teléfono [5-64] caracteres.

The screenshot displays the 'Administración de Sitios' interface. At the top, there are location filters for 'Imbabura' and 'Ibarra'. Below this is a table listing various sites with their names, categories, and action icons. A modal window titled 'Ingresar Sitio' is open, showing a form to add a new site. The form includes fields for 'Nombre', 'Descripción', 'Web', 'Mail', 'Facebook', 'Twitter', 'Dirección', and 'Teléfono'. The 'Nombre' field contains 'Ajavi' and the 'Descripción' field contains 'Es un sitio adecuado..'. The 'Web' field contains 'www.hotelajavi.com', 'Mail' contains 'reservas@hotelajavi.com', 'Facebook' contains 'www.facebook.com/hotel.ajavi', 'Twitter' contains '@hotelajavi', 'Dirección' contains 'Avenida Mariano Acosta 16-38', and 'Teléfono' contains '(593) 06 2955-555, 2955-221'. There are 'Guardar' and 'Cerrar' buttons at the bottom of the form. Below the table and form is a map of Ecuador showing various provinces and cities.

Nombre	Categoría	Acciones
7-45 CAFÉ BAR	RESTAURANTES	✖ 📍
ABADÍA	BARES	✖ 📍
AGUSTÍN DELGAD.	HOSTERÍAS	✖ 📍
AJAVÍ	HOTELES	✖ 📍
ALAS DORADAS # .	RESTAURANTES	✖ 📍
ALAS DORADAS # .	RESTAURANTES	✖ 📍
ALLEQUATORIAL T.	AGENCIAS DE VIA.	✖ 📍
ALPARGATE EL # .	RESTAURANTES CO.	✖ 📍
AMAZONAS 2	HOSTALES RESIDE.	✖ 📍

Primero [Pagina 1/36](#) > Ultimo

Ingresar Sitio

Categoría: BARES

Nombre: Ajavi

Descripción: Es un sitio adecuado..

Web: www.hotelajavi.com

Mail: reservas@hotelajavi.com

Facebook: www.facebook.com/hotel.ajavi

Twitter: @hotelajavi

Dirección: Avenida Mariano Acosta 16-38

Teléfono: (593) 06 2955-555, 2955-221

Guardar Cerrar

Ilustración 42: Formulario para ingresar una entidad sitio

Fuente: Propia

Editar sitio: El usuario deberá presionar sobre la imagen editar sitio, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres , descripción [5-256] caracteres, página web [5-32] caracteres, mail [5-32] caracteres, facebook [5-32] caracteres, twitter[5-32] caracteres dirección [5-64] caracteres, teléfono [5-64] caracteres.

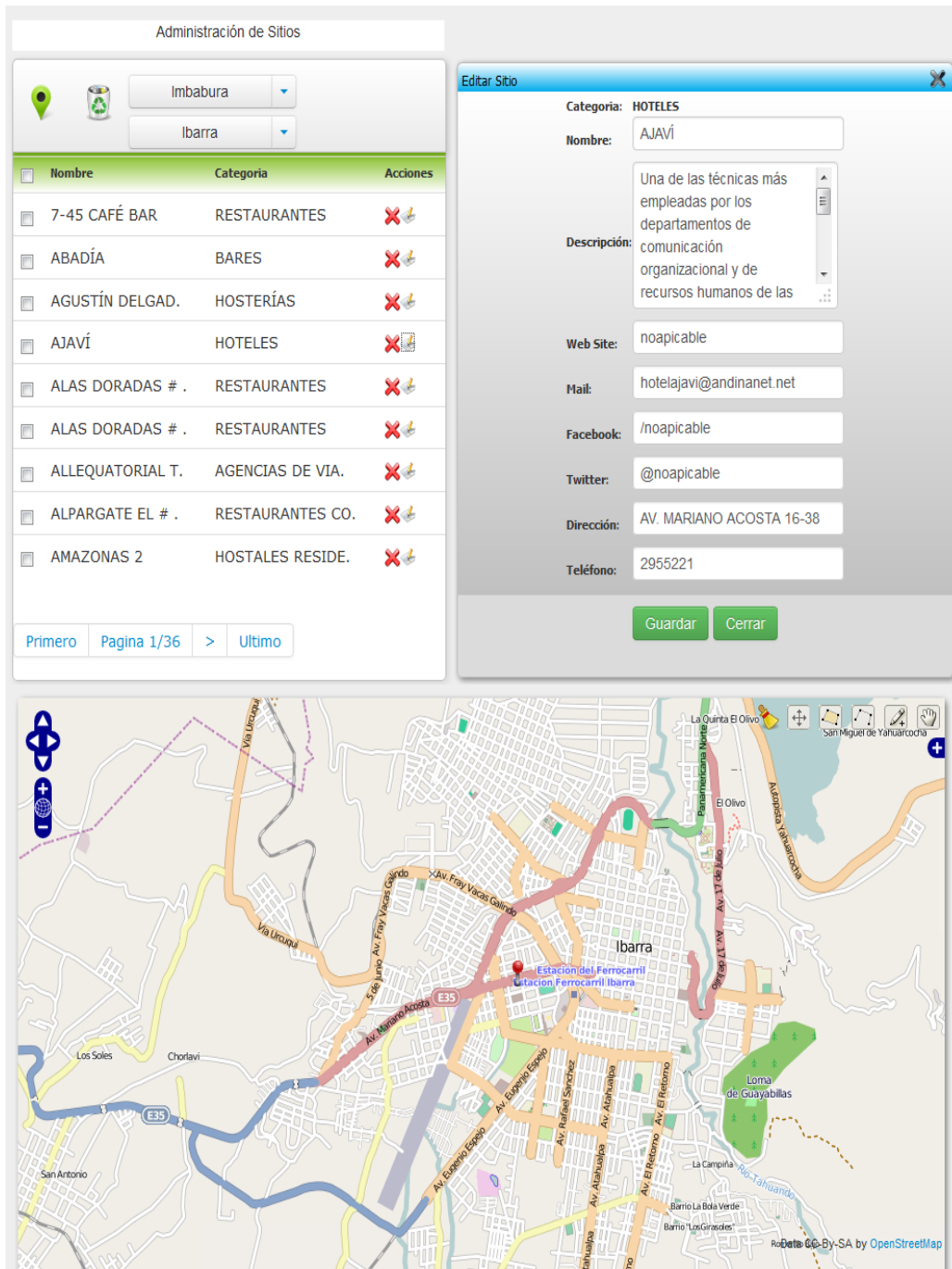


Ilustración 43: Formulario para editar una entidad sitio

Fuente: Propia

Historia 10: Gestión de historias (10 días)

Ingresar historia: El usuario deberá presionar sobre el botón agregar historia, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-1024] caracteres.

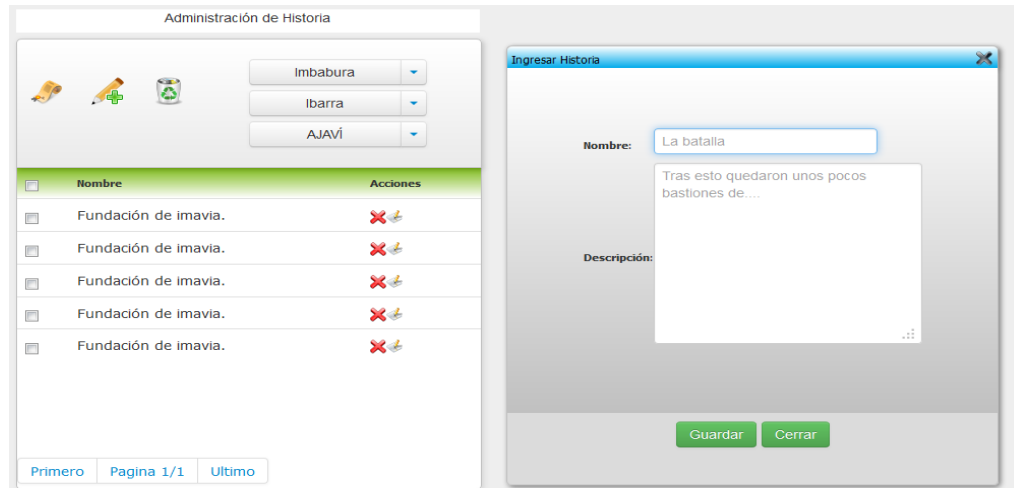


Ilustración 44: Formulario para ingresar una entidad historia

Fuente: Propia

Editar historia: El usuario deberá presionar sobre la imagen editar historia, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-1024] caracteres.

Eliminar historia: Presionar sobre la imagen eliminar historia.

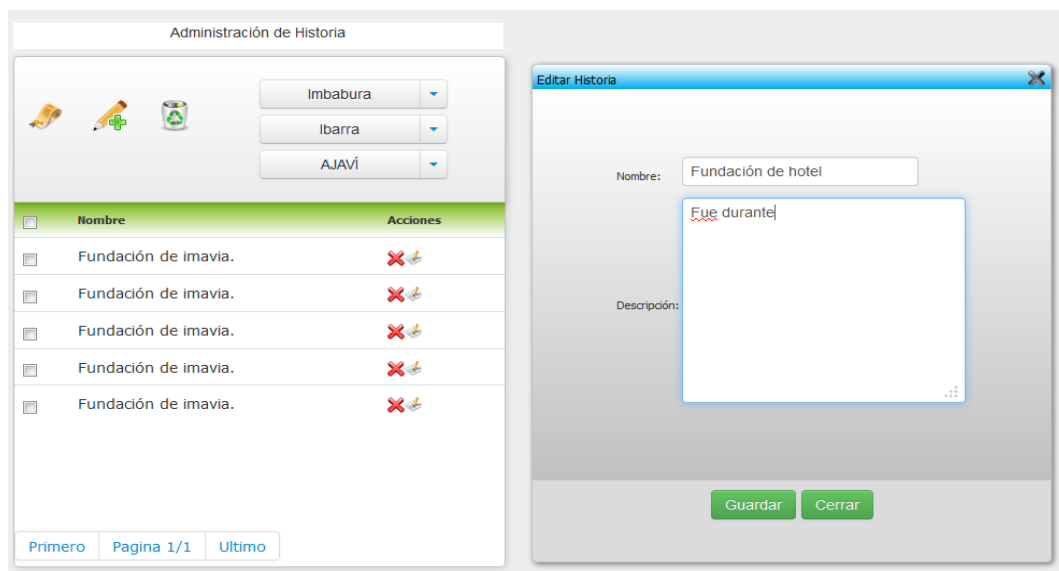


Ilustración 45: Formulario para editar una entidad historia

Fuente: Propia

Historia 11: Gestión de URL'S hacia YouTube (7 días)

Ingresar video: El usuario deberá presionar sobre el botón agregar video, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, URL [5-64] caracteres.

The screenshot shows the 'Administración de Video' interface. On the left, there is a table with columns 'Nombre' and 'Acciones'. The table lists seven video entities, each with a checkbox and a red 'X' icon. Below the table are navigation buttons: 'Primero', 'Pagina 1/3', '>', and 'Ultimo'. On the right, a modal window titled 'Editar video' is open. It contains three input fields: 'Nombre' (filled with 'Nuestros Servicios 1a'), 'Descripción' (filled with 'Nosotros, por otro lado, hemos estado pensando, buscando y seleccionando servicios que se ajusten a tus necesidades. Escapadas en Europa, combinados con Inter-rail,'), and 'Video' (filled with 'http://www.youtube.com/watch?v:'). At the bottom of the modal are two buttons: 'Guardar' and 'Cerrar'.

Ilustración 46: Formulario para ingresar una entidad video

Fuente: Propia

Editar video: El usuario deberá presionar sobre la imagen editar video, la cual habilitará un formulario para editar los campos respectivos.

Nombre [5-32] caracteres, descripción [5-256] caracteres, URL [5-64] caracteres.

Eliminar video: Presionar sobre la imagen eliminar video.

The screenshot shows the 'Administración de Video' interface. On the left, there is a table with columns 'Nombre' and 'Acciones'. The table lists seven video entities, each with a checkbox and a red 'X' icon. Below the table are navigation buttons: 'Primero', 'Pagina 1/3', '>', and 'Ultimo'. On the right, a modal window titled 'Ingresar video' is open. It contains three input fields: 'Nombre' (filled with 'Mirador Cuitocha'), 'Descripción' (filled with 'Puede realizar excursión al mirador Cuitocha'), and 'Url' (filled with 'www.youtube.com/watch?v=6zH7'). At the bottom of the modal are two buttons: 'Guardar' and 'Cerrar'.

Ilustración 47: Formulario para editar una entidad video

Fuente: Propia

Historia 12: Gestión de imágenes (6 días)

Ingresar imágenes: El usuario deberá presionar sobre el botón agregar imagen, el cual habilitará un formulario para ingresar los campos respectivos, y una imagen.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso menor a 200 KB, preferencias 500X500 PX.

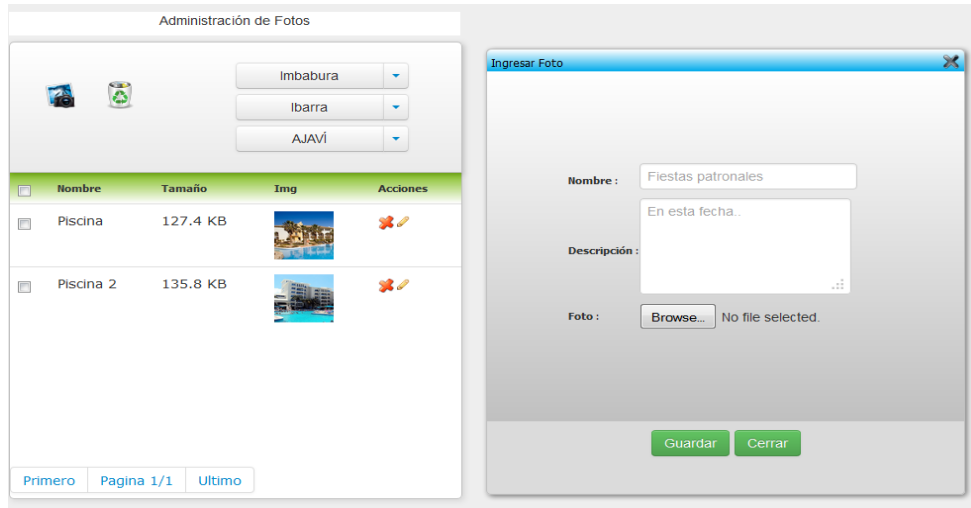


Ilustración 48: Formulario para ingresar una entidad foto

Fuente: Propia

Editar imágenes: El usuario deberá presionar sobre el botón editar imagen, el cual habilitará un formulario para editar los campos respectivos, y una imagen.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso menor a 200 KB, preferencias 500X500 PX.

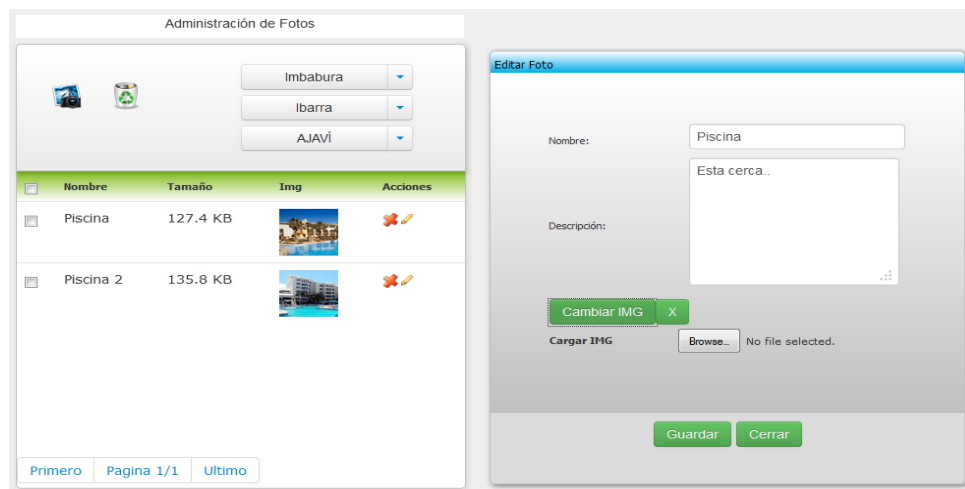


Ilustración 49: Formulario para editar una entidad foto

Fuente: Propia

Historia 13: Gestión de festivos (6 días)

Ingresar festivos: El usuario deberá presionar sobre el botón agregar festivo, el cual habilitará un formulario para ingresar los campos respectivos, y fechas.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX, la fecha inicial debe ser menor que la fecha mayor.

