



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

**“SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS
TURÍSTICOS DE LA COMUNIDAD DE CARIACU CON LA UTILIZACIÓN DE
SOFTWARE LIBRE”**

AUTOR: CARLOS EFRAÍN CALUGUILLÍN YASCUAL.

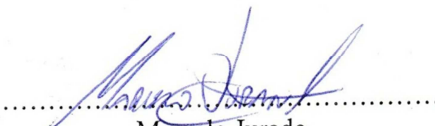
DIRECTOR: ING. EDWIN MARCELO JURADO ÁVILA

IBARRA – ECUADOR

2014

CERTIFICACIÓN

Certifico que la Tesis “**SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU CON LA UTILIZACIÓN DE SOFTWARE LIBRE**” con el aplicativo “**SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU**” ha sido realizada en su totalidad por el señor: Carlos Efraín Caluguillín Yascual portador de la cédula de identidad número: 171766023-5.



Marcelo Jurado
Director de la Tesis

CERTIFICACIÓN



**GOBIERNO AUTÓNOMO DESCENTRALIZADO INTERCULTURAL
Y PLURINACIONAL DEL MUNICIPIO DE CAYAMBE**

Cayambe, 14 de Julio del 2014
Cert. Acad. Nº 001 -DGDES- GADIPMC

CERTIFICADO

En calidad de Auspiciantes del proyecto de tesis del Egresado **CARLOS EFRAÍN CALUGUILLIN YASCUAL**, portador de la Cédula de identidad Nº 171766023-5, quien ha desarrollado su trabajo de grado previo a la obtención del título de Ingeniero en Sistemas Computacionales con el tema **"SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU CON LA UTILIZACIÓN DE SOFTWARE LIBRE"** tengo a bien certificar que luego de la revisión del cumplimiento de los requerimientos del mencionado proyecto, la Dirección de Desarrollo Económico del GADIP Municipal de Cayambe, se ha procedido con el recibimiento de la documentación respectiva y nos comprometemos a continuar con el apoyo necesario para la puesta en ejecución del mencionado Proyecto en beneficio de la Comunidad de Cariacu y la Actividad Turística del Cantón.

Además extendemos nuestros agradecimientos al Egresado y la Universidad por tomar en cuenta estos temas de beneficio colectivo para el cantón Cayambe.

El mencionado Egresado, Sr. Carlos Efraín Caluguillín Yascual, puede hacer uso del presente para los fines solicitados por la Universidad Técnica del Norte.

Cordialmente;



Ing. Braulio Novoa

**DIRECTOR DE GESTIÓN DE DESARROLLO ECONÓMICO SOSTENIBLE
GADM - CAYAMBE**



Dir: Terán SO-54 y Sucre **Telf:** 022 361 591 / 022 361 832 / 022 360 052
Web: www.municipiocayambe.gob.ec

CERTIFICACIÓN

Cariacu, 29 de Enero de 2014

Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Egresado CARLOS EFRAÍN CALUGUILLIN YASCUAL con CI: 171766023-5 quién desarrolló su trabajo con el tema "SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU CON LA UTILIZACIÓN DE SOFTWARE LIBRE" con el aplicativo "SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado CARLOS EFRAÍN CALUGUILLIN YASCUAL. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa.

El egresado CARLOS EFRAÍN CALUGUILLIN YASCUAL puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



Germánico Guacán *

Presidente

GRUPO DE ECOTURISMO "PAKCHAPI-PAKARI"



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

Yo, CARLOS EFRAÍN CALUGUILLÍN YASCUAL, con cedula de identidad Nro. 171766023-5, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, artículo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU CON LA UTILIZACIÓN DE SOFTWARE LIBRE”** con el aplicativo **“SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, 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 mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este 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: CARLOS EFRAÍN CALUGUILLÍN YASCUAL.

Cédula: 171766023-5

Ibarra a los 14 días del mes de Julio del 2014



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 determina la necesidad de disponer los textos completos de forma 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 ponemos a disposición la siguiente investigación:

| DATOS DE CONTACTO | |
|---------------------|--|
| CÉDULA DE IDENTIDAD | 171766023-5 |
| APELLIDOS Y NOMBRES | CARLOS EFRAÍN CALUGUILLÍN YASCUAL |
| DIRECCIÓN | Comunidad de Cariacu – Barrio Central – Cayambe |
| EMAIL | krlos_81@hotmail.com, cecaluguilliny@utn.edu.ec. |
| TELÉFONO FIJO | (02) 2129 190 |
| TELÉFONO MÓVIL | 0997 695 311 |

| DATOS DE LA OBRA | |
|------------------------|---|
| TÍTULO | “SISTEMA WEB DE GESTIÓN Y PROMOCIÓN DE LOS ATRACTIVOS TURÍSTICOS DE LA COMUNIDAD DE CARIACU CON LA UTILIZACIÓN DE SOFTWARE LIBRE” |
| AUTOR | CARLOS EFRAÍN CALUGUILLÍN YASCUAL |
| FECHA | 29 DE ENERO DEL 2014 |
| PROGRAMA | PREGRADO |
| TITULO POR EL QUE OPTA | INGENIERÍA EN SISTEMAS COMPUTACIONALES |
| DIRECTOR | MARCELO JURADO |

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, CARLOS EFRAÍN CALUGUILLÍN YASCUAL, con cédula de identidad Nro. 1717660235, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el 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 Artículo 144.



Firma

Nombre: CARLOS EFRAÍN CALUGUILLÍN YASCUAL.