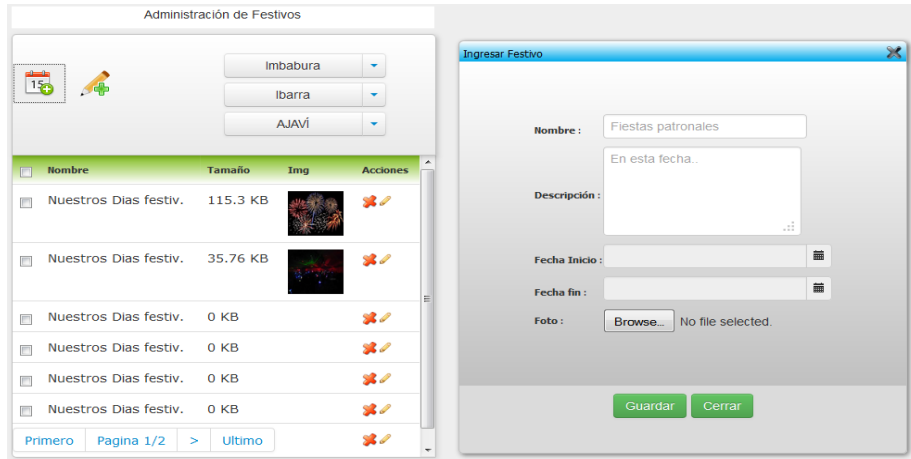


Ilustración 50: Formulario para ingresar una entidad festivo

Fuente: Propia

Editar festivo: El usuario deberá presionar sobre la imagen editar festivo, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX, la fecha inicial debe ser menor que la fecha mayor.

Eliminar festivo: Presionar sobre la imagen eliminar festivo.

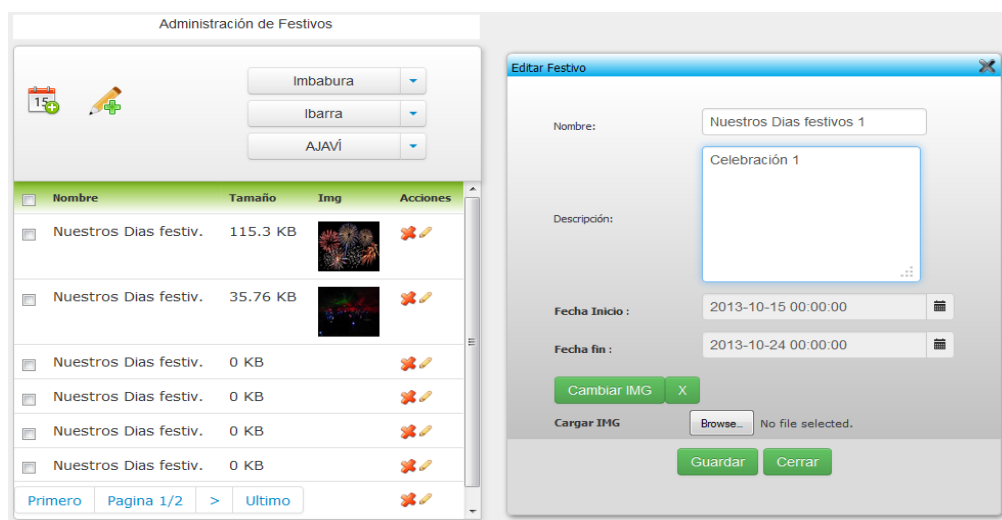


Ilustración 51: Formulario para editar una entidad festivo

Fuente: Propia

Historia 14: Gestión de gastronomías (6 días)

Ingresar gastronomía: El usuario deberá presionar sobre el botón agregar gastronomía, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX.

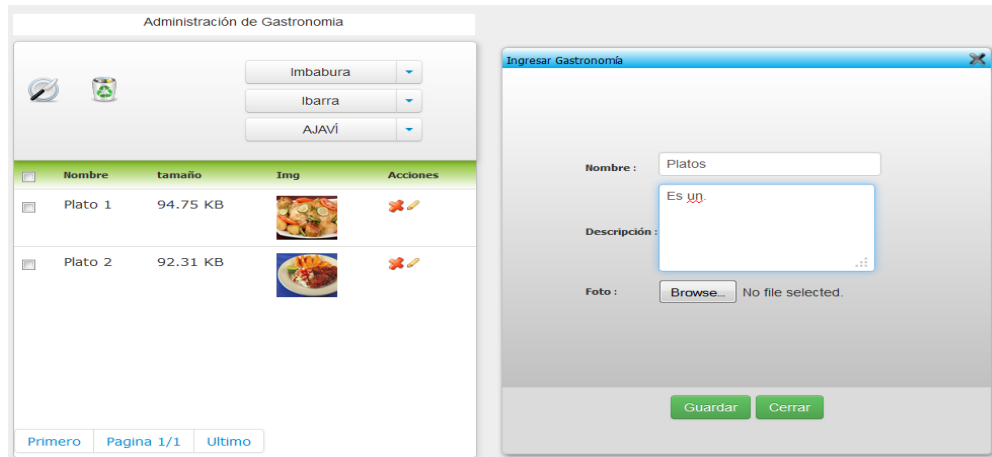


Ilustración 52: Formulario para ingresar una entidad gastronomía

Fuente: Propia

Editar gastronomía: El usuario deberá presionar sobre el botón editar gastronomía, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-256] caracteres, peso de imagen menor a 200 KB, preferencia 500X500 PX.

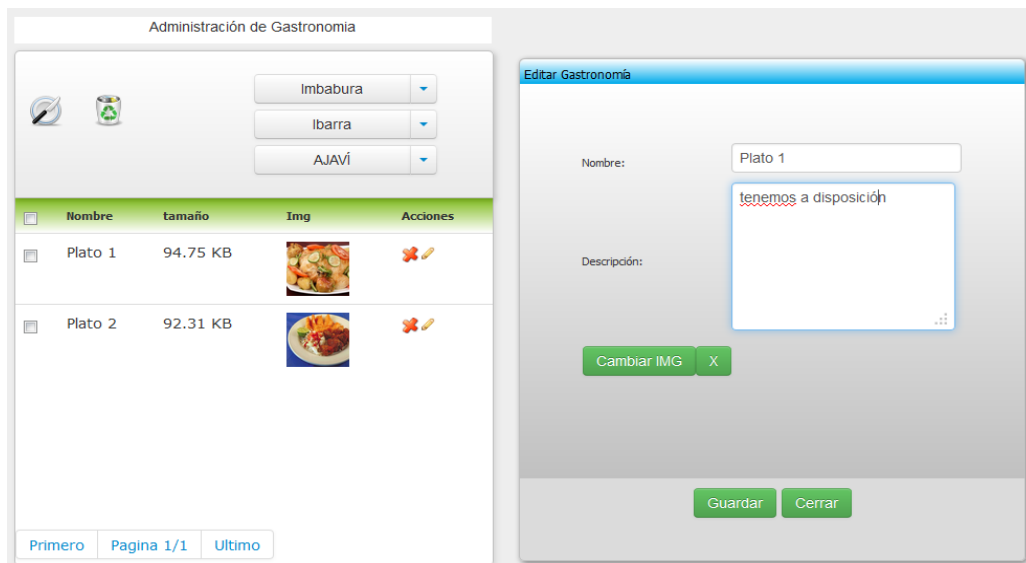


Ilustración 53: Formulario para editar una entidad gastronomía

Fuente: Propia

Historia 15: Gestión de rutas (9 días)

Ingresar ruta: El usuario deberá presionar sobre el botón agregar ruta, el cual habilitará un formulario para ingresar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-64] caracteres, graficar una ruta correspondiente.

Administración de Rutas

Imbabura
Ibarra
AJAVÍ

Nombre	Acciones
Aduana -Oficinasx3	✖ 🗑️
Entrada Ibarra-Oficinasx2	✖ 🗑️
Terminar-Oficinasx1	✖ 🗑️

Primero Pagina 1/1 Ultimo

Ingresar Ruta

Nombre : Termina- Hotel

Descripción : Es una ruta ..

Guardar Cerrar

Map data ©2013 Google

Ilustración 54: Formulario para ingresar una entidad ruta

Fuente: Propia

Editar ruta: El usuario deberá presionar sobre la imagen editar ruta, la cual habilitará un formulario para editar los campos respectivos.

Número de caracteres por campo permitidos: Nombre [5-32] caracteres, descripción [5-64] caracteres, graficar una ruta correspondiente.

Eliminar ruta: Presionar sobre la imagen eliminar ruta.

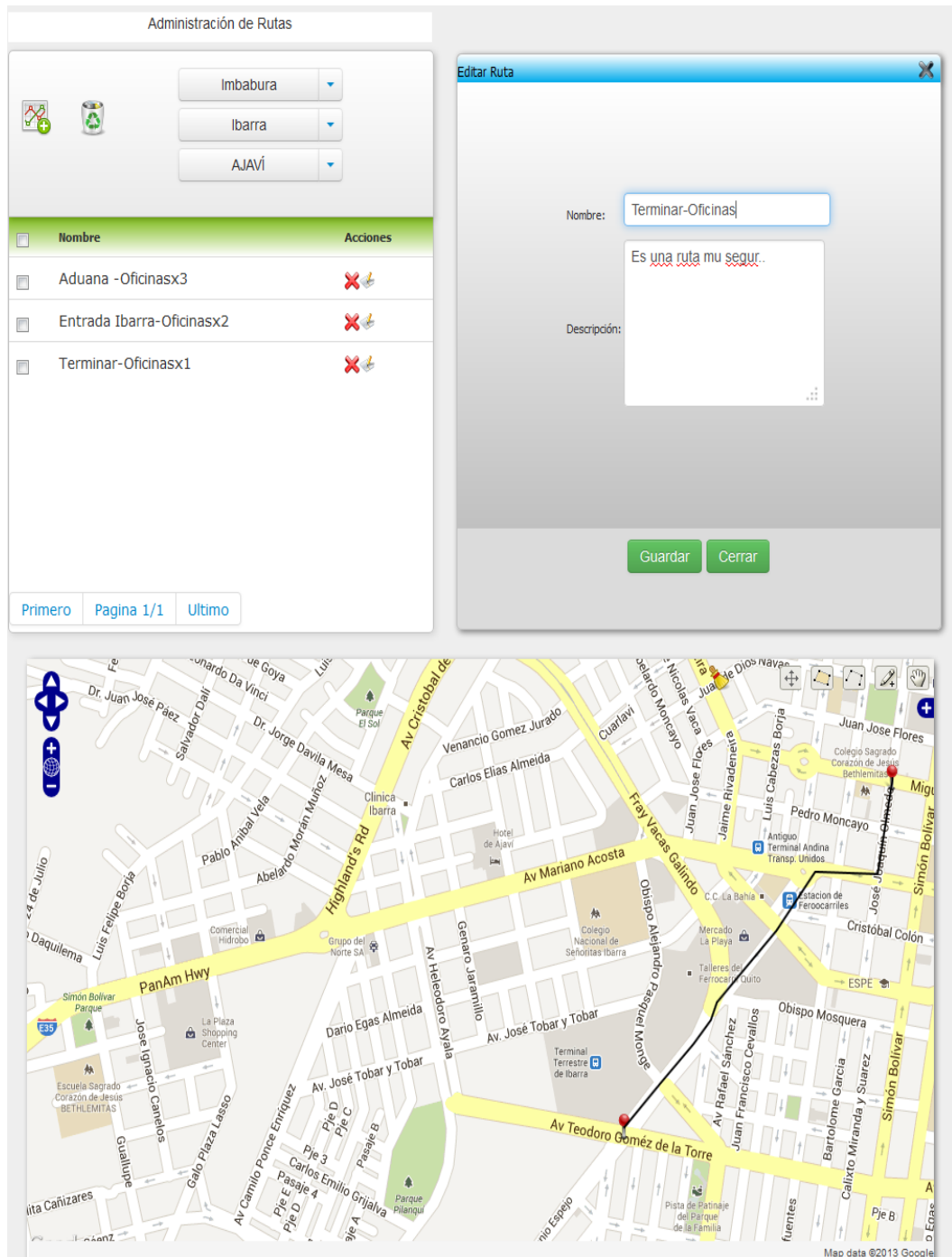


Ilustración 55: Formulario para editar una entidad ruta

Fuente: Propia

3.4.6 Pruebas de aceptación

Historia 6: Gestión de provincias

Acceso hacia las interfaces para administrar provincias, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad provincia

- Formulario para ingresar una provincia, sí ingresa los respectivos campos.
- Formulario para editar una provincia, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de provincias.

Historia 7: Gestión de ciudades

Acceso hacia las interfaces para administrar ciudades, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar, Eliminar) de la entidad ciudad.

- Formulario para ingresar una ciudad, sí ingresa los respectivos campos.
- Formulario para editar una ciudad, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de ciudades.

Historia 8: Gestión de categorías

Acceso hacia las interfaces para administrar categorías, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad categoría.

- Formulario para ingresar una categoría, sí ingresa los respectivos campos.
- Formulario para editar una categoría, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de categorías.

Historia 9: Gestión de sitios

Acceso hacia las interfaces para administrar sitios, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad sitio.

- Formulario para ingresar un sitio, sí ingresa los respectivos campos.
- Formulario para editar un sitio, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de sitios.

Historia 10: Gestión de historias

Acceso hacia las interfaces para administrar historias, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad historia.

- Formulario para ingresar una historia, sí ingresa los respectivos campos.
- Formulario para editar una historia, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de historias.

Historia 10: Gestión de URL'S hacia YouTube

Acceso hacia las interfaces para administrar enlaces hacia videos *YouTube*, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad video.

- Formulario para ingresar un video, sí ingresa los respectivos campos.
- Formulario para editar un video, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de videos.

Historia 12: Gestión de imágenes

Acceso hacia las interfaces para administrar imágenes, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad imagen.

- Formulario para ingresar una imagen, sí ingresa los respectivos campos.
- Formulario para editar una imagen, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de imágenes.

Historia 13: Gestión de festivos

Acceso hacia las interfaces para administrar festivos, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad festivo.

- Formulario para ingresar un festivo, sí ingresa los respectivos campos.
- Formulario para editar un festivo, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de festivos.

Historia 14: Gestión de gastronomías

Acceso hacia las interfaces para administrar gastronomías, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad gastronomías.

- Formulario para ingresar una gastronomía, sí ingresa los respectivos campos.
- Formulario para editar una gastronomía, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de gastronomías.

Historia 15: Gestión de rutas

Acceso hacia las interfaces para administrar rutas, el usuario sí realiza el *CRUD* (Crear, Leer, Actualizar y Eliminar) de la entidad ruta.

- Formulario para ingresar una ruta, sí ingresa los respectivos campos.
- Formulario para editar una ruta, sí edita los respectivos campos.
- El formulario sí cuenta con un listado descriptivo de rutas.