Cédula: 171766023-5.

Ibarra a los 14 días del mes de Julio del 2014.

DEDICATORIA

A dios, por regalarme cada día en mi vida.

A mis padres Lorenzo y Erlinda por darme la vida, ser un ejemplo de vida, por su perseverancia y constancia en mi formación como hijo, persona, esposo y profesional. Gracias por todo su amor.

A mi esposa Andrea por estar junto a mí en todo momento y apoyarme en todo lo planificado. Gracias por tu comprensión.

A mis hermanos Carmen, Luis, Nancy, Marlene, Ivan por ser excelentes hermanos y amigos. Gracias hermanos.

A mis suegros Ignacio y Marcela por brindarme su ayuda y apoyarme en mi formación profesional.

Además este trabajo lo dedico a todos mis amigos de aula con los cuales compartí lindos momentos.

Carlos Caluguillín.

AGRADECIMIENTOS

Agradezco a dios por guiar mis pasos.

Agradezco de todo corazón a mis padres porque con su ayuda moral y económica y sus consejos he logrado concluir una etapa de mi vida estudiantil e iniciar una nueva etapa profesional.

Agradezco a mis hermanos y hermanas ya que su ayuda fue muy importante en mis estudios. Principalmente a Luis que es la mano derecha y apoyo de mi padre.

Agradezco a mi linda esposa, ya que enrumbo mi vida, su ayuda incondicional en los buenos y malos momentos.

Agradezco a los padres de mi esposa, quienes son mis segundos padres.

Agradezco a mis cuñadas y cuñado, con quienes he formado una linda amistad.

Agradezco al Ing. Marcelo Jurado, docente de la Facultad, quién me guió y ayudó en este trabajo final.

Agradezco a mis docentes quienes me formaron profesionalmente en las aulas de la imponente UTN – FICA.

Agradezco al grupo de ecoturismo de la comunidad de Cariacu por colaborar en el desarrollo de este proyecto.

Carlos Caluguillín.

RESUMEN

El presente documento contiene la información del proyecto “**Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu**”, el cual beneficia a la comunidad de Cariacu, ubicada en el cantón Cayambe, provincia de Pichincha.

Este documento se distribuye en cinco capítulos; en cada uno de ellos se describen los conceptos, procesos que se realizan para la elaboración del aplicativo.

El capítulo uno contiene una breve reseña de la Comunidad de Cariacu y el turismo comunitario, además contiene el problema que impulsó la creación de este proyecto.

En el capítulo dos se describe la metodología de desarrollo, arquitectura del sistema, las herramientas de software aplicadas y conceptos importantes para la construcción del software.

En el capítulo tres se encuentra el análisis de las herramientas usadas en el desarrollo del aplicativo.

En el capítulo cuatro se detalla la metodología usada en el proceso de desarrollo del software (SCRUM¹), con sus respectivos artefactos y/o documentos.

El capítulo cinco muestra las conclusiones, recomendaciones, e impacto del sistema desarrollado.

Palabras clave: Symfony, turismo comunitario, gestión y promoción, Cariacu, visita virtual, GIS, MapServer, Pmapper, ventas en línea, catálogo, productos y servicios.

¹SCRUM.- es un marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software.

SUMMARY

This document contains the information of the "Web system of management and promotion of Cariacu Community's tourist attractions", which benefits the Cariacu community, located in Cayambe, Pichincha province.

This document is distributed in five chapters; in each of them are described the concepts and processes made for the application.

Chapter one contains a brief overview of the Cariacu community and the community tourism, also contains the problem that impulse the creation of this project.

In chapter two describes the development methodology, system architecture, software tools and important concepts for the construction of the software.

In chapter three is the analysis of the tools used in the development of the application.

In the chapter four there is detailed the methodology used in the process of development of the software (SCRUM), with their respective artifacts and/or documents.

The chapter five shows the conclusions, recommendations, and impact of the developed system.

Key words: Symphony, Community Tourism, management and promotion, Cariacu, virtual tour, GIS, MapServer, Pmapper, online sales, catalog, products and services.

TABLA DE CONTENIDOS

Pág.

| | |
|---|----------|
| CERTIFICACIÓN | ii |
| CERTIFICACIÓN | iii |
| CERTIFICACIÓN | iv |
| CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN | v |
| AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE | vi |
| DEDICATORIA..... | viii |
| AGRADECIMIENTOS | ix |
| RESUMEN..... | x |
| SUMMARY | xi |
| TABLA DE CONTENIDOS | xii |
| ÍNDICE DE TABLAS | xviii |
| ÍNDICE DE GRÁFICOS | xxi |
| 1. CAPÍTULO I: INTRODUCCIÓN | 1 |
| 1.1. LA COMUNIDAD DE CARIACU Y EL TURISMO COMUNITARIO | 1 |
| 1.1.1. Antecedentes de la comunidad..... | 1 |
| 1.1.2. Turismo comunitario en la comunidad..... | 1 |
| 1.2. EL PROBLEMA..... | 2 |
| 1.2.1. Antecedentes..... | 2 |
| 1.2.2. Situación actual del turismo comunitario..... | 2 |
| 1.2.3. Planteamiento del problema..... | 2 |
| 1.3. OBJETIVOS..... | 3 |
| 1.3.1. Objetivo general..... | 3 |
| 1.3.2. Objetivos específicos..... | 3 |
| 1.4. ALCANCE..... | 3 |
| 1.5. JUSTIFICACIÓN..... | 4 |
| 2. CAPÍTULO II: MARCO TEÓRICO | 7 |
| 2.1. TEORÍA DEL DESARROLLO DEL SISTEMA..... | 7 |
| 2.1.1. Metodología ágil..... | 7 |
| 2.1.2. Scrum..... | 7 |
| 2.1.3. Marco estándar de Scrum..... | 7 |
| 2.1.4. Elementos de Scrum | 8 |
| 2.1.5. Reuniones de Scrum..... | 8 |
| 2.1.6. Roles de Scrum..... | 8 |