3.4.7 Incidencias

- Se rediseño la bases de datos frecuentemente,
- El visor de mapas fue cambiando de aspecto, pues de agregaron capas de *Google* y *OSM*.
- La versión de *POSTGIS* fue incompatible, por lo cual se instaló una versión compatible.

3.5 Iteración III

3.5.1 Cronograma de actividades

ITE.	Historias de usuario	Fecha	Duración en Semanas	Duración en horas
3	Historia 16: Reporte plano de Sitios.	03/4/13 07/4/13	1	45
3	Historia 17: Reporte geo-referenciado	09/4/13 18/4/13	2	71
3	Historia 18: Enlaces hacia historias	20/4/13 24/4/13	1	40
	Historia 19: Enlaces hacia YouTube	25/4/13 29/4/13	1	40
3	Historia 20: Enlaces hacia imágenes	03/5/13 09/5/13	1,2	56
3	Historia 21: Encales hacia festivos	12/5/13 14/5/13	0,3	30
3	Historia 22: Encales hacia gastronomías	15/5/13 19/5/13	1	40
3	Historia 23: Enlaces hacia rutas	20/5/13 26/5/13	1,2	56
3	Historia 24: Reportes plano de sitios (App móvil)	05/6/13 14/6/13	2	71
3	Historia 25: Reporte de historia videos (App móvil)	15/6/13 19/6/13	1,2	50
3	Historia 26: Reporte de festivos, fotos y gastronomías. (App móvil)	21/6/13 28/6/13	1,2	45
				Total: 544 h

Tabla 157: Cronograma de actividades Iteración III

Fuente: Propia

3.5.2 Tareas por historia Iteración III

Reporte plano de Sitios

Historia 16: Reporte plano de Sitios.		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	4 horas
2	Diseño de interfaz gráfica principal (<i>index.php</i>)	30 horas
3	Creación de archivos PHP	5 horas
4	Rediseño y maquetado	6 horas
		45 oras

Tabla 158: Tareas de reporte plano de sitios

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 16
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administrador de base de datos	Tiempo estimado: 4 horas
Fecha inicio: 03/4/13	Fecha fin: 03/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe, la creación de un script <i>SQL</i>, para realizar la búsqueda personalizada de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i>. Ingresamos como parámetros las claves primarias de: Provincia, ciudad y categoría; Obtenernos como respuesta un reporte de todos los sitios encontrados.</p> <p><i>busqSitio.sql:</i> Ejecuta el operador <i>WHERE</i>, además retorna los valores encontrados.</p>	

Tabla 159: Reporte plano de sitios – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica principal (index.php)

Número tarea: 2	Número historia: 16
Nombre tarea: Diseño de interfaz gráfica principal (<i>index.php</i>)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 30 horas
Fecha inicio: 03/4/13	Fecha fin: 05/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el diseño de la página web principal denominada <i>index.php</i>, la cual no estará sujeta a ningún tipo de restricción de usuarios; En esta pantalla se efectuarán las búsquedas principales de sitios.</p> <ul style="list-style-type: none"> - Utilizamos la librería <i>TWITTER BOOTSTRAPMASTER</i> para mejorar el estilo de nuestra pantalla principal. - Implementamos componentes responsivos, el cual nos permitirá que las etiquetas <i>HTML</i>, se autoajusten al tamaño de navegador web. - Construcción de menús: Provincia y ciudades. - Diseño de un campo de texto para que el usuario digite la categoría a buscar. <p>buscarSitio(): función <i>JAVASCRIPT</i> que recibe como parámetro las claves primarias de: Provincia, ciudad y categoría, y como resultado imprime los sitios turísticos y destinos que cumplan con los valores introducidos por el usuario.</p>	

Tabla 160: Reporte plano de sitios – Tarea Nro. 2

Fuente: Propia

3. Creación de archivos PHP

Número tarea: 3	Número historia: 16
Nombre tarea: Creación de archivos PHP.	
Tipo de tarea : Programación	Tiempo estimado: 5 horas
Fecha inicio: 06/4/13	Fecha fin: 06/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la búsqueda que se realice en lado del servidor; referente a la información de sitios turísticos disponible.</p> <p>categoriaNombre.php: Recibe como parámetro el nombre de una categoría y nos devuelve la clave respectiva.</p> <p>sitioConsBusqueda.php: Archivo en el cual instanciamos el objeto sitio, el cual nos permite recibir como parámetros las variables correspondientes a: Provincia, ciudad, y categoría.</p>	

Tabla 161: Reporte plano de sitios – Tarea Nro. 3

Fuente: Propia

4. Rediseño y maquetado

Número tarea: 4	Número historia: 16
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 07/4/13	Fecha fin: 07/4/13
Programador responsable: Jairo Morales	
<p>Descripción: Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>index.php</i>, que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación. Una vez que él usuarios se registra, en la mayor brevedad posible recibirá un mail, con la contraseña de acceso respectiva hacia el sistema.</p> <p>Funciones JAVASCRIPT: validarCamposUsuario(): Permite validar las variables ingresadas por el usuario. insertUsuario(): Recibe como parámetro las variables de un usuario respectivo y las envía, hacia el servidor utilizando el objeto <i>AJAX</i>. Construcción de archivos <i>PHP</i>. usuarioFormInsert.php: Formulario encargado de enviar las variables hacia el servidor, para ingresar los campos respectivos. usuarioFuncion.php: Incorporamos funciones para ingresar un usuario.</p>	

Tabla 162: Reporte plano de sitios – Tarea Nro. 4

Fuente: Propia

Reporte geo-referenciado

Historia 17: Reporte geo-referenciado		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	5 horas
2	Creación de archivos PHP	20 horas
3	Rediseño de interfaz gráfica principal (<i>index.php</i>)	30 horas
4	Graficar marcadores sobre mapa	16 horas
		71 Horas

Tabla 163: Tareas de reporte geo-referenciado

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 17
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administrador de base de datos	Tiempo estimado: 5 horas
Fecha inicio: 09/4/13	Fecha fin: 09/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe, la creación de un script <i>SQL</i>, para realizar la búsqueda personalizada de un sitio geo-referenciado a nivel de bases de datos, utilizando el operador <i>WHERE</i>.</p> <p>Insertamos como parámetro las claves primarias de provincia, ciudad y categoría. Obtenemos como respuesta un reporte de todos los sitios encontrados, y la información geográfica.</p> <p><i>busqSitio.sql:</i> Ejecuta el operador <i>WHERE</i>, además retorna los valores encontrados.</p>	

Tabla 164: Reporte geo-referenciado – Tarea Nro. 1

Fuente: Propia

2. Creación de archivos PHP

Número tarea: 2	Número historia: 17
Nombre tarea: Creación de archivos PHP.	
Tipo de tarea : Programación	Tiempo estimado: 20 horas
Fecha inicio: 10/4/13	Fecha fin: 12/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso de creación de archivos <i>PHP</i>, que nos permitirá imprimir la búsqueda geo-referenciada que se realice en lado del servidor.</p> <p><i>sitioConsBusquedaJson.php:</i> Instanciamos el objeto sitio, el cual nos permite recibir como parámetros las claves primarias de: Provincia, ciudad y categoría; Como respuesta obtenemos una matriz con la información de un sitios encontrado, en formato <i>JSON_ENCODE</i>.</p>	

Tabla 165: Reporte geo-referenciado – Tarea Nro. 2

Fuente: Propia

3. Rediseño de interfaz gráfica principal (index.php)

Número tarea: 3	Número historia: 17
Nombre tarea: Rediseño de interfaz gráfica principal (<i>index.php</i>)	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 30 horas
Fecha inicio: 13/4/13	Fecha fin: 16/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el rediseño de la página web principal, la cual no estará sujeta a ninguna restricción de usuarios, para su acceso.</p> <p>Funciones <i>JAVASCRIPT</i>, que actúan a los eventos de búsqueda plana; fue necesario instanciar una comunicación <i>AJAX</i> asíncrona con el servidor.</p> <p>sitioConsMapa(): Recibe como parámetro el nombre de una categoría y retorna su respectiva clave primaria</p> <p>buscarSitioMapa(): Obtenemos la información geográfica y plana asociada de un sitio, para posteriormente graficar los marcadores en el mapa.</p>	

Tabla 166: Reporte geo-referenciado – Tarea Nro. 3

Fuente: Propia

4. Graficar marcador sobre mapa

Número tarea: 4	Número historia: 17
Nombre tarea: Graficar marcadores sobre el visor mapas	
Tipo de tarea : programación GIS	Tiempo estimado: 16 horas
Fecha inicio: 17/4/13	Fecha fin: 18/4/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el diseño de un visor de mapas pre-configurado, el cual recibe el archivo en formato <i>JSON</i> enviado desde el servidor.</p> <p>Instanciamos un objeto mapa, y añadimos los marcadores correspondientes a un sitio, mediante la utilización de la librería <i>Open Layers</i>.</p> <p>Campos a mostrar: Nombre, facebook, twitter, dirección, teléfono, latitud y longitud.</p> <p>graficarbuscarSitioMapa(): Permite graficar los marcadores de la búsqueda respectiva.</p>	

Tabla 167: Reporte geo-referenciado – Tarea Nro. 4

Fuente: Propia

✚ Enlaces hacia historias

Historia 18: Enlaces hacia historias		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	6 horas
2	Diseño de interfaz gráfica <i>historias.php</i>	8 horas
3	Construcción de reporte historias	20 horas
4	Rediseño y maquetado	6 horas
		40 Horas

Tabla 168: Tareas de enlaces hacia historias

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 18
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 20/4/13	Fecha fin: 20/4/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe, la creación de un script <i>SQL</i> , para realizar la búsqueda personalizada de historias de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i> . Se recibe como parámetro las claves primarias de: Provincia, ciudad, categoría, sitio y nos devuelve un objeto con todas las historias encontradas.	

Tabla 169: Reporte geo-referenciado – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica historias.php

Número tarea: 2	Número historia: 18
Nombre tarea: Diseño de interfaz gráfica <i>historias.php</i>	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 8 horas
Fecha inicio: 21/4/13	Fecha fin: 21/4/13
Programador responsable: Jairo Morales	
Descripción: La presente tareas describe el proceso de diseño, para la pantalla de reporte, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i> , además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> .	

Tabla 170: Enlaces hacia historias – Tarea Nro. 2

Fuente: Propia

3. Construcción de reporte historias

Número tarea: 3	Número historia: 18
Nombre tarea: Construcción de reporte historias.	
Tipo de tarea : Programación	Tiempo estimado: 20 horas
Fecha inicio: 22/4/13	Fecha fin: 23/4/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso, para imprimir los reportes de los enlaces hacia las historias encontradas. Campos a mostrar: Nombre y descripción.	

Tabla 171: Enlaces hacia historias – Tarea Nro. 3

Fuente: Propia

4. Rediseño y maquetado

Número tarea: 4	Número historia: 18
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 24/4/13	Fecha fin: 24/4/13
Programador responsable: Jairo Morales	
Descripción: Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>historias.php</i> , que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación.	

Tabla 172: Enlaces hacia historias – Tarea Nro. 4

Fuente: Propia

- **Enlaces hacia *YouTube***

Historia 19: Enlaces hacia <i>YouTube</i>		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	6 horas
2	Diseño de interfaz gráfica <i>videos.php</i>	24 horas
3	Construcción de reporte videos	12 horas
4	Rediseño y maquetado	6 horas
		Total: 40 horas

Tabla 173: Tareas de enlace hacia YouTube
Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 19
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 25/4/13	Fecha fin: 25/4/13
Programador responsable: Jairo Morales	
Descripción: Creación de un script <i>SQL</i> , para realizar la búsqueda personalizada de enlaces <i>YouTube</i> de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i> . Se escribió un script de prueba que nos permite efectuar una búsqueda sobre las entidades: provincia, ciudad, categoría y sitio. Se recibe como parámetro las claves primarias y nos devuelve un objeto con todos los enlaces hacia <i>YouTube</i> encontrados.	

Tabla 174: Enlaces hacia YouTube – Tarea Nro. 1
Fuente: Propia

2. Diseño de interfaz gráfica videos.php

Número tarea: 2	Número historia: 19
Nombre tarea: Diseño de interfaz gráfica <i>videos.php</i>	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 24 horas
Fecha inicio: 26/4/13	Fecha fin: 27/4/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla enlace hacia videos <i>YouTube</i> , mediante la construcción de hojas de estilo CSS y archivos <i>JAVASCRIPT</i> .	

Tabla 175: Enlaces hacia YouTube – Tarea Nro. 2

Fuente: Propia

3. Construcción de reporte videos

Número tarea: 3	Número historia: 19
Nombre tarea: Construcción de reporte videos.	
Tipo de tarea : Programación	Tiempo estimado: 12 horas
Fecha inicio: 28/4/13	Fecha fin: 28/4/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso, para imprimir los reportes de video hacia <i>YouTube</i> encontrados. Campos a mostrar: Nombre, descripción y <i>URL</i> . Instanciamos el objeto video que nos recibe como parámetros, las claves primarias de una provincia, ciudad, categoría y sitio.	

Tabla 176: Enlaces hacia YouTube – Tarea Nro. 3

Fuente: Propia

4. Rediseño y maquetado

Número tarea: 4	Número historia: 19
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 6 horas
Fecha inicio: 29/4/13	Fecha fin: 29/4/13
Programador responsable: Jairo Morales	
Descripción: <p>Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>videos.php</i>, que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación.</p>	

Tabla 177: Enlaces hacia YouTube – Tarea Nro. 4

Fuente: Propia

- **Enlaces hacia imágenes**

Historia 20: Enlaces hacia imágenes		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales.	6 horas
2	Diseño de interfaz gráfica <i>imagenes.php</i>	30 horas
3	Construcción de reporte imágenes.	16 horas
4	Rediseño y maquetado	4 horas
		Total: 56 horas

Tabla 178: Tareas de enlaces hacia imágenes

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 20
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 03/5/13	Fecha fin: 03/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe, la creación de un script <i>SQL</i> , para realizar la búsqueda personalizada de imágenes de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i> . Se escribió un script de prueba que nos permite efectuar una búsqueda sobre las entidades: provincia, ciudad, categoría y sitio. Se recibe como parámetro las claves primarias y nos devuelve un objeto con todas las imágenes encontradas, así también su nombre y descripción.	

Tabla 179: Enlaces hacia imágenes – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica imagenes.php

Número tarea: 2	Número historia: 20
Nombre tarea: Diseño de interfaz gráfica <i>imagenes.php</i>	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 30 horas
Fecha inicio: 04/5/13	Fecha fin: 06/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño de un carrusel de imágenes, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i> , además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para mejorar el diseño de nuestros componentes <i>HTML</i> .	

Tabla 180: Enlaces hacia imágenes – Tarea Nro. 2

Fuente: Propia

3. Construcción de reporte imágenes

Número tarea: 3	Número historia: 20
Nombre tarea: Construcción de reporte imágenes.	
Tipo de tarea : Programación	Tiempo estimado: 16 horas
Fecha inicio: 07/5/13	Fecha fin: 08/5/13
Programador responsable: Jairo Morales	
Descripción: Instanciamos el objeto imagen que nos recibe como parámetros, las claves primarias de una provincia, ciudad, categoría y sitio. Combinamos código <i>PHP</i> y <i>HTML</i> para dar forma a nuestro carrusel de imágenes, el cual procederá a mostrar las imágenes, con un nombre y descripción.	

Tabla 181: Enlaces hacia imágenes – Tarea Nro. 3

Fuente: Propia

4. Rediseño y maquetado

Número tarea: 4	Número historia: 20
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 4 horas
Fecha inicio: 09/4/13	Fecha fin: 09/4/13
Programador responsable: Jairo Morales	
Descripción: Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>imagenes.php</i> , que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación. Campos a mostrar: Nombre y descripción. Reutilizamos las funciones <i>PHP</i> y <i>JAVASCRIPT</i> implementadas en la HISTORIA 16: Reporte plano.	