| | | |
|-------------|--|-----------|
| 2.2. | ARQUITECTURA DEL SISTEMA. | 8 |
| 2.3. | SERVIDOR DE APLICACIONES WEB. | 11 |
| 2.3.1. | Servidor web | 11 |
| 2.3.2. | Apache | 11 |
| 2.4. | SISTEMA GESTOR DE BASE DE DATOS (SGBD). | 11 |
| 2.4.1. | Definición. | 11 |
| 2.4.2. | Herramientas de administración. | 11 |
| 2.4.3. | PgAdmin 3 | 12 |
| 2.5. | BASE DE DATOS ESPACIAL. | 12 |
| 2.5.1. | PostGIS. | 12 |
| 2.6. | SISTEMAS DE INFORMACIÓN GEOGRÁFICA. | 12 |
| 2.6.1. | Antecedentes | 12 |
| 2.6.2. | Definición. | 13 |
| 2.6.3. | Funcionalidad de los SIG. | 13 |
| 2.6.4. | Componentes. | 13 |
| 2.6.5. | Cartografía. | 14 |
| 2.6.6. | Mapa. | 14 |
| 2.6.7. | Proyecciones. | 14 |
| 2.6.8. | Escala. | 15 |
| 2.6.9. | Modelos para la información geográfica. | 15 |
| 2.6.10. | Modelos geográficos. | 15 |
| 2.6.11. | Modelos de representación. | 15 |
| 2.6.12. | Modelos de almacenamiento. | 16 |
| 2.7. | SERVIDOR DE MAPAS. | 17 |
| 2.7.1. | MapServer. | 17 |
| 2.7.2. | PHP – MapScript. | 17 |
| 2.7.3. | Mapfile. | 17 |
| 2.7.4. | OGC. | 18 |
| 2.7.5. | Shape. | 18 |
| 2.7.6. | WFS. | 18 |
| 2.7.7. | WCS | 18 |
| 2.7.8. | WMS. | 19 |
| 2.7.9. | WMC. | 19 |
| 2.8. | CLIENTES LIVIANOS GIS. | 19 |
| 2.8.1. | Definición. | 19 |
| 2.8.2. | Características. | 20 |
| 2.8.3. | Pmapper. | 20 |
| 2.9. | CLIENTES PESADOS GIS. | 20 |

| | | |
|--------------|--|-----------|
| 2.9.1. | Definición..... | 20 |
| 2.9.2. | Características..... | 20 |
| 2.9.3. | Quantum GIS..... | 21 |
| 2.10. | REALIDAD VIRTUAL..... | 21 |
| 2.10.1. | Antecedentes..... | 21 |
| 2.10.2. | Definición..... | 21 |
| 2.10.3. | Visita virtual..... | 21 |
| 2.10.4. | Características de la visita virtual..... | 21 |
| 2.10.5. | Beneficios de las visitas virtuales..... | 22 |
| 2.10.6. | Fotografía panorámica..... | 22 |
| 2.10.7. | Fotografía 360°..... | 22 |
| 2.10.8. | Fotografía cilíndrica..... | 22 |
| 2.10.9. | Fotografía esférica..... | 23 |
| 2.10.10. | Fotografía cúbica..... | 23 |
| 2.11. | LENGUAJE DE PROGRAMACIÓN PHP..... | 23 |
| 2.11.1. | Antecedentes..... | 23 |
| 2.11.2. | Definición..... | 23 |
| 2.12. | FRAMEWORK SYMFONY..... | 23 |
| 2.12.1. | Definición..... | 23 |
| 2.12.2. | MVC en Symfony..... | 24 |
| 2.12.3. | ORM..... | 24 |
| 2.12.4. | Doctrine..... | 25 |
| 2.12.5. | DQL..... | 25 |
| 2.12.6. | YAML..... | 25 |
| 2.13. | COMERCIO ELECTRÓNICO..... | 25 |
| 2.13.1. | Antecedentes..... | 25 |
| 2.13.2. | Definición..... | 26 |
| 2.13.3. | PayPal..... | 26 |
| 2.14. | WEB 2.0..... | 26 |
| 2.14.1. | Antecedentes..... | 26 |
| 2.14.2. | Definición..... | 26 |
| 3. | CAPÍTULO III: HERRAMIENTAS DE DESARROLLO..... | 27 |
| 3.1. | INTRODUCCIÓN GENERAL..... | 27 |
| 3.2. | LENGUAJE DE PROGRAMACIÓN..... | 27 |
| 3.2.1. | Lenguaje HTML..... | 27 |
| 3.2.2. | Lenguaje JavaScript..... | 28 |
| 3.2.3. | Lenguaje PHP..... | 28 |
| 3.2.4. | XML..... | 30 |

| | | |
|-------------|---|-----------|
| 3.2.5. | CSS | 30 |
| 3.2.6. | AJAX..... | 31 |
| 3.2.7. | YAML..... | 32 |
| 3.2.8. | DQL | 34 |
| 3.3. | BASES DE DATOS POSTGRESQL | 35 |
| 3.3.1. | Introducción..... | 35 |
| 3.3.2. | Características principales..... | 35 |
| 3.3.1. | Arquitectura de PostgreSQL..... | 36 |
| 3.3.2. | Requerimientos hardware | 37 |
| 3.3.3. | Requerimientos software..... | 37 |
| 3.3.4. | Limitaciones. | 37 |
| 3.3.5. | Ventajas. | 38 |
| 3.3.6. | Desventajas..... | 38 |
| 3.4. | MAPSERVER | 38 |
| 3.4.1. | Introducción..... | 38 |
| 3.4.2. | Requerimientos hardware | 39 |
| 3.4.3. | Requerimientos software..... | 39 |
| 3.4.4. | Arquitectura básica de MapServer. | 39 |
| 3.4.5. | Funcionamiento de MapServer | 40 |
| 3.4.6. | Características principales..... | 41 |
| 3.4.7. | Ventajas. | 41 |
| 3.4.8. | Desventajas..... | 41 |
| 3.5. | IDE DE DESARROLLO – NETBEANS 7.0.1 | 41 |
| 3.5.1. | Introducción..... | 41 |
| 3.5.2. | Definición..... | 42 |
| 3.5.3. | Requerimientos de software..... | 42 |
| 3.5.4. | Requerimientos de hardware..... | 42 |
| 3.5.5. | Características..... | 42 |
| 3.5.6. | Ventajas..... | 43 |
| 3.5.7. | Desventajas..... | 43 |
| 3.5.8. | Integración con Symfony..... | 43 |
| 3.6. | FRAMEWORK SYMFONY 1.4.16 | 44 |
| 3.6.1. | Introducción..... | 44 |
| 3.6.2. | Pre requisitos para instalación..... | 44 |
| 3.6.3. | Características de Symfony | 44 |
| 3.6.4. | Implementación MVC de Symfony. | 45 |
| 3.6.5. | Estructura del proyecto..... | 46 |
| 3.6.6. | Estructura del árbol de archivos..... | 47 |

| | | |
|--------------|---|-----------|
| 3.6.7. | Los entornos | 48 |
| 3.6.8. | Sistema de configuraciones | 49 |
| 3.6.9. | El Modelo (DOCTRINE) | 50 |
| 3.6.10. | Ventajas..... | 50 |
| 3.6.11. | Desventajas..... | 51 |
| 3.7. | POSTGIS..... | 51 |
| 3.7.1. | Introducción..... | 51 |
| 3.7.2. | Requerimientos..... | 51 |
| 3.7.3. | Características únicas de PostGIS..... | 52 |
| 3.7.4. | Objetos SIG soportados..... | 52 |
| 3.7.5. | Aplicaciones que soportan PostGIS..... | 53 |
| 3.7.6. | Arquitectura..... | 53 |
| 3.8. | PMAPPER..... | 54 |
| 3.8.1. | Introducción..... | 54 |
| 3.8.2. | Características..... | 55 |
| 3.8.3. | Estructura de archivos..... | 55 |
| 3.8.4. | Cuadros comparativos..... | 55 |
| 3.8.5. | Relación entre clientes web..... | 56 |
| 3.9. | QUANTUM GIS..... | 57 |
| 3.9.1. | Introducción..... | 57 |
| 3.9.2. | Características..... | 57 |
| 3.9.3. | Principales plugins..... | 58 |
| 3.9.4. | GUI de Quantum GIS..... | 59 |
| 3.9.5. | Exportar mapa a MapServer..... | 59 |
| 3.10. | PLANTEAMIENTO DEL DISEÑO ARQUITECTÓNICO DEL SISTEMA..... | 60 |
| 3.11. | APORTE DE GIS AL TURISMO..... | 61 |
| 3.12. | METODOLOGÍA SCRUM..... | 62 |
| 3.12.1. | Introducción a la metodología..... | 62 |
| 3.12.2. | Gestión de la evolución del proyecto..... | 63 |
| 3.12.3. | Elementos de Scrum:..... | 63 |
| 3.12.4. | Reuniones de Scrum..... | 64 |
| 3.12.5. | Roles de Scrum..... | 65 |
| 3.12.6. | Fases de Scrum..... | 66 |
| 4. | CAPÍTULO IV: IMPLEMENTACIÓN..... | 67 |
| 4.1. | Proyecto de software..... | 67 |
| 4.2. | Asignación de roles..... | 68 |
| 4.3. | Análisis inicial de requerimientos..... | 69 |
| 4.4. | SPRINT 0: Desarrollo de módulo de visita virtual..... | 69 |

| | | |
|-------|--|-----|
| 4.5. | SPRINT 1: Desarrollo de módulo gestión de usuarios..... | 79 |
| 4.6. | SPRINT 2: Desarrollo de módulo foro..... | 87 |
| 4.7. | SPRINT 3: Desarrollo de módulo de contrataciones..... | 95 |
| 4.8. | SPRINT 4: Desarrollo de módulo de gestión de la información..... | 104 |
| 4.9. | SPRINT 5: Desarrollo de módulo GIS..... | 127 |
| 4.10. | ENTREGABLES | 142 |
| 5. | CAPÍTULO V: CONCLUSIONES, RECOMENDACIONES, E IMPACTO | 143 |
| | GLOSARIO DE TÉRMINOS | 149 |
| | BIBLIOGRAFÍA..... | 153 |
| | ANEXOS | 157 |

ÍNDICE DE TABLAS

Pág.