Tabla 182: Enlaces hacia imágenes – Tarea Nro. 4

Fuente: Propia

- **Enlaces hacia festivos**

Historia 21: Enlaces hacia festivos		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	6 horas
2	Diseño de interfaz gráfica <i>festivos.php</i>	12 horas
3	Construcción de reporte festivo	8 horas
4	Rediseño y maquetado	4 horas
		Total: 30 horas

Tabla 183: Tareas de enlaces hacia festivos

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 21
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 12/5/13	Fecha fin: 12/5/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe, la creación de un script <i>SQL</i>, para realizar la búsqueda personalizada de festivos de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i>.</p> <p>Se escribió un script de prueba que nos permite efectuar una búsqueda sobre las entidades: provincia, ciudad, categoría y sitio. Se recibe como parámetro las claves primarias y nos devuelve un objeto con todos los festivos encontrados.</p>	

Tabla 184: Enlaces hacia festivos – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica festivos.php

Número tarea: 2	Número historia: 21
Nombre tarea: Diseño de interfaz gráfica <i>festivos.php</i>	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 12 horas
Fecha inicio: 13/5/13	Fecha fin: 13/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla <i>festivos.php</i> , mediante la construcción de hojas de estilo CSS y archivos <i>JAVASCRIPT</i> , además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para mejorar el diseño de nuestros componentes <i>HTML</i> .	

Tabla 185: Enlaces hacia festivos – Tarea Nro. 2

Fuente: Propia

3. Construcción de reporte festivo.

Número tarea: 3	Número historia: 21
Nombre tarea: Construcción de reporte festivo.	
Tipo de tarea : Programación	Tiempo estimado: 8 horas
Fecha inicio: 14/5/13	Fecha fin: 14/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso, para imprimir los reportes de festivos encontrados. Campos a mostrar: Nombre, descripción, fecha inicio, fecha fin y foto. Instanciamos el objeto festivo que nos recibe como parámetros, las claves primarias de una provincia, ciudad, categoría y sitio. Combinamos código <i>PHP</i> y <i>HTML</i> para dar forma a nuestro carrusel de imágenes.	

Tabla 186: Enlaces hacia festivos – Tarea Nro. 3

Fuente: Propia

4. Rediseño y maquetado

Número tarea: 4	Número historia: 21
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 4 horas
Fecha inicio: 14/5/13	Fecha fin: 14/5/13
Programador responsable: Jairo Morales	
Descripción: Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>festivos.php</i> , que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación.	

Tabla 187: Enlaces hacia festivos – Tarea Nro. 4

Fuente: Propia

- **Enlaces hacia gastronomías**

Historia 22: Enlaces hacia gastronomías		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	6 horas
2	Diseño de interfaz gráfica <i>gastronomías.php</i>	20 horas
3	Construcción de reporte gastronomías.	10 horas
4	Rediseño y maquetado	4 horas
		Total: 40 horas

Tabla 188: Tareas de enlaces hacia gastronomía

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 22
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 15/5/13	Fecha fin: 15/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe, la creación de un script <i>SQL</i> , para realizar la búsqueda personalizada de gastronomías de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i> . Se recibe como parámetro las claves primarias y nos devuelve un objeto con todas las gastronomías encontradas, así también: nombre, descripción e imagen.	

Tabla 189: Enlaces hacia gastronomía – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica gastronomías.php

Número tarea: 2	Número historia: 22
Nombre tarea: Diseño de interfaz gráfica <i>gastronomías.php</i>	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 20 horas
Fecha inicio: 16/5/13	Fecha fin: 17/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la creación de un carrusel de imágenes, mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i> , además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para mejorar el diseño de nuestros componentes <i>HTML</i> .	

Tabla 190: Enlaces hacia gastronomía – Tarea Nro. 2

Fuente: Propia

3. Construcción de reporte gastronomías

Número tarea: 3	Número historia: 22
Nombre tarea: Construcción de reporte gastronomías.	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 18/5/13	Fecha fin: 18/5/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe el proceso, para imprimir los reportes de gastronomías encontradas</p> <p>Instanciamos el objeto gastronomía que nos recibe como parámetros, las claves primarias de una provincia, ciudad, categoría y sitio. Combinamos código <i>PHP</i> y <i>HTML</i> para dar forma a nuestro carrusel de gastronomías.</p>	

Tabla 191: Enlaces hacia gastronomía – Tarea Nro. 3

Fuente: Propia

4. Rediseño y maquetado

Número tarea: 4	Número historia: 22
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 4 horas
Fecha inicio: 19/5/13	Fecha fin: 19/5/13
Programador responsable: Jairo Morales	
<p>Descripción: Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>gastronomia.php</i>, que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación.</p>	

Tabla 192: Enlaces hacia gastronomía – Tarea Nro. 4

Fuente: Propia

- **Enlaces hacia rutas**

Historia 23: Enlaces hacia rutas		
Número	Nombre	Tiempo estimado
1	Creación de scripts utilizando operadores relacionales	6 horas
2	Diseño de interfaz gráfica <i>rutas.php</i>	25 horas
3	Construcción de reporte rutas	21 horas
4	Construcción de reporte rutas	4 horas
		Total: 56 horas

Tabla 193: Tarea de enlaces hacia rutas

Fuente: Propia

1. Creación de scripts utilizando operadores relacionales

Número tarea: 1	Número historia: 23
Nombre tarea: Creación de scripts utilizando operadores relacionales.	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 6 horas
Fecha inicio: 20/5/13	Fecha fin: 20/5/13
Programador responsable: Jairo Morales	
<p>Descripción: La presente tarea describe, la creación de un script <i>SQL</i>, para realizar la búsqueda personalizada de rutas de un sitio a nivel de bases de datos, utilizando el operador <i>WHERE</i>. Se escribió un script de prueba que nos permite efectuar una búsqueda sobre las entidades: provincia, ciudad, categoría y sitio. Se recibe como parámetro las claves primarias y nos devuelve un objeto con todas las rutas encontradas: Nombre, descripción y ruta.</p>	

Tabla 194: Enlaces hacia rutas – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica rutas.php

Número tarea: 2	Número historia: 23
Nombre tarea: Diseño de interfaz gráfica <i>rutas.php</i>	
Tipo de tarea : Diseño y Maquetado	Tiempo estimado: 25 horas
Fecha inicio: 21/5/13	Fecha fin: 23/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso de diseño, para la pantalla <i>rutas.php</i> , mediante la construcción de hojas de estilo <i>CSS</i> y archivos <i>JAVASCRIPT</i> , además utilizamos la librería <i>TWITTER BOOTSTRAP MASTER</i> , para mejorar el diseño de nuestros componentes <i>HTML</i> .	

Tabla 195: Enlaces hacia rutas – Tarea Nro. 2

Fuente: Propia

3. Construcción de reporte rutas

Número tarea: 3	Número historia: 23
Nombre tarea: Construcción de reporte rutas.	
Tipo de tarea : Programación	Tiempo estimado: 21 horas
Fecha inicio: 24/5/13	Fecha fin: 25/5/13
Programador responsable: Jairo Morales	
Descripción: La presente tarea describe el proceso, para imprimir los reportes de rutas encontradas Instanciamos el objeto ruta que nos recibe como parámetros, las claves primarias de una provincia, ciudad, categoría y sitio. Combinamos código PHP y HTML para dar forma a nuestros reportes.	

Tabla 196: Enlaces hacia rutas – Tarea Nro. 3

Fuente: Propia

4. Construcción de reporte rutas

Número tarea: 4	Número historia: 23
Nombre tarea: Rediseño y maquetado.	
Tipo de tarea : Programación	Tiempo estimado: 4 horas
Fecha inicio: 26/5/13	Fecha fin: 26/5/13
Programador responsable: Jairo Morales	
Descripción: Construimos e incorporamos un formulario de ingreso, sobre la pantalla principal <i>rutas.php</i> , que le permitirá a un usuario registrarse directamente hacia la base de datos, sin la necesidad de dependencia del administrador de la aplicación	

Tabla 197: Enlaces hacia rutas – Tarea Nro. 4

Fuente: Propia

- **Reporte plano de sitios (App móvil)**

Historia 24: Reporte plano de sitios (App móvil)		
Número	Nombre	Tiempo estimado
1	Instalación y configuración de herramientas <i>Android</i>	20 horas
2	Creación de servicios web con inyección.	20 horas
3	Diseño de interfaz gráfica de <i>Android</i>	15 horas
4	Consumir servicios web utilizando <i>Android</i> .	16 horas
		Total: 71 horas

Tabla 198: Tareas de reporte plano de sitios (App móvil)

Fuente: Propia

1.Instalación y configuración de herramientas Android

Número tarea: 1	Número historia: 24
Nombre tarea: Instalación y configuración de herramientas <i>Android</i> .	
Tipo de tarea : Administración de base de datos	Tiempo estimado: 20 horas
Fecha inicio: 05/6/13	Fecha fin: 06/6/13
Programador responsable: Jairo Morales	
<p>Descripción: Se configuro un ambiente para desarrollar aplicaciones móviles.</p> <p>Eclipse Indigo: Plataforma de desarrollo para <i>JAVA</i>. SDK manager: Administrador de la API de <i>Android</i>. JDK Java: Conjunto de herramientas y librerías que permiten desarrollar y compilar programas en lenguaje Java. API Level: <i>Android 4.2.2 API (17)</i> AVD Manager: Emulador virtual para realizar pruebas locales.</p>	

Tabla 199: Reporte plano de sitios (App móvil) – Tarea Nro. 1

Fuente: Propia

2. Creación de servicios web utilizando lenguaje *PHP*

Número tarea: 2	Número historia: 24
Nombre tarea: Creación de servicios web con inyección.	
Tipo de tarea : Programación	Tiempo estimado: 20 horas
Fecha inicio: 07/6/13	Fecha fin: 09/6/13
Programador responsable: Jairo Morales	
<p>Descripción: En la presente tarea se implementó servicios web, utilizando el lenguaje de programación <i>PHP</i> y notación <i>JSON</i>, para la transferencia de estructuras de datos del servidor <i>VPS</i> hacia el cliente.</p> <p>Se implementó servicios web <i>REST</i> con inyección para él envió de información especializada de sitios turísticos.</p> <p>El archivo <i>wbSitio.php</i> retorna una estructura de dato en formato <i>JSON</i> con los sitios encontrados.</p>	

Tabla 200: Reporte plano de sitios (App móvil) – Tarea Nro. 2

Fuente: Propia

3. Diseño de interfaz gráfica de *Android*

Número tarea: 3	Número historia: 24
Nombre tarea: Diseño de interfaz gráfica de <i>Android</i>	
Tipo de tarea : Programación	Tiempo estimado: 15 horas
Fecha inicio: 10/6/13	Fecha fin: 11/6/13
Programador responsable: Jairo Morales	
Descripción: En la presente tarea de diseño la interfaz gráfica de usuarios utilizando <i>Spinners</i> (<i>MENUS</i>), y archivos <i>XML</i> para la presentación. Se diseñó varias actividades que trabajan en conjunto. <i>SpinnerProvincias:</i> Listará las provincias ingresadas en la web de administración. <i>SpinnerCiudades:</i> Despliega las ciudades de una provincia. <i>SpinnerCategorías:</i> Despliega una lista de categorías.	

Tabla 201: Reporte plano de sitios (App móvil) – Tarea Nro. 3

Fuente: Propia

4. Consumir servicios web utilizando *Android*

Número tarea: 4	Número historia: 24
Nombre tarea: Consumir servicios web utilizando <i>Android</i> .	
Tipo de tarea : Programación	Tiempo estimado: 16 horas
Fecha inicio: 13/6/13	Fecha fin: 14/6/13
Programador responsable: Jairo Morales	
Descripción: Se utilizó el lenguaje de programación <i>JAVA</i> , para acceder a los servicios web generados en <i>PHP</i> . También fue necesaria la utilización del objeto <i>HTTPCLIENT</i> para establecer una conexión asíncrona entre el cliente y el servidor.	

Tabla 202: Reporte plano de sitios (App móvil) – Tarea Nro. 4

Fuente: Propia

- **Reporte de historias y videos (App móvil)**

Historia 25: Reporte de historias y videos (App móvil).		
Número	Nombre	Tiempo estimado
1	Creación de servicios web con inyección.	20 horas
2	Diseño de interfaz gráfica de <i>Android</i> .	10 horas
3	Consumir servicios web utilizando <i>Android</i> .	10 horas
		Total: 40 horas

Tabla 203: Tareas de reporte de historias y videos (App móvil)

Fuente: Propia

1. Creación de servicios web con inyección

Número tarea: 1	Número historia: 25
Nombre tarea: Creación de servicios web con inyección.	
Tipo de tarea : Programación	Tiempo estimado: 20 horas
Fecha inicio: 15/6/13	Fecha fin: 16/6/13
Programador responsable: Jairo Morales	
<p>Descripción: Se implementó dos servicios web utilizando <i>PHP</i> y la notación para objetos <i>JSON</i>. Servicios web video: Retorna una estructura de datos con todos los videos encontrados. Servicios web historia: Retorna una estructura de datos con todas las historias encontradas.</p>	

Tabla 204: Reporte de historias y videos (App móvil) – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica de Android

Número tarea: 2	Número historia: 25
Nombre tarea: Diseño de interfaz gráfica de <i>Android</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 17/6/13	Fecha fin: 17/6/13
Programador responsable: Jairo Morales	
Descripción: Para el diseño de la interfaz de usuario para <i>Android</i> , se implementó 2 botones: Historias y Videos, los cuales tienen sus propias actividades y clases <i>JAVA</i> .	

Tabla 205: Reporte de historias y videos (App móvil) – Tarea Nro. 2

Fuente: Propia

3. Consumir servicios web utilizando Android

Número tarea: 3	Número historia: 24
Nombre tarea: Consumir servicios web utilizando <i>Android</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 18/6/13	Fecha fin: 18/6/13
Programador responsable: Jairo Morales	
Descripción: Se utilizó el objeto <i>HTTPCLIENT</i> de <i>JAVA</i> para acceder a los servicios web generados.	

Tabla 206: Reporte de historias y videos (App móvil) – Tarea Nro. 3

Fuente: Propia

- **Reporte de festivos, fotos y gastronomías (App móvil)**

Historia 26: Reporte de festivos, fotos y gastronomías (App móvil).		
Número	Nombre	Tiempo estimado
1	Creación de servicios web con inyección.	15 horas
2	Diseño de interfaz gráfica de <i>Android</i> .	15 horas
3	Consumir servicios web utilizando <i>Android</i> .	5 horas
		Total: 35 horas

Tabla 206: Tareas de reporte de festivos, fotos y gastronomías (App móvil)

Fuente: Propia

1. Creación de servicios web con inyección

Número tarea: 1	Número historia: 25
Nombre tarea: Creación de servicios web con inyección.	
Tipo de tarea : Programación	Tiempo estimado: 20 horas
Fecha inicio: 21/6/13	Fecha fin: 23/6/13
Programador responsable: Jairo Morales	
Descripción: Se implementó dos servicios web utilizando <i>PHP</i> y la notación para objetos <i>JSON</i> . Servicio web festivo: Retorna una estructura de datos con todos los días festivos. Servicios web foto: Retorna una estructura de datos con todas las fotos. Servicios web gastronomía: Retorna una estructura de datos con todas las gastronomías.	

Tabla 207: Reporte de festivos, fotos y gastronomías (App móvil) – Tarea Nro. 1

Fuente: Propia

2. Diseño de interfaz gráfica de Android

Número tarea: 2	Número historia: 25
Nombre tarea: Diseño de interfaz gráfica de <i>Android</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 24/6/13	Fecha fin: 25/6/13
Programador responsable: Jairo Morales	
Descripción: Para el diseño de la interfaz de usuario para <i>Android</i> , se implementó 3 botones: Festivo, Fotos y Gastronomías, los cuales tienen sus propias actividades	

Tabla 208: Reporte de festivos, fotos y gastronomías (App móvil) – Tarea Nro. 2

Fuente: Propia

3. Consumir servicios web utilizando Android.

Número tarea: 3	Número historia: 24
Nombre tarea: Consumir servicios web utilizando <i>Android</i> .	
Tipo de tarea : Programación	Tiempo estimado: 10 horas
Fecha inicio: 27/6/13	Fecha fin: 28/6/13
Programador responsable: Jairo Morales	
Descripción: Se utilizó el objeto <i>HTTPCLIENT</i> de <i>JAVA</i> para acceder a los servicios web generados.	

Tabla 209: Reporte de festivos, fotos y gastronomías (App móvil) – Tarea Nro. 3

Fuente: Propia

3.5.3 Pruebas de Iteración III

1. Especificación de Prueba: Reporte plano de sitios.

Historia de usuario 16

Versión <2.6>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
08/4/2013	2.6	Revisión	Jairo Morales
08/4/2013	2.7	Modificación	Jairo Morales

Tabla 210: Iteración III, Historial de revisiones Nro. 1

Fuente: Propia

En esta historia se verifica, que los datos introducción en el gestor, cumpla las especificaciones establecidas en la Historia 16.

Selección de provincia y ciudad correcta.

Descripción

Los valores introducidos por el administrador deben cumplir, con las reglas establecidas.

Condiciones de ejecución

El usuario deberá seleccionar una provincia y ciudad, para efectuar la búsqueda de sitios.

Entrada

Ingresar una categoría, por la cual se efectuará la búsqueda.

Resultado esperado

Tras la introducción los datos de una provincia y ciudad; sí los datos ingresados han sido correctos, se imprimirá un reporte de los sitios.

Evaluación de la prueba

Prueba satisfactoria.

Selección de provincia y ciudad sin información asociada.

Descripción

Si los datos seleccionados del menú provincia y ciudad no concuerdan, no se procederá a realizar la búsqueda.

Condiciones de ejecución

El usuario deberá seleccionar una provincia y ciudad, para efectuar la búsqueda de sitios.

Entrada

Seleccionar una provincia y ciudad.

Ingresar una categoría, por la cual se efectuará la búsqueda.

✚ Resultado esperado

Tras la introducción incorrecta de una provincia y ciudad; no procede con la búsqueda.

✚ Evaluación de la prueba

Prueba satisfactoria.

2. Especificación de Prueba: Reporte geo-referenciado

Historia de usuario 17

Versión <2.7>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
19/04/2013	2.7	Revisión	Jairo Morales
19/04/2013	2.8	Revisión	Jairo Morales

Tabla 211: Iteración III, Historial de revisiones Nro. 2

Fuente: Propia

En esta historia se verifica, que los usuarios, accedan hacia la página web y puedan realizar la búsqueda geo localizada de sitios.

De esta manera se controlará, a que la información a ingresarse cumpla con las especificaciones.

Selección de una provincia y ciudad correcta

✚ Descripción

Los valores introducidos por el administrador deben cumplir, con las reglas establecidas.

✚ Condiciones de ejecución

El usuario deberá seleccionar una provincia y ciudad, para efectuar la búsqueda geo localizada de sitios.

✚ Entrada

Seleccionar una provincia y ciudad.

Ingresar una categoría, por la cual se efectuará la búsqueda.

✚ Resultado esperado

Tras la introducción los datos de una provincia y ciudad; sí los datos ingresados han sido correctos, se imprimirá un reporte geo localizado de los sitios.

✚ Evaluación de la prueba

Prueba satisfactoria.

Selección de provincia y ciudad incorrecta

Descripción

Si los datos seleccionados del menú provincia y ciudad no concuerdan, no se procederá a realizar la búsqueda.

Condiciones de ejecución

El usuario deberá seleccionar una provincia y ciudad, para efectuar la búsqueda de sitios.

Entrada

Seleccionar una provincia y ciudad.

Ingresar una categoría, por la cual se efectuará la búsqueda.

Resultado esperado

Tras la introducción incorrecta de una provincia y ciudad; no procede con la búsqueda.

Evaluación de la prueba

Prueba satisfactoria.

3. Especificación de Prueba: Enlaces hacia historias

Historia de usuario 18

Versión <2.8>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
24/4/2013	2.8	Revisión	Jairo Morales
24/4/2013	2.9	Revisión	Jairo Morales

Tabla 212: Iteración III, Historial de revisiones Nro. 3

Fuente: Propia

En esta historia se verificar, que los usuarios accedan a los reportes, sin ningún tipo de restricciones

No es necesario que se ingrese información por parte del usuario, pues los enlaces no reciben como parámetro ninguna información.

Reporte de historias

Descripción

Los usuario de la red, acceden directamente hacia el enlace *historias.php*

El reporte contiene todas las historias de un sitio, dependiendo de la categoría.

Condiciones de ejecución

Acceso total, sin importar el tipo de perfil de usuario.

✚ Entrada

Presionar *click* sobre enlace historias.

✚ Resultado esperado

El reporte sí imprime las historias disponibles de un sitio encontrado.

✚ Evaluación de la prueba

Prueba satisfactoria.

4. Especificación de Prueba: Enlaces hacia *YouTube*

Historia de usuario 19

Versión <2.9>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
01/5/2013	2.9	Revisión	Jairo Morales
02/5/2013	3.0	Revisión	Jairo Morales

Tabla 213: Iteración III, Historial de revisiones Nro. 4

Fuente: Propia

En esta historia hay que comprobar, que los usuarios accedan a los reportes, sin ningún tipo de restricciones

No es necesario que se ingrese información por parte del usuario, pues los enlaces no reciben como parámetro ninguna información.

Reporte de enlaces *YouTube*

✚ Descripción

Los usuario de la red, acceden directamente hacia el enlace *videos.php*

El reporte contiene todos los videos de un sitio, dependiendo de la categoría.

✚ Condiciones de ejecución

Acceso total, sin importar el tipo de perfil de usuario.

✚ Entrada

Presionar *click* sobre enlace videos.

✚ Resultado esperado

El reporte sí imprime los videos disponibles de un sitio encontrado.

✚ Evaluación de la prueba

Prueba satisfactoria.

5. Especificación de Prueba: Enlaces hacia imágenes

Historia de usuario 20

Versión <3.0>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
09/5/2013	3.0	Revisión	Jairo Morales
09/5/2013	3.1	Modificación	Jairo Morales

Tabla 214: Iteración III, Historial de revisiones Nro. 5

Fuente: Propia

En esta historia hay que comprobar, que los usuarios accedan a los reportes, sin ningún tipo de restricciones

No es necesario que se ingrese información por parte del usuario, pues los enlaces no reciben como parámetro ninguna información.

Reporte de enlaces imágenes

🚦 Descripción

Los usuario de la red, acceden directamente hacia el enlace *imagenes.php*

El reporte contiene todas las imágenes de un sitio, dependiendo de la categoría.

🚦 Condiciones de ejecución

Acceso total, sin importar el tipo de perfil de usuario.

🚦 Entrada

Presionar *click* sobre enlace imágenes.

🚦 Resultado esperado

El reporte sí imprime las imágenes disponibles de un sitio encontrado, en forma de un carrusel.

🚦 Evaluación de la prueba

Prueba satisfactoria.

6. Especificación de Prueba: Enlaces hacia festivos

Historia de usuario 21

Versión <3.1>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
14/5/2013	3.1	Revisión	Jairo Morales
14/5/2013	3.2	Modificación	Jairo Morales

Tabla 215: Iteración III, Historial de revisiones Nro. 6

Fuente: Propia

En esta historia hay que comprobar, que los usuarios accedan a los reportes, sin ningún tipo de restricciones

No es necesario que se ingrese información por parte del usuario, pues los enlaces no reciben como parámetro ninguna información.

Reporte de enlaces festivos

+ Descripción

Los usuario de la red, acceden directamente hacia el enlace *festivos.php*

El reporte contiene todos los festivos de un sitio, dependiendo de la categoría.

+ Condiciones de ejecución

Acceso total, sin importar el tipo de perfil de usuario.

+ Entrada

Presionar *click* sobre enlace festivo.

+ Resultado esperado

El reporte sí imprime los festivos disponibles de un sitio encontrado.

+ Evaluación de la prueba

Prueba satisfactoria.

7. Especificación de Prueba: Encales hacia gastronomías

Historia de usuario 22

Versión <3.2>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
19/5/2013	3.2	Revisión	Jairo Morales
19/5/2013	3.3	Modificación	Jairo Morales

Tabla 216: Iteración III, Historial de revisiones Nro. 7

Fuente: Propia

En esta historia hay que comprobar, que los usuarios accedan a los reportes, sin ningún tipo de restricciones

No es necesario que se ingrese información por parte del usuario, pues los enlaces no reciben como parámetro ninguna información.

Reporte enlaces de gastronomías

Descripción

Los usuario de la red, acceden directamente hacia el enlace *gastronomías.php*
El reporte contiene todas las gastronomías de un sitio, dependiendo de la categoría.

Condiciones de ejecución

Acceso total, sin importar el tipo de perfil de usuario.

Entrada

Presionar *click* sobre enlace gastronomías.

Resultado esperado

El reporte sí imprime las gastronomías disponibles de un sitio encontrado, en forma de un carrusel.

Evaluación de la prueba

Prueba satisfactoria.

8. Especificación de Prueba: Enlaces hacia rutas

Historia de usuario 23

Versión <3.3>

Historial de Revisiones

Fecha	Versión	Descripción	Autor
27/5/2013	3.3	Revisión	Jairo Morales
28/5/2013	3.4	Modificación	Jairo Morales

Tabla 217: Iteración III, Historial de revisiones Nro. 8

Fuente: Propia

En esta historia hay que comprobar, que los usuarios accedan a los reportes, sin ningún tipo de restricciones

No es necesario que se ingrese información por parte del usuario, pues los enlaces no reciben como parámetro ninguna información.

Enlaces hacia rutas

+ Descripción

Los usuario de la red, acceden directamente hacia el enlace *rutas.php*

El reporte contiene todas las rutas de un sitio, dependiendo de la categoría.

+ Condiciones de ejecución

Acceso total, sin importar el tipo de perfil de usuario.

+ Entrada

Presionar *click* sobre enlace rutas.

+ Resultado esperado

El reporte sí imprime las rutas disponibles de un sitio encontrado, en forma de un carrusel.

+ Evaluación de la prueba

Prueba satisfactoria.

9. Especificación de Prueba: Reporte plano de sitios (*App móvil*)

Historia de usuario 24

Versión <1.0>*APP MÓVIL*

Historial de Revisiones

Fecha	Versión	Descripción	Autor
14/6/2013	1.0	Revisión	Jairo Morales
14/6/2013	1.1	Modificación	Jairo Morales

Tabla 218: Iteración III, Historial de revisiones Nro. 9

Fuente: Propia

Se procede a comprobar si la historia 24 genera los reportes de los sitios turísticos.

Reporte de Sitios Turísticos

+ Descripción

Los usuarios deben instalar el aplicativo móvil, en dispositivos móviles que tengan un *SSOO Android* de fábrica.

+ Condiciones de ejecución

Conexión a internet.

+ Entrada

Los usuarios deberán seleccionar una provincia, ciudad y categoría.

Presionar click sobre el botón Sitios.

✚ Resultado esperado

El reporte sí imprime los sitios turísticos encontrados.

✚ Evaluación de la prueba

Prueba satisfactoria.

10. Especificación de Prueba: Reporte de historia y videos (*App móvil*)

Historia de usuario 25

Versión <1.1>APP MÓVIL

Historial de Revisiones

Fecha	Versión	Descripción	Autor
19/6/2013	1.1	Revisión	Jairo Morales
19/6/2013	1.2	Modificación	Jairo Morales

Tabla 219: Iteración III, Historial de revisiones Nro. 10

Fuente: Propia

Se procede a comprobar si la historia 24 genera los reportes de los sitios turísticos.

Reporte de historias y videos.

✚ Descripción

Los usuarios deben instalar el aplicativo móvil, en dispositivos móviles que tengan un SSOO *Android* de fábrica.

✚ Condiciones de ejecución

Conexión a internet.

✚ Entrada

Los usuarios deberán seleccionar una provincia, ciudad y categoría.

El usuario debe de presionar sobre el botón historias o videos.

✚ Resultado esperado

Sí imprime los reportes respectivos.

✚ Evaluación de la prueba

Prueba satisfactoria.

11. Especificación de Prueba: Reporte de festivos, fotos y gastronomías (App móvil)

Historia de usuario 26

Versión <1.2>APP MÓVIL

Historial de Revisiones

Fecha	Versión	Descripción	Autor
29/2/2013	1.2	Revisión	Jairo Morales
30/2/2013	1.3	Modificación	Jairo Morales

Tabla 220: Iteración III, Historial de revisiones Nro. 11

Fuente: Propia

Se procede a comprobar si la historia 24 genera los reportes de los sitios turísticos.

Reporte de historias y videos.

✚ Descripción

Los usuarios deben instalar el aplicativo móvil, en dispositivos móviles que tengan un SSOO *Android* de fábrica.

✚ Condiciones de ejecución

Conexión a internet.

✚ Entrada

Seleccionar del menú provincia, ciudad y categoría.

El usuario debe de presionar sobre el botón festivo, fotos o gastronomías.

✚ Resultado esperado

Sí imprime los reportes respectivos.

✚ Evaluación de la prueba

Prueba satisfactoria.

3.5.4 Resultado de la 3ª Iteración de construcción

Se tuvo retrasos en los tiempos establecidos, ya que el periodo de desarrollo se extendió hacia 1 y dos días, los cuales fueron controlados a su debido tiempo, acelerando el desarrollo en tareas posteriores.

Plan de entrega inicial

Consta de 8 historias de usuario:

Historia 16: Reporte plano

Historia 17: Reporte geo referenciado

Historia 18: Enlaces hacia historias

Historia 19: Enlaces hacia *YouTube*

Historia 20: Enlaces hacia imágenes

Historia 21: Enlaces hacia festivos

Historia 22: Enlaces hacia gastronomías

Historia 23: Enlaces hacia rutas

Historia 24: Reporte plano de sitios (App móvil)

Historia 25: Reporte de historias y videos (App móvil)

Historia 26: Reporte de festivos, fotos y gastronomías (App móvil)

Versión sistema web 2.6

Versión App móvil 1.3

Modificación de requerimientos

Historia 16: Reporte plano (Se mantiene)

Historia 17: Reporte geo referenciado (Se mantiene)

Historia 18: Enlaces hacia historias (Se mantiene)

Historia 19: Enlaces hacia *YouTube* (Se mantiene)

Historia 20: Enlaces hacia imágenes (Se mantiene)

Historia 21: Enlaces hacia festivos (Se mantiene)

Historia 22: Enlaces hacia gastronomías (Se mantiene)

Historia 23: Enlaces hacia rutas (Se mantiene)

Historia 24: Reporte plano de sitios (App móvil) (Se mantiene)

Historia 25: Reporte de historias y videos (App móvil) (Se mantiene)

Historia 26: Reporte de festivos, fotos y gastronomías (App móvil) (Se mantiene)

3.5.5 Demo de la versión

Historia 16: Reporte plano de sitios (5 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Después deberá presionar sobre el botón texto y observara un reporte de la información solicitada con sus características principales, como página *web*, cuenta *Facebook*, cuenta *twitter*, teléfono y entre

Imbabura Ibarra HOTELES Texto Mapa

AJAVÍ

noapicable
hotelajavi@andinanet.net
AV. MARIANO ACOSTA 16-38
2955221
@noapicable
/noapicable

Una de las técnicas más empleadas por los departamentos de comunicación organizacional y de recursos humanos de las empresas para motivar el trabajo de las diferentes áreas, es el uso de frases inspiradoras que incentiven la creatividad y sobre todo la responsabilidad de cada empleado y de los grupos de trabajo..