| | |
|--|----|
| Tabla 1: Soporte de CSS en los navegadores. | 31 |
| Tabla 2: Características generales de SGBD PostgreSQL. | 36 |
| Tabla 3: Limitaciones de PostgreSQL. | 38 |
| Tabla 4: Ventajas de PostgreSQL. | 38 |
| Tabla 5: Desventajas de PostgreSQL. | 38 |
| Tabla 6: Características de servidor de mapas MapServer. | 41 |
| Tabla 7: Directorios de proyecto Symfony. | 47 |
| Tabla 8: Directorio de aplicación Symfony. | 48 |
| Tabla 9: Directorio de módulo de Symfony. | 48 |
| Tabla 10: Evaluación por el método QSOs. | 51 |
| Tabla 11: Aplicaciones soportadas por PostGIS. | 53 |
| Tabla 12: Comparativa entre PostGIS y MySQL Spatial Extensions. | 54 |
| Tabla 13: Comparación clientes ligeros web gis – Descripción General. | 55 |
| Tabla 14: Características técnicas de clientes ligeros web SIG. | 56 |
| Tabla 15: Plugins principales de Quantum GIS 1.8. | 59 |
| Tabla 16: Definición de tecnologías y herramientas. | 60 |
| Tabla 17: Uso de tecnología GIS en el turismo. | 62 |
| Tabla 18: Roles de Scrum. | 65 |
| Tabla 19: Proyecto de software. | 68 |
| Tabla 20: Asignación de roles SCRUM. | 69 |
| Tabla 21: Módulos del sistema. | 69 |
| Tabla 22: Requerimientos tecnológicos del aplicativo. | 69 |
| Tabla 23: Historia de usuario N° 1. | 70 |
| Tabla 24: Historia de usuario N° 2. | 70 |
| Tabla 25: Pila de producto N°1. | 70 |
| Tabla 26: Pila de sprint N° 1. | 71 |
| Tabla 27: Tarjeta CRC de la clase BaseSitiosVv. | 76 |
| Tabla 28: Tarjeta CRC de la clase SitiosVvTable. | 76 |
| Tabla 29: Tarjeta CRC de las clase SitiosVv. | 77 |
| Tabla 30: Prueba de aceptación de creación de base de datos. | 78 |
| Tabla 31: Prueba de aceptación de creación de tablas de módulo VR. | 78 |
| Tabla 32: Prueba de aceptación de la implementación del módulo VR (Visita virtual). | 78 |
| Tabla 33: Historia de usuario N° 3. | 79 |
| Tabla 34: Historia de usuario N° 4. | 80 |
| Tabla 35: Pila de producto N° 2. | 80 |
| Tabla 36: Pila de sprint N° 2. | 80 |
| Tabla 37: Tarjeta CRC de la clase BaseUsuario. | 83 |
| Tabla 38: Tarjeta CRC de la clase UsuarioTable. | 83 |
| Tabla 39: Tarjeta CRC de la clase BaseContacto. | 83 |
| Tabla 40: Tarjeta CRC de la clase ContactoTable. | 84 |
| Tabla 41: Tarjeta CRC de la clase SessionTable. | 84 |
| Tabla 42: Tarjeta CRC de la clase sign_userActions. | 84 |
| Tabla 43: Prueba de aceptación de creación de tablas para módulo gestión usuarios. | 86 |
| Tabla 44: Prueba de aceptación de creación y modificación de cuentas de usuarios. | 87 |
| Tabla 45: Prueba de aceptación de autenticación de usuarios. | 87 |
| Tabla 46: Historia de usuario N° 5. | 88 |