COLOMBIA CONFORT

noapicable
juliosoriocesar@hotmail.com
AVDA. JAIME RIVADENEIRA #222 Y ELIAS ALMEIDA
2604058
@noapicable
/noapicable

Una de las técnicas más empleadas por los departamentos de comunicación organizacional y de recursos humanos de las empresas

COCINA
FIESTAS
FOTOS
HISTORIAS
RUTAS
VIDEOS
LOCALIZACIÓN

COCINA
FIESTAS
FOTOS
HISTORIAS
RUTAS
VIDEOS

Ilustración 56: Formulario para buscar sitios turísticos

Fuente: Propia

Historia 17: Reporte geo-referenciado (10 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Des pues deberá presionar sobre el botón Mapa y observara un reporte de marcadores, para visualizar la información deberá presionar sobre el marcador correspondiente.

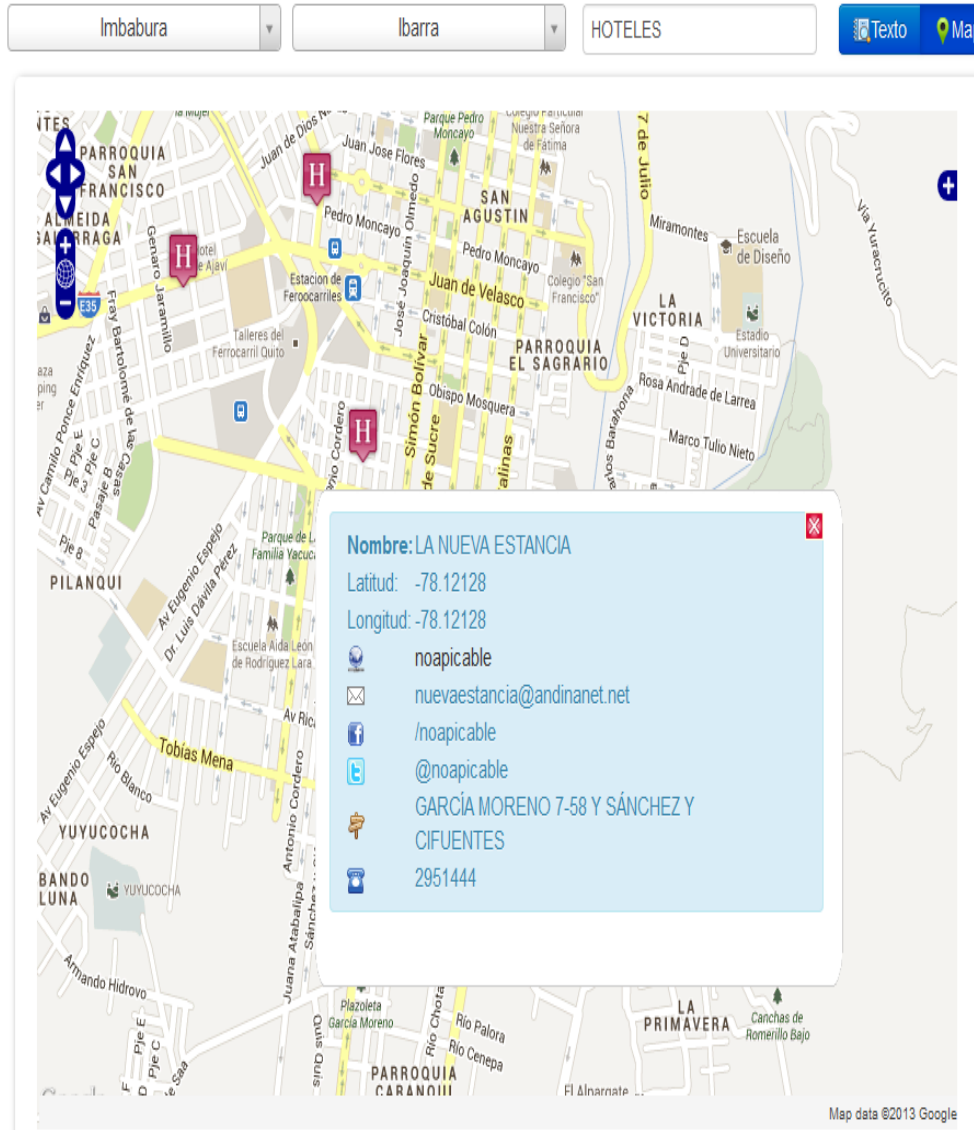


Ilustración 57: Formulario para realizar una geo-búsqueda

Fuente: Propia

Historia 18: Enlaces hacia historias (5 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón historias, el usuario visualizará una nueva página con las historias.



Ilustración 58: Enlaces hacia historias

Fuente: Propia

Historia 19: Enlaces hacia YouTube (5 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón videos, el usuario visualizará una nueva página con los videos.

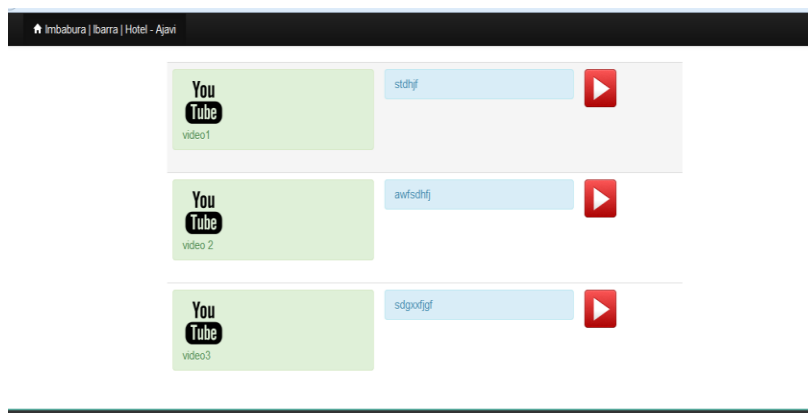


Ilustración 59: Enlaces hacia videos YouTube

Fuente: Propia

Historia 20: Enlaces hacia imágenes (7 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón imágenes, el usuario visualizara una nueva página con las imágenes.

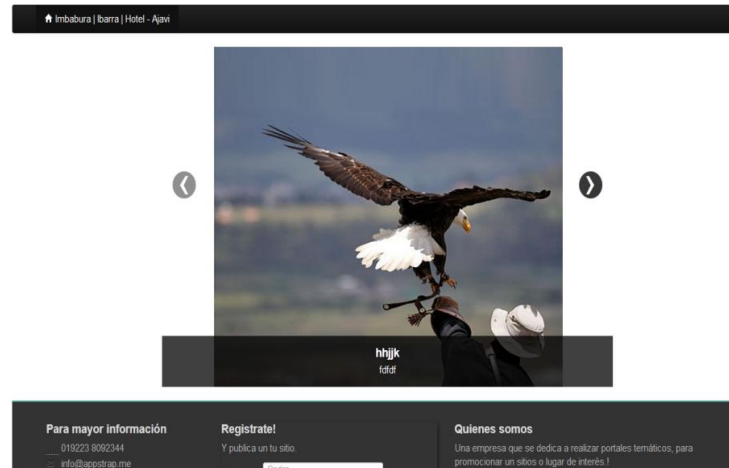


Ilustración 60: Enlaces hacia imágenes

Fuente: Propia

Historia 21: Enlaces hacia festivos (3 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón festivo, el usuario visualizara una nueva página con los días festivos.



Ilustración 61: Enlaces hacia días festivos

Fuente: Propia

Historia 22: Encales hacia gastronomías (5 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón gastronomías, el usuario visualizara una nueva página con las gastronomías.

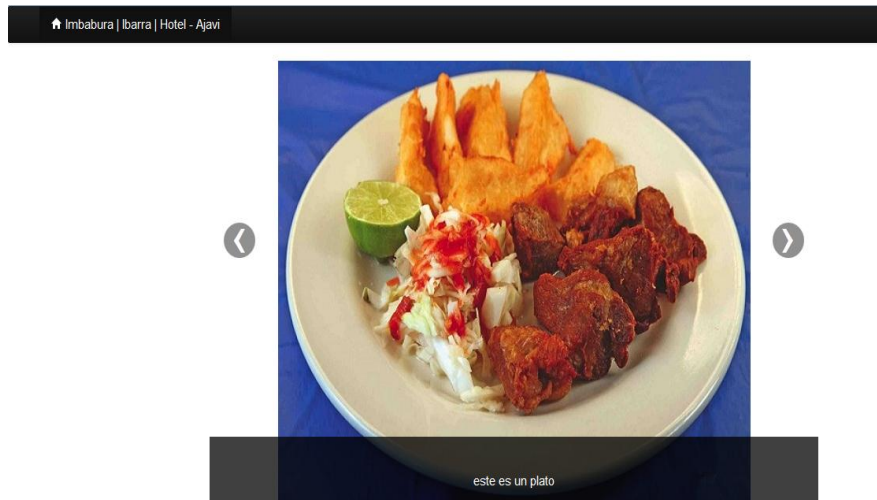


Ilustración 62: Enlaces hacia gastronomías

Fuente: Propia

Historia 23: Enlaces hacia rutas (7 días)

El usuario debe seleccionar una provincia y una ciudad, posteriormente digitar el tipo de categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón rutas, el usuario visualizara una nueva página con las rutas

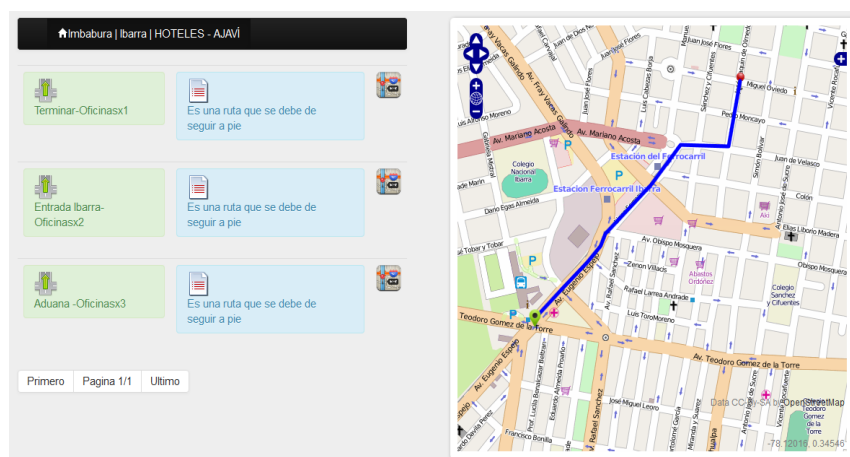


Ilustración 63: Enlaces hacia rutas

Fuente: Propia

Historia 24: Reporte Plano de sitios (App móvil, 10 días)

El usuario debe seleccionar una provincia, ciudad y posteriormente una categoría a buscar como: hoteles, hosterías, parques temáticos, museos y entre otros.

Al presionar sobre el botón Buscar, el usuario visualizara una nueva actividad con los sitios turísticos

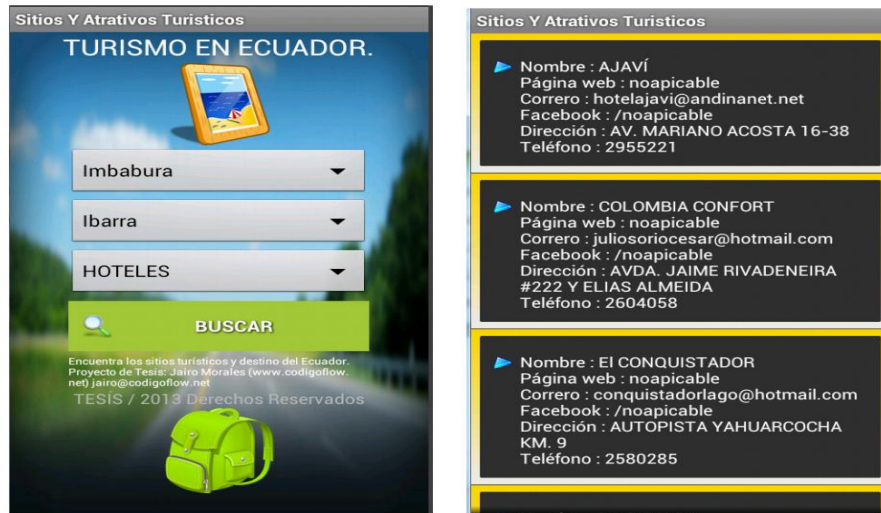


Ilustración 64: Reporte plano de sitios turísticos (App móvil)

Fuente: Propia

Historia 25: Reporte de Historias y Videos (App móvil, 7 días)

Botón historias: Obtienen un reporte de historias, sobre un sitio en específico.

Botón videos: Obtienen un reporte de videos, sobre un sitio en específico.



Ilustración 65: Reporte de historias y videos (App móvil)

Fuente: Propia

Historia 26: Reporte de festivos, fotos y gastronomías (App móvil, 7 días)

Botón festivo: Obtienen un reporte de festivos, sobre un sitio en específico.

Botón fotos: Obtienen un reporte de fotos, sobre un sitio en específico.

Botón gastronomías: Obtienen un reporte de gastronomías, sobre un sitio en específico.



Ilustración 66: Reporte de festivos, fotos y gastronomías (App móvil)

Fuente: Propia

3.5.6 Pruebas de aceptación

Historia 16: Reporte plano

Acceso hacia las interface de reporte, sí permite visualizar los reportes de la información requerida

- Pantalla *index.php*, sí tiene los menús funcionales, que permite al usuario, seleccionar una provincia y ciudad.
- El campo buscar, sí auto complementa la palabra, sin necesidad de escribirla por completo.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 17: Reporte geo-referenciado

Acceso hacia las interface de reporte geo-localizado, sí permite visualizar los marcadores con la información requerida

- Pantalla *index.php*, sí tiene los menús funcionales, que permite al usuario, seleccionar una provincia y ciudad.
- El campo buscar, sí auto complementa la palabra, sin necesidad de escribirla por completo.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 18: Enlaces hacia historias

Acceso hacia las interface de reporte historias, sí permite visualizar las historias de un sitio, con su respectivo nombre y descripción.

- Pantalla *historias.php*, sí tiene la información requerida, por el usuario.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 19: Enlaces hacia YouTube

Acceso hacia las interface de reporte videos, sí permite visualizar los videos de sitio, con su respectivo nombre y descripción y URL.

- Pantalla *videos.php*, sí tiene la información requerida, por el usuario.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 20: Enlaces hacia imágenes

Acceso hacia las interface de reporte imágenes, sí permite visualizar las imágenes sobre un carrusel interactivo.

- Pantalla *imagenes.php*, sí tiene la información requerida, por el usuario.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 21: Encales hacia festivos

Acceso hacia las interface de reportes festivos, sí permite visualizar los festivos de un sitio.

- Pantalla *festivos.php*, sí tiene la información requerida, por el usuario.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 22: Encales hacia gastronomías

Acceso hacia las interface de reporte gastronomías, sí permite visualizar las gastronomías sobre un carrusel interactivo.

- Pantalla *gastronomías.php*, sí tiene la información requerida, por el usuario.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 23: Enlaces hacia rutas.

Acceso hacia las interface de reporte rutas, sí permite visualizar las rutas de un sitio en específico,

- Pantalla *rutas.php*, sí tiene la información requerida, por el usuario.
- El usuario puede registrarse, sin una dependencia del administrador de la aplicación.

Historia 24: Reporte plano de sitios (App móvil).

Acceso hacia la actividad de reportes de sitios turísticos sí permite visualizar los sitios encontrados.

Historia 25: Reporte de historias y videos (App móvil)

Acceso hacia las interface de reporte historia, sí permite visualizar las historias de un sitio en específico,

Acceso hacia las interface de reporte videos, sí permite visualizar los videos de un sitio en específico

Historia 26: Reporte de festivos, fotos y gastronomías (App móvil)

Acceso hacia las interface de reporte festivos, sí permite visualizar los festivos de un sitio en específico,

Acceso hacia las interface de reporte videos, sí permite visualizar los videos de un sitio en específico

Acceso hacia las interface de reporte gastronomías, sí permite visualizar las gastronomías de un sitio en específico

3.5.7 Incidencias

- Inicialmente los servicios web no se generaban correctamente.
- El acceso hacia la API de Google maps fue denegado, por esta razón no se implementó un visor de mapas, para el aplicativo móvil.

3.5.8 Diagrama de base de Base de datos

Después de finalizar la tercera iteración y culminar con las pruebas de aceptación, se presenta el modelo final de la base de datos, que es el resultado de un modelado constante, pues de esta manera ajustamos el sistema a los requerimientos reales del usuario.

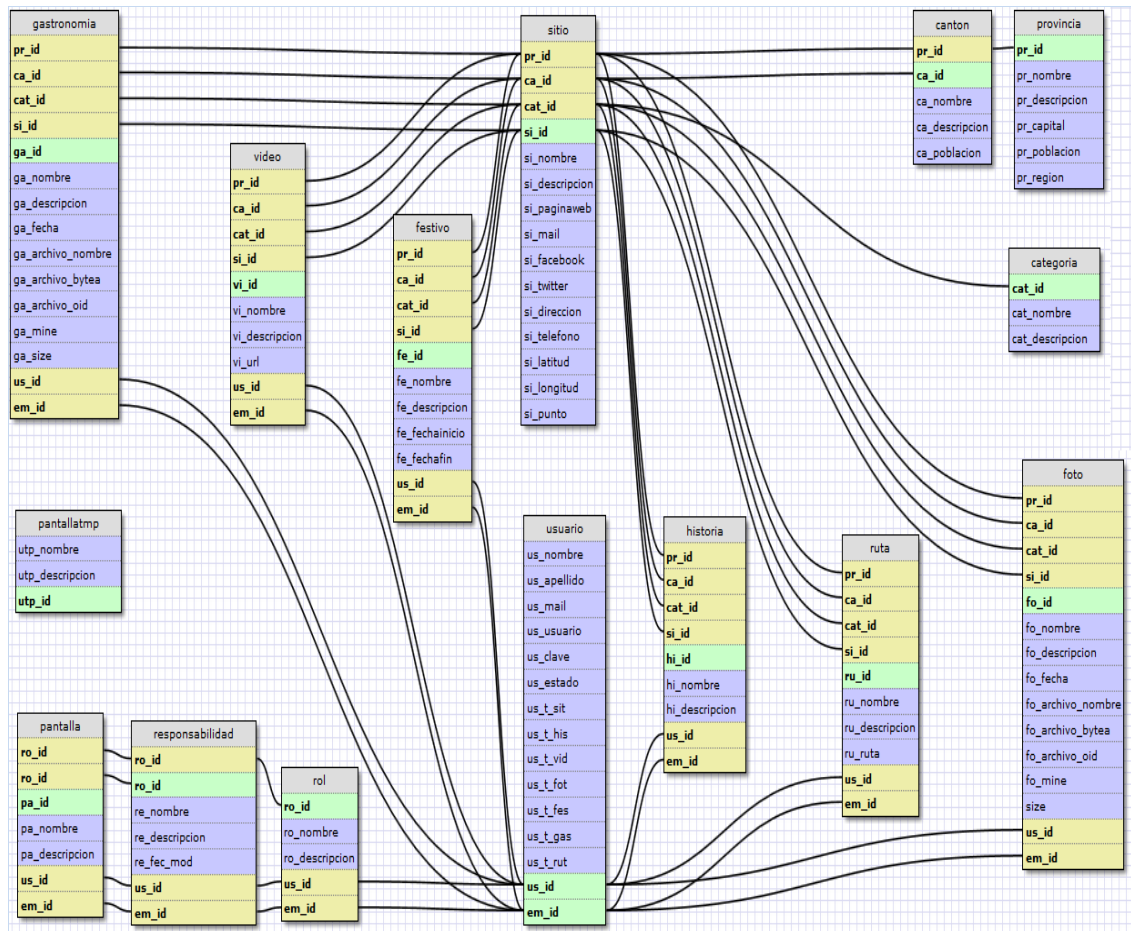


Ilustración 67: Modelo físico de la base de datos

Fuente: Propia

3.6 Producción

De acuerdo a la arquitectura se proceda a configurar el servidor privado virtual (Cloud VPS) de las siguientes características:

Sistema Operativo: Centos 5.5 | 64 bits.

Almacenamiento en Disco: 30GB

Memoria RAM: 1 GB.

3.6.1 Pasos para la configuración de servidor LINUX

En vista de que nuestro servidor esta disponible en linea, procedemos a instalar las herramientas libres, que sirvan para alojar la aplicación y brindar el servicio a los clientes.

Fue necesaria una conexión remota.

- 1.- Instalación de postgresql 9.1.
- 2.- Instalación de postGIS.
- 3.- Instalación de apache.
- 4.- Instalación de *PHP*
- 5.- Instalación de OSSEO.

3.6.2 Publicar la aplicación móvil en Google Play

Crear cuenta de Google

Esta podrá ser una cuenta GMail, Youtube, Google Apps. o cualquier de los servicios que proporciona Google. Esta cuenta estará asociada a la app y valdrá para realizar futuras modificaciones.

Acceder a la consola Google Play para desarrolladores

La consola de Google Play será el centro donde controlaremos todas las apps, su estado y estadísticas. Se accede desde <https://play.google.com/apps/publish/v2/> con la cuenta GMail que anteriormente hemos creado o queremos usar.

La primera vez que accedamos, deberemos registrar dicha cuenta como desarrollador pagando la cuota de 25\$.

Subir App

Seguidamente, haremos click en Añadir nueva aplicación y subiremos los archivos:

APK: Archivo generado tras la compilación del proyecto

Capturas de pantalla: Al menos un par y se pueden hacer desde el simulador

Icono de alta resolución: como emblema en Play Store

A continuación especificaremos los detalles de la app como nombre, descripción, tipo, idioma, categoría opciones de publicación, información de contacto.

Deberemos rellenar cuidadosamente esta información ya que será la que se verá públicamente como información del app.

na vez hecho el envío, podremos crear traducciones para diferentes idiomas, capturas de pantalla, gráficos de funciones, iconos de alta resolución, URL de vídeo de YouTube y gráficos promocionales localizados geográficamente.

CAPÍTULO IV

4 Análisis Costo-Beneficio, Conclusiones y Recomendaciones

4.1 Análisis de Costo-beneficio

El análisis costo-beneficio es una herramienta financiera que mide la relación entre los costos y beneficios asociados a un proyecto de inversión con el fin de evaluar su rentabilidad, entendiéndose por proyecto de inversión no solo como la creación de un nuevo negocio, sino también, como inversiones que se pueden hacer en un negocio en marcha, tales como el desarrollo de nuevo producto o la adquisición de nueva maquinaria.

4.1.1 Objetivo

Determinar la viabilidad de un proyecto, mediante el análisis de costos.

Costos de Hardware

Descripción	Costo estudiante \$	costo real \$
Computador	800.00	800.00
Total hardware	800.00	800.00

Costos de Servidor

Descripción	Costo estudiante \$	costo real \$
Google play	25.00	25.00
Servidor Privado Virtual (VPS)	340.00	340.00
Total hardware	365.00	365.00

Costos de software

Descripción	Costo estudiante \$	costo real \$
Bases de datos PostgreSQL	00.00	000.00
Costo de desarrollo (8 meses a 400)	3200.00	3200.00
Total software	3200.00	3200.00

Materiales de oficina

Descripción	Costo estudiante \$	costo real \$
Copias (documentos, libros)	50.00	20.00
DVD'S, esferográficos	10.00	10.00
Memoria flash	30.00	20.00
Total Materiales de oficina	90.00	90.00

✚ Varios

Descripción	Costo estudiante \$	costo real \$
Movilización	50.00	50.00
Libros	75.00	75.00
Internet	150.00	150.00
Total de varios	275.00	275.00
COSTOS TOTALES	4730.00	4730.00

Tabla 221: Tabla de costos

Fuente: Propia

4.1.3 Políticas de precio

Entre los ingresos que se obtengan, vendrán por concepto de publicad, pues se enviara propuestas, para promocionar sitios turísticos y atractivos turísticos.

✚ Tabla de cotización

Los atractivos turísticos públicos y que no cobren por su acceso, podrán publicarse gratuitamente.

Por un tiempo ilimitado.

Sitio	Gastronomía	Historia	URL (YouTube)	Foto	Festivo	Ruta	Plan	Tiempo
1	1	1	1	1	1	1	Gratis	1 semana
2	6	4	8	6	10	8	20 \$	Anual
3	12	6	12	12	15	12	25 \$	Anual
4	16	8	16	16	20	16	35 \$	Anual
5	20	10	20	20	25	20	40 \$	Anual

Tabla 222: Tabla de cotización

Fuente: Propia

4.1.4 Ejemplo Ilustrativo

Se desea determinar si nuestro proyecto seguirá siendo rentable y para ello decidimos analizar la relación costo-beneficio para los próximos 2 años.

Se estima recibir los siguientes ingresos cada mes, por concepto de publicar información de sitios turísticos, durante dos periodos.

a) Encontrar la predicción del valor que tomará el ingreso para dos periodos (2 años)

Días	Valor
30	100 \$
60	150 \$
90	200 \$
120	300 \$
150	350 \$
180	400 \$
210	410 \$
240	300 \$
270	350 \$
300	450 \$
330	450 \$
360	500 \$

Tabla 223: Tabla de valores

Fuente: Propia

Solución:

1.- Tabla de ingresos durante el primer periodo

Mes	Días	Valor
1	30	100 \$
2	60	150 \$
3	90	200 \$
4	120	300 \$
5	150	350 \$
6	180	400 \$
7	210	410 \$
8	240	300 \$
9	270	350 \$
10	300	450 \$
11	330	450 \$
12	360	500

Tabla 224: Tabla de ingresos

Fuente: Propia

2.- Realizamos una regresión lineal utilizando software propio. y obtenemos el siguiente gráfico

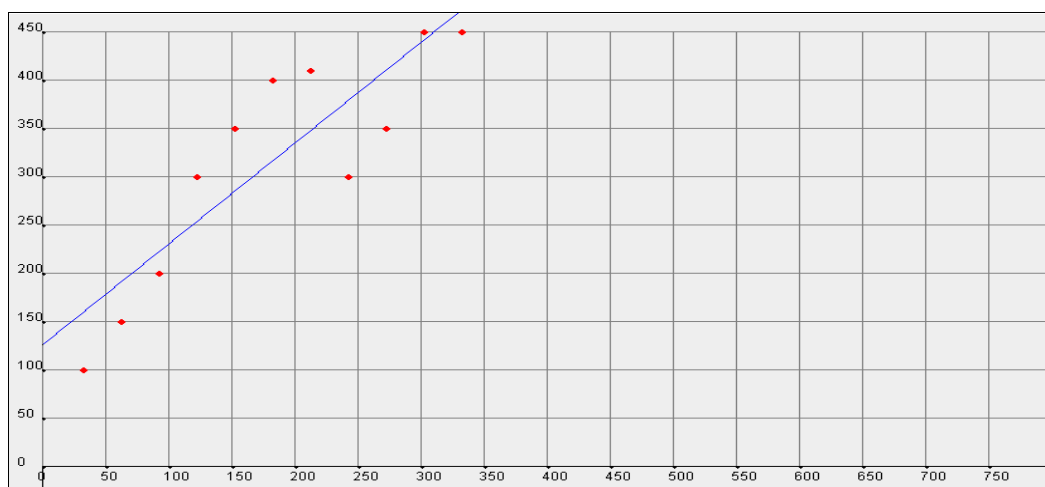


Ilustración 68: Regresión lineal

Fuente: Propia

3.- Considerando que el ingreso se puede expresar como una función lineal, a continuación tenemos la siguiente ecuación:

$$y(x) = a + b(x)$$

4.- Reemplazamos x en la ecuación.

$$X=720$$

$$y(x) = 126.363 + 1.044 * (x)$$

$$y(x) = 878.043$$

La predicción de los ingresos en dos años es: \$ 878.043.

4.1.5 Conclusiones del análisis Costo-beneficio

INVERSIÓN: US \$ 4730.00

INGRESO: US\$ 878,043

Cabe recalcar que los costos por concepto de promocionar y vender el producto pueden aumentar considerablemente, así también los ingresos por la promoción de sitios turísticos.

4.1.6 Riesgos y oportunidades

- **Riesgos internos**

- Poca disponibilidad de los servidores en línea.
- Incompatibilidad de las versiones del sistema operativo Android.
- Retraso en la publicación de la información, por problemas eventuales de conexión.
- Interrupciones en las bases de datos relacionales.
- Que los ingresos sean menores que los gastos.

- **Riesgos Externos.**

- Altos precios, por concepto de alquilar un VPS (Servidor Privado Virtual).
- La competencia.
- Que las empresas vinculadas, cierren el mercado.
- Los clientes pueden tomar en cuenta otros servicios, para promocionar su información turística.
- Falta de interés por las empresas para ingresar su información.

- **Oportunidades**

- Explotar el área del desarrollo de las aplicaciones móviles.
- Los sistemas GIS están aumentando considerablemente.
- Explotar las herramientas GIS gratuitas y libres.

4.2 CONCLUSIONES

- Un proyecto de software libre se desarrolla gracias al trabajo de la comunidad de desarrolladores, quienes trabajan en nuevos estándares de calidad y los pone a disposición de la comunidad, para un uso posterior y modificación de los códigos fuentes.
- La utilización de software libre constituye una alternativa eficiente para el desarrollo de aplicaciones web, pues no se limita a gestionar información crítica; en el caso de sistemas de información geográfica.
- La utilización de una base de datos libre nos permite trabajar con diversas herramientas de lado del cliente, sin importar el estándar o modelos adoptados.
- El empleo del motor de base de datos PostgreSQL, para la creación y mantenimiento de la información permitió disminuir los costos de desarrollo del sistema; pues los costos por concepto de licencias disminuyeron notablemente.
- El uso de la metodología de desarrollo ágil XP (Programación Extrema), permitió desarrollar el sistema en base a las necesidades reales del usuario, ya que las historias de usuario contribuyeron a definir correctamente la funcionalidad y propósito de la aplicación.
- La implementación de *TWITTER BOOTSTRAP MASTER* mejoró notablemente la interfaz de usuario, pues la mayoría de los componentes de diseño, son estándares utilizados por la mayoría de desarrolladores web, pues la mayoría de sus diseños y plantillas incorporan HTML 5 y CSS3. Así brindando una interface amigable al usuario.
- El uso de *AJAX*, en las aplicaciones web, reduce en gran medida los tiempos de respuesta entre el cliente y el servidor, pues *AJAX* consume recursos locales, por lo cual mantiene al servidor altamente eficiente y generando un importante avance en la forma de percibir los sistemas.
- *AJAX* es un mecanismo para enviar peticiones al servidor, no un estándar general, por lo cual el equipo de desarrolladores pueden determinar si su uso cumple con las necesidades y especificaciones de la arquitectura de software planteada, en la fase de exploración.
- El alcance de la aplicación aumento, ya que los mapas en línea de OSM (Open StreetMaps) y Google cuentan con la infraestructura espacial de todo el Ecuador.
- La integración de reportes sobre un dispositivo móvil, cumple con los objetivos planteados inicialmente, gracias a que los servicios *WebRest*, sí envían la

información desde un servidor Linux, hacia una aplicación móvil con sistema operativo *Android*, publicada en la tienda *Google Play (Android Market)*.