| | |
|---|-----|
| Tabla 47: Historia de usuario N° 6. | 88 |
| Tabla 48: Pila de producto del N° 3. | 88 |
| Tabla 49: Pila de Sprint N° 3. | 89 |
| Tabla 50: Tarjeta CRC de la clase BaseForo. | 92 |
| Tabla 51: Tarjeta CRC de la clase BaseComentarios. | 92 |
| Tabla 52: Tarjeta CRC de la clase ForoTable. | 92 |
| Tabla 53: Tarjeta CRC de la clase ComentariosTable. | 92 |
| Tabla 54: Tarjeta CRC de la clase iforoActions. | 93 |
| Tabla 55: Prueba de aceptación de creación de tablas para módulo foro. | 94 |
| Tabla 56: Prueba de aceptación de autenticación de usuarios. | 94 |
| Tabla 57: Prueba de aceptación de los temas de discusión. | 94 |
| Tabla 58: Prueba de aceptación de creación de comentarios. | 94 |
| Tabla 59: Historia de usuario N° 7. | 95 |
| Tabla 60: Historia de usuario N° 8. | 95 |
| Tabla 61: Pila de producto N° 4. | 96 |
| Tabla 62: Pila de Sprint N° 4. | 96 |
| Tabla 63: Tarjeta CRC de la clase BaseCatalogo. | 99 |
| Tabla 64: Tarjeta CRC de la clase BaseCabReservaContratacion. | 99 |
| Tabla 65: Tarjeta CRC de la clase BaseDetReservaContrato. | 99 |
| Tabla 66: Tarjeta CRC de la clase CatalogoTable. | 99 |
| Tabla 67: Tarjeta CRC de la clase DetReservaContratoTable. | 99 |
| Tabla 68: Tarjeta CRC de la clase CabReservaContratacionTable. | 100 |
| Tabla 69: Tarjeta CRC de la clase contratacionesActions. | 100 |
| Tabla 70: Prueba de aceptación de creación de tablas para módulo contrataciones. | 103 |
| Tabla 71: Prueba de aceptación de disponibilidad de catálogo en línea y compra. | 104 |
| Tabla 72: Prueba de generación de informe PDF y envío por email. | 104 |
| Tabla 73: Historia de usuario N° 9. | 105 |
| Tabla 74: Historia de usuario N° 10. | 105 |
| Tabla 75: Pila de producto del módulo de gestión de la información. | 105 |
| Tabla 76: Pila de Sprint del módulo de gestión de la información. | 106 |
| Tabla 77: Creación de módulos de administración. | 108 |
| Tabla 78: Creación de módulos frontales. | 108 |
| Tabla 79: Tarjeta CRC de la clase BaseContacto. | 111 |
| Tabla 80: Tarjeta CRC de la clase BaseCatalogo. | 111 |
| Tabla 81: Tarjeta CRC de la clase BaseFtRecursoMultimedia. | 111 |
| Tabla 82: Tarjeta CRC de la clase BaseFtMenu. | 111 |
| Tabla 83: Tarjeta CRC de la clase BaseFtModulo. | 111 |
| Tabla 84: Tarjeta CRC de la clase BaseFtArticulo. | 112 |
| Tabla 85: Tarjeta CRC de la clase BaseFtConfig. | 112 |
| Tabla 86: Tarjeta CRC de la clase BaseFtComentario. | 112 |
| Tabla 87: Tarjeta CRC de la clase BaseCalendarioEvento. | 112 |
| Tabla 88: Tarjeta CRC de la clase ContactoTable. | 112 |
| Tabla 89: Tarjeta CRC de la clase EmpresaTuristicaTable. | 112 |
| Tabla 90: Tarjeta CRC de la clase FtRecursoMultimediaTable. | 113 |
| Tabla 91: Tarjeta CRC de la clase FtMenuTable. | 113 |
| Tabla 92: Tarjeta CRC de la clase FtModuloTable. | 113 |
| Tabla 93: Tarjeta CRC de la clase FtArticuloTable. | 113 |
| Tabla 94: Tarjeta CRC de la clase FtConfigTable. | 113 |
| Tabla 95: Tarjeta CRC de la clase FtComentarioTable. | 114 |
| Tabla 96: Tarjeta CRC de la clase CalendarioEventoTable. | 114 |
| Tabla 97: Tarjeta CRC de la clase busquedaActions. | 114 |
| Tabla 98: Tarjeta CRC de la clase contactenosActions. | 114 |

| | |
|--|-----|
| Tabla 99: Tarjeta CRC de la clase errorActions | 114 |
| Tabla 100: Tarjeta CRC de la clase eventoActions | 115 |
| Tabla 101: Tarjeta CRC de la clase galeriaActions | 115 |
| Tabla 102: Tarjeta CRC de la clase inicioActions | 115 |
| Tabla 103: Tarjeta CRC de las clases para gestión de artículos | 115 |
| Tabla 104: Tarjeta CRC de las clases mapaSitioActions y noticiasActions | 115 |
| Tabla 105: Tarjeta CRC de la clase BasesfGuardAuthActions | 117 |
| Tabla 106: Tarjeta CRC de la clase BasesfGuardUser | 117 |
| Tabla 107: Creación de módulos frontales | 122 |
| Tabla 108: Tarjeta CRC de la clase addcatalogoActions | 122 |
| Tabla 109: Tarjeta CRC de la clase busquedaActions | 122 |
| Tabla 110: Tarjeta CRC de la clase contratacionesActions | 122 |
| Tabla 111: Tarjeta CRC de la clase eventoActions | 122 |
| Tabla 112: Tarjeta CRC de la clase registroActions | 123 |
| Tabla 113: Tarjeta CRC de la clase reportActions | 123 |
| Tabla 114: Tarjeta CRC de la clase uploadActions | 123 |
| Tabla 115: Prueba de aceptación de creación de tablas para módulo gestión de información | 126 |
| Tabla 116: Prueba de aceptación de integración de módulos e ingreso a backend | 126 |
| Tabla 117: Historia de usuario N° 11 | 127 |
| Tabla 118: Historia de usuario N° 12 | 128 |
| Tabla 119: Pila de producto 6 | 128 |
| Tabla 120: Pila de Sprint 6 | 129 |
| Tabla 121: Prueba de aceptación de estudio de herramientas GIS de software libre | 141 |
| Tabla 122: Prueba de aceptación de visualización de sitios turísticos en GIS web | 141 |

ÍNDICE DE GRÁFICOS

| | Pág. |
|--|------|
| Gráfica 1: Entrada de extranjeros al Ecuador | 5 |
| Gráfica 2: Visitantes extranjeros por mes. | 6 |
| Gráfica 3: Arquitectura del sistema. | 9 |
| Gráfica 4: Arquitectura funcional del sistema. | 10 |
| Gráfica 5: Interfaz gráfica de usuario de PgAdmin 3..... | 12 |
| Gráfica 6: Componentes básicos de un SIG. | 14 |
| Gráfica 7: Tipos de Proyección | 14 |
| Gráfica 8: Modelo ráster. | 16 |
| Gráfica 9: Modelo vectorial SIG..... | 16 |
| Gráfica 10: Patrón MVC de Symfony. | 24 |
| Gráfica 11: Tecnologías agrupadas bajo el concepto AJAX..... | 32 |
| Gráfica 12: Componentes importantes de SGBD PostgreSQL. | 37 |
| Gráfica 13: Arquitectura de MapServer..... | 39 |
| Gráfica 14: Funcionamiento de MapServer. | 40 |
| Gráfica 15: Características de IDE Netbeans 7.0.1. | 43 |
| Gráfica 16: Integración de Symfony con Netbeans..... | 44 |
| Gráfica 17: Componentes de aplicación básica en Symfony..... | 45 |
| Gráfica 18: Flujo de trabajo de Symfony..... | 46 |
| Gráfica 19: Organización del código en Symfony..... | 47 |
| Gráfica 20: Jerarquía de tipo de datos espaciales. | 53 |
| Gráfica 21: Arquitectura de PostGIS | 54 |
| Gráfica 22: Relación entre clientes de servicios web..... | 56 |
| Gráfica 23: GUI de Quantum GIS | 59 |
| Gráfica 24: Diseño arquitectónico del sistema..... | 61 |
| Gráfica 25: Proceso metodología SCRUM. | 62 |
| Gráfica 26: Marco Scrum estándar. | 63 |
| Gráfica 27: Diagrama de ciclo iterativo de SCRUM. | 63 |
| Gráfica 28: Diagrama entidad - relación de la tabla del módulo VR..... | 72 |
| Gráfica 29: Configurar PHP en Netbeans..... | 73 |
| Gráfica 30: Configurar Symfony en Netbeans. | 74 |
| Gráfica 31: Crear nuevo proyecto web PHP con Symfony. | 74 |
| Gráfica 32: Estructura de directorio de nuevo proyecto. | 74 |
| Gráfica 33: Creación de aplicación frontend - sis_itur. | 74 |
| Gráfica 34: Creación de módulo visitavirtual..... | 75 |
| Gráfica 35: Interfaz de ingreso a módulo VR. | 75 |
| Gráfica 36: Pantalla de lista de visitas. | 76 |
| Gráfica 37: Pantalla de visita virtual. | 76 |
| Gráfica 38: Componentes interfaz de la visita virtual..... | 77 |
| Gráfica 39: Prueba funcional del módulo VR. | 78 |
| Gráfica 40: Diagrama entidad relación de tablas ft_modulo, ft_menu..... | 79 |
| Gráfica 41: Diagrama entidad - relación de tablas del módulo de gestión de usuarios..... | 81 |
| Gráfica 42: Creación de módulo sign_user. | 81 |
| Gráfica 43: Ventana usuario – para autenticarse. | 82 |
| Gráfica 44: Ventana de creación de cuentas de usuarios..... | 82 |
| Gráfica 45: Ventana de perfil de usuario. | 82 |
| Gráfica 46: Ventana editar perfil usuario..... | 83 |
| Gráfica 47: Prueba de funcionalidad de módulo de gestión de usuarios..... | 84 |

| | |
|--|-----|
| Gráfica 48: Campos inválidos en ventana creación de cuentas de usuarios. | 85 |
| Gráfica 49: Creación correcta de nuevo usuario en ventana Perfil. | 85 |
| Gráfica 50: Actualización errónea en ventana editar perfil usuario. | 85 |
| Gráfica 51: Actualización correcta en ventana editar perfil usuario. | 86 |
| Gráfica 52: Autenticación errónea y correcta de usuarios. | 86 |
| Gráfica 53: Integración de módulo gestión de usuarios al sistema. | 87 |
| Gráfica 54: Diagrama entidad – relación de tablas para módulo foro. | 89 |
| Gráfica 55: Creación de módulo foro. | 90 |
| Gráfica 56: Pantalla de bienvenida de foro | 90 |
| Gráfica 57: Pantalla de temas publicados. | 90 |
| Gráfica 58: Pantalla de creación de nuevo tema. | 91 |
| Gráfica 59: Pantalla de edición de nuevo tema. | 91 |
| Gráfica 60: Pantalla de citar tema. | 91 |
| Gráfica 61: Pantalla mis temas. | 91 |
| Gráfica 62: Pantalla de búsqueda de temas. | 92 |
| Gráfica 63: Prueba de ingreso al módulo foro. | 93 |
| Gráfica 64: Prueba de creación y actualización de tema de discusión. | 93 |
| Gráfica 65: Prueba de creación de comentario. | 94 |
| Gráfica 66: Integración de módulo foro al sistema. | 95 |
| Gráfica 67: Diagrama entidad - relación de tablas para módulo contrataciones. | 96 |
| Gráfica 68: Creación de módulo de contrataciones. | 97 |
| Gráfica 69: Pantalla de inicio de módulo contrataciones | 97 |
| Gráfica 70: Pantalla de catálogo de productos – servicios. | 97 |
| Gráfica 71: Pantalla de detalle de pedido. | 98 |
| Gráfica 72: Pantalla de forma de pago. | 98 |
| Gráfica 73: Pantalla mis pedidos. | 98 |
| Gráfica 74: Ver información de elemento de catálogo. | 101 |
| Gráfica 75: Prueba agregar elemento a carrito compras. | 101 |
| Gráfica 76: Prueba de validación de datos en pedido. | 101 |
| Gráfica 77: Prueba “quitar” elemento de pedido. | 102 |
| Gráfica 78: Prueba de cálculo de costos. | 102 |
| Gráfica 79: Prueba "cancelar pedido". | 102 |
| Gráfica 80: Prueba forma de pago, “elegir y finalizar”. | 102 |
| Gráfica 81: Prueba compra correcta. | 103 |
| Gráfica 82: Prueba envío de informe por email. | 103 |
| Gráfica 83: Integración del módulo de contrataciones. | 104 |
| Gráfica 84: Diagrama entidad - relación de tablas para módulo de gestión de la información. | 107 |
| Gráfica 85: Creación de aplicación administrativa. | 108 |
| Gráfica 86: Pantalla frontal de inicio. | 109 |
| Gráfica 87: Pantalla turismo de aventura. | 109 |
| Gráfica 88: Pantalla de galería de imágenes. | 109 |
| Gráfica 89: Pantalla ver artículo. | 110 |
| Gráfica 90: Pantalla contáctenos. | 110 |
| Gráfica 91: Pantalla de búsqueda. | 110 |
| Gráfica 92: Pantalla de eventos. | 110 |
| Gráfica 93: Instalación de sfDoctrineGuardPlugin | 116 |
| Gráfica 94: Diagrama entidad–relación para el plugin sfDoctrineGuardPlugin. | 116 |
| Gráfica 95: Pantalla de autenticación de aplicación administrativa. | 116 |
| Gráfica 96: Pantalla de inicio de aplicación administrativa. | 117 |
| Gráfica 97: Pantalla de módulo de gestión de usuarios. | 117 |
| Gráfica 98: Pantalla de gestión de módulo foro. | 118 |
| Gráfica 99: Pantalla de gestión de productos y servicios de módulo contrataciones. | 118 |