- *Netbeans* permitió aumentar la productividad de desarrollo, al generar código *HTML* y *PHP* sin mucho esfuerzo, gracias a su autocompletado y sugerencias de código. Eclipse se acopla a la *API* de *Android*, El uso de generadores de código es opcional, pues existen una variedad de herramientas que nos ayudan a optimizar la escritura.
- Cuando se prueba nuevas tecnologías, para una aplicación distribuida, se corre el riesgo, de que la arquitectura prevista a utilizar, no pueda desempeñar las funciones requeridas de acorde a los objetivos, por lo cual, se recomienda verificar si las plataformas separadas, poseen los estándares adecuados para la transferencia de información.

4.3 RECOMENDACIONES

- Utilizar herramientas y estándares necesarios dependiendo de la naturaleza de la aplicación o proyecto a implementarse, la arquitectura propuesta cumple con el objetivo de gestionar información plana y geográfica, tanto en una aplicación web y móvil.
- Durante la fase de desarrollo es necesario tener escrituras *SQL* por separado, para gestionar la información s nivel de base de datos, pues inicialmente no se dispone de formularios.
- Desarrollar la aplicación en una plataforma Linux, para evitar errores de compilación mientras se publica la aplicación en el servidor Web.
- Es vital buscar un equilibrio mientras desarrollamos una aplicación, pues lo ideal es combinar *AJAX* con los modelos tradicionales y actuales. El uso excesivo de *AJAX* no es recomendable, por la cantidad limitada de recursos en ciertos computadores, pues el objeto definido *HTTP* se almacena en la memoria chace del navegador.
- Utilizar *JSON* para la transferencia de archivos entre distintas plataformas, pues el flujo de información es más eficiente. *JSON* 224 bytes, considerando datos de *XML* 365 bytes.
- Antes de iniciar un proyecto de desarrollo de software, se debe establecer las alternativas de desarrollo existentes y definir correctamente un *MVC*, pues la arquitectura se integra a distintas plataformas de desarrollo: *.NETJ2EE* y entre otras.

- Se recomienda utilizar cartografía en línea, pues los mapas de *Google* y *OSM* cuentan con una infraestructura espacial más sólida y disponible. Consume los recursos del cliente para imprimir mapas en el navegador web con tiempos de espera efectivos.
- Se recomienda pulir el aplicativo móvil previa publicación, pues un ente humano califica si el APP estará disponible en la tienda de Google, además el tamaño no debe exceder los 5MB para un óptimo desempeño.

GLOSARIO

A

ADODB5.- Acceso a datos de objeto de una base de datos, es una librería que implementa una capa de abstracción de la Base de Datos.

AJAX.-, es una técnica de desarrollo web para crear aplicaciones interactivas.

API.-Está formado por un conjunto de comandos, funciones y protocolos que los programadores pueden utilizar en la construcción de software.

APK.-Archivo ejecutable de Android que te permite disfrutar del mercado Android de Google en un móvil o tableta.

ARCGIS.-Provee la estructura necesaria para la implementación de GIS para un solo usuario o múltiples accesos simultáneos.

B

BSD.-distribuciones de código fuente que se hicieron en la Universidad de Berkeley en California.

C

CSS.-hojas de estilo usado para describir la presentación semántica de un documento escrito en lenguaje HTML.

D

DAO.-, Objeto que proporciona un resumen de interfaz a algún tipo de base de datos u otro mecanismo de persistencia.

DBMS.-Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.

DOM.- Es una interfaz de la plataforma y del lenguaje neutro que permitirá a los programas y scripts acceder y actualizar dinámicamente el contenido, la estructura y el estilo de los documentos.

E

EPSG.- Organización científica vinculada a la industria del petróleo europea. Estaba formada por especialistas que trabajaban en el campo de la geodesia, la topografía y la cartografía aplicadas en relación con la exploración petrolífera.

G

GEOS. - Biblioteca de código abierto que proporciona herramientas para datos geoespaciales.

GIS.- Sistema diseñado para capturar, almacenar, manipular, analizar, administrar y presentar todo tipo de datos geográfico.

H

HTML.- Es un lenguaje de marcado para la elaboración de páginas web.

I

IIS.- Servidor web y conjunto de servicios para el sistema operativo Microsoft Windows.

J

J2EE.- Entorno centrado de SUN para desarrollar, construir y desplegar aplicaciones empresariales basadas en Web en línea.

JDBC.- API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

JRE.- Conjunto de utilidades que permite la ejecución de programas Java.

JSON.- Es un formato ligero para el intercambio de datos.

M

MD5.- Codificación utilizada para dar seguridad a las contraseñas o passwords guardadas en una base de datos.

MEZOGIS.- Proporcionar herramientas para el análisis geo-espacial con PostGIS.

O

OBDC.- Es un estándar de acceso a Bases de Datos desarrollado por Microsoft Corporation, el objetivo de ODBC es hacer posible el acceder a cualquier dato de cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos.

OGC.- Son un conjunto de tecnologías que facilitan la disponibilidad y el acceso a la información espacial haciendo uso de un conjunto de estándares y especificaciones.

OPENJUMP.- Es un Sistema de Información Geográfica de código abierto, escrito en el lenguaje de programación Java.

OSM.- Es un proyecto colaborativo para crear mapas libres y editables.

P

postGIS.- Extensión de Postgresql que añade soporte de objetos geográficos a la base de datos objeto-relacional.

PROJ4.- Proyecciones cartográficas originalmente escrito por Gerald Evenden.

R

REST.- Es un estilo arquitectónico que abstrae los elementos arquitectónicos dentro de un entorno distribuido.

S

SDK.- Conjunto de herramientas y programas de desarrollo que permite al programador crear aplicaciones para un determinado paquete de software.

SHAPE.- Archivo que contienen información geográfica, de tipo puntos, líneas o polígonos, así también como una tabla de atributos asociada a los elementos que visualizamos.

SOAP.- Estándar que define cómo dos objetos en diferentes procesos puede comunicarse por medio de intercambio de datos XML.

SRID.- Identificador de sistema de referencia espacial es un identificador único asociado con un sistema de coordenadas específico, tolerancia y resolución.

U

UDDI.- Un estándar basado en XML para la creación de directorios en línea de los servicios Web.

URL.- Localizador de recursos uniforme, la primera parte de la dirección indica qué protocolo utilizar, la segunda parte especifica la dirección IP o nombre de dominio donde se localiza el recurso.

V

VPS.- Es una máquina virtual que se vende como un servicio por un servicio de alojamiento en Internet.

W

W3C.-Es una organización encargada de regular los estándares web.

WGS84.- Es un sistema de coordenadas geográficas mundial que permite localizar cualquier punto de la Tierra (sin necesitar otro de referencia) por medio de tres unidades dadas.

WS-1.- Es una organización diseñada para promover la interoperabilidad de los servicios Web a través de plataformas, sistemas operativos y lenguajes de programación.

WSDL.- Describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo.

X

XML.- Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.

Z

ZIGIS.- Conector de escritorio para PostGIS.

BIBLIOGRAFÍA

LIBROS Y PÁGINAS WEB

- programacion-extrema*. (2013). Obtenido de programacion-extrema: <http://programacion-extrema.wikispaces.com/5.+Ciclo+de+vida+y+fases>
- AngelFire. (2013). *estudio de factibilidad*. Obtenido de estudio de factibilidad: http://www.angelfire.com/dragon2/informatica/estudio_de_factibilidad.htm
- Aplicable, N. (2008). *Xpmetodologia*. Recuperado el 3 de Febrero de 2013, de Xpmetodologia: <https://sites.google.com/site/xpmetodologia/home>
- Developers, A. (2013). *New Cross-Platform Single Sign On*. Obtenido de New Cross-Platform Single Sign On: <http://developer.android.com/develop/index.html>
- Espinoza, E. (2013). *Slidr Share*. Obtenido de Slidr Share: <http://www.slideshare.net/elianaespinoza/ventajas-y-desventajas-de-los-servidores-apache-y1>
- Franco, J. M. (15 de 10 de 2001). *Servidores Web* . Obtenido de Servidores Web : <http://trevinca.ei.uvigo.es/~txapi/espanol/proyecto/superior/memoria/node20.html>
- GIRONÉS, J. T. (2011). *EL GRAN LIBRO DE ANDROID*. México: MARCOCOMBO.
- GIS, S. (2013). *Servicios GIS*. Obtenido de Servicios GIS: <http://www.pccomahue.com.ar/productos-y-servicios/servicios-gis>
- HOLZER, S. (2009). *PHP*. México: MC GRAW HILL.
- HSU, R. O. (2011). *POSTGIS IN ACTION*. Madrid: PUBLICATIOS CO.
- Javier, Á. P. (3 de NOV de 2011). *Aplicaciones distribuidas*. Obtenido de Aplicaciones distribuidas: <http://www.slideshare.net/Javierialv/seguridad-de-sistemas-distribuidos>
- JOE, F., JEREMY, M., & C, Z. N. (2006). *PROFESIONAL AJAX*. España: ANAYA.
- Kay., P. M. (Ocubre de 2004). *MS MVPs*. Obtenido de MS MVPs: <http://msmvps.com/blogs/pmackay/archive/2004/10/04/14900.aspx>
- KOTH, H. F. (2002). *FUNDAMENTOS DE BASES DE DATOS*. México: MC GRAW HILL.
- Librosweb. (2003). *Introducción a AJAX*. Obtenido de Introducción a AJAX: http://librosweb.es/ajax/capitulo_1.html
- Librosweb. (2013). *Capítulo 1. Introducción*. Obtenido de Capítulo 1. Introducción: http://librosweb.es/javascript/capitulo_1.html
- librosweb. (2013). *Capítulo 1. Introducción a AJAX*. Obtenido de Capítulo 1. Introducción a AJAX: http://librosweb.es/ajax/capitulo_1.html
- LLARIO, J. M. (2012). *POSTGIS II*. Valencia: UNIVERSIDAD DE VALENCIA ESPAÑA.

- ONess. (12 23:12 de January de 2005). *ONess* . Recuperado el 3 de Abril de 2013, de ONess : <http://oness.sourceforge.net/proyecto/html/ch05.html>
- PÉREZ, A. S. (2012). *OPENLAYERS COOKBOOK*. Madrid: OPEN SOURCE.
- Rabolini, N. M. (2013). *Técnicas de muestreo y determinación del tamaño de la muestra en investigación cuantitativa*. Obtenido de Técnicas de muestreo y determinación del tamaño de la muestra en investigación cuantitativa: http://www.sai.com.ar/metodologia/rahycs/rahycs_v7_n2_06.htm
- Rediam. (2013). *Códigos EPSG de Sistemas de Referencia*. Obtenido de Códigos EPSG de Sistemas de Referencia: http://www.juntadeandalucia.es/medioambiente/site/rediam/menuitem.04dc44281e5d53cf8ca78ca731525ea0/?vgnnextoid=2a412abcb86a2210VgnVCM1000001325e50aRCRD&lr=lang_es
- Rodriguez, T. (16 de JUN de 2012). *GENBETA*. Obtenido de GENBETA: <http://www.genbetadev.com/frameworks/bootstrap>
- Rodriguez, Y. (11 de Nov de 2011). *Metod.* Obtenido de Scrib: <http://www.scribd.com/doc/72420606/METODOLOGIA-XP>
- SIERRA, A. M. (2012). *AJAX EN J2EE*. Madrid: ALFAOMEGA.
- SORIANO, J. E. (2012). *EL GRAN LIBRO DE PROGRAMACIÓN AVANZADA CON ANDROID*. México: MARCOCOMBO.
- Thetuzaro. (2011). *La proyección de Mercator*. Obtenido de La proyección de Mercator: <http://thetuzaro.wordpress.com/2011/03/20/la-proyeccion-de-mercator-vs-la-proyeccion-de-gall-peters-2/>
- varstein, E. H. (2013). *European Petroleum Survey Group*. Obtenido de European Petroleum Survey Group: <http://www.ihsenergy.com/epsg/epsg21.html>
- Veiss. (2013). *Sitios Web Móvil Vs. Aplicaciones Móviles*. Obtenido de Sitios Web Móvil Vs. Aplicaciones Móviles: <http://www.veiss.com/blog/sitios-web-movil-vs-aplicaciones-moviles/>
- Web-developía. (2013). *Ciclo de vida en Android*. Obtenido de Ciclo de vida en Android: <http://www.web-developia.com/2011/desarrollo-movil/ciclo-de-vida-en-android/>

ENCUESTA DE TELEFONÍA CELULAR

Me permite un momento de su tiempo para que me pueda contestar la siguiente encuesta, la cual estamos realizando con motivo de una Investigación de Mercado con la finalidad de conocer, el uso de los teléfonos Android.

Correo:

Edad:

Fecha:

Subraye o escriba según corresponda.

1. **¿Cuál es la marca de su celular?**
 LG Samsung Ericsson Motorola HTC iPhone Nokia Alcatel
 BlackBerry Otro
2. **¿Cada cuánto tiempo cambia su celular?**
 1 a 6 meses 6 – 12 meses 1 año en adelante
3. **¿Su celular tiene acceso a Internet?**
 Total zonas wi-fi No
4. **¿Su celular tiene un Sistema Operativo Android?**
 Si No Desconoce
5. **¿Conoce de alguna aplicación móvil, para la promoción de sitios turísticos y atractivos de la ciudad de Ibarra, desarrollada en Ecuador?**
 SI NO Desconozco
6. **¿Considerando el precio, cómo se deberían promocionar los sitios turísticos y atractivos de una ciudad?**
 Aplicación móvil Televisión Prensa Escrita Radio Emisoras Páginas web
7. **¿Le gustaría probar una aplicación móvil, para la promoción de sitios turísticos de Ibarra?**
 SI NO

Muchas gracias por su atención.

ENCUESTA DE TELEFONIA CELULAR

Me permite un momento de su tiempo para que me pueda contestar la siguiente encuesta, la cual estamos realizando con motivo de una Investigación de Mercado con la finalidad de conocer, el uso de los teléfonos Android.

Correo: marceloamrango@hotmail.com

Edad: 27

Fecha: 10 de Junio del 2013

Subraye o escriba según corresponda.

1. ¿Cuál es la marca de su celular?
• LG • Samsung • Ericsson • Motorola • HTC • iPhone • Nokia • Alcatel
• BlackBerry • Otro
2. ¿Cada cuánto tiempo cambia su celular?
• 1 a 6 meses • 6 – 12 meses • 1 año en adelante
3. ¿Su celular tiene acceso a Internet?
• Total • zonas wi-fi • No
4. ¿Su celular tiene un Sistema Operativo Android?
• Si • No • Desconoce
5. ¿Conoce de alguna aplicación móvil, para la promoción de sitios turísticos y atractivos de la ciudad de Ibarra, desarrollada en Ecuador?
• SI • NO • Desconozco
6. ¿Considerando el precio, cómo se deberían promocionar los sitios turísticos y atractivos de una ciudad?
• Aplicación móvil • Televisión • Prensa Escrita • Radio Emisoras • Páginas web
7. ¿Le gustaría probar una aplicación móvil, para la promoción de sitios turísticos de Ibarra?
• SI • NO

Muchas gracias por su atención.