| | |
|--|-----|
| Gráfica 100: Pantalla de gestión de contratos de módulo contrataciones..... | 118 |
| Gráfica 101: Pantalla de gestión de módulo visita virtual..... | 118 |
| Gráfica 102: Pantalla de creación de visitas de módulo visita virtual..... | 119 |
| Gráfica 103: Pantalla gestión de artículos del módulo gestión de información..... | 119 |
| Gráfica 104: Pantalla gestión de recursos del módulo gestión de información. | 119 |
| Gráfica 105: Pantalla creación de recursos del módulo gestión de información. | 120 |
| Gráfica 106: Pantalla gestión de empresas turísticas..... | 120 |
| Gráfica 107: Pantalla creación de eventos. | 120 |
| Gráfica 108: Pantalla gestión de reportes..... | 121 |
| Gráfica 109: Pantalla de resultado de búsqueda..... | 121 |
| Gráfica 110: Pantalla configuraciones extras..... | 121 |
| Gráfica 111: Prueba ingreso a aplicación administrativa. | 123 |
| Gráfica 112: Prueba de creación de información. | 124 |
| Gráfica 113: Prueba de edición errónea de información. | 124 |
| Gráfica 114: Prueba de edición correcta de información. | 124 |
| Gráfica 115: Prueba de eliminación de información..... | 125 |
| Gráfica 116: Prueba de creación de visitas virtuales..... | 125 |
| Gráfica 117: Prueba de visualización correcta de artículos. | 125 |
| Gráfica 118: Prueba de creación de comentarios. | 125 |
| Gráfica 119: Integración de módulos en aplicación frontal. | 126 |
| Gráfica 120: Integración de módulos en aplicación backend. | 127 |
| Gráfica 121: MapServer en windows..... | 131 |
| Gráfica 122: MapServer en linux..... | 131 |
| Gráfica 123: Carga de archivos .shp en PostGIS..... | 132 |
| Gráfica 124: Importar archivo .shp a base de datos PostGIS. | 132 |
| Gráfica 125: Importación correcta de datos. | 133 |
| Gráfica 126: Crear conexión a base de datos PostGIS..... | 133 |
| Gráfica 127: Agregar capa PostGIS a Quantum GIS..... | 134 |
| Gráfica 128: Visualizar capa PostGIS. | 134 |
| Gráfica 129: Añadir capa desde archivo .shp | 135 |
| Gráfica 130: Exportar mapa a archivo .map..... | 135 |
| Gráfica 131: Estructura de archivo .map..... | 136 |
| Gráfica 132: Integración de nuevo mapa a Pmapper | 136 |
| Gráfica 133: Visor de mapas sin configurar. | 136 |
| Gráfica 134: Visor de mapas completo. | 139 |
| Gráfica 135: Prueba de funcionalidad de visualización de capas..... | 139 |
| Gráfica 136: Prueba de funcionalidad de selección de objetos del mapa. | 140 |
| Gráfica 137: Prueba de funcionalidad de búsqueda de sitios en el mapa..... | 140 |
| Gráfica 138: Prueba de funcionalidad de impresión de mapas..... | 140 |
| Gráfica 139: Integración de módulo GIS en el sistema. | 141 |
| Gráfica 140: Costos de desarrollo del proyecto..... | 144 |
| Gráfica 141: Ingresos económicos por actividad turística..... | 144 |
| Gráfica 142: Promoción y difusión de la actividad turística..... | 145 |

CAPÍTULO I: INTRODUCCIÓN

1.1. LA COMUNIDAD DE CARIACU Y EL TURISMO COMUNITARIO.

1.1.1. Antecedentes de la comunidad.

La comunidad de Cariacu está ubicada al noreste de la Parroquia Ayora, en la provincia de Pichincha, cantón Cayambe, se encuentra a 7 kilómetros aproximadamente de la ciudad de Cayambe. Con una población aproximada de 750 habitantes. Geográficamente se encuentra de 2.800 a 3.600 msnm²; el clima, es frío con un promedio de 15° C.

La población de la comunidad se dedica a la agricultura y la ganadería, utilizando para la primera actividad 140 ha³ aproximadamente. Un gran porcentaje de los productos los emplean para el consumo y el resto lo venden, de la venta de estos obtienen muy bajos costos, lo que provoca que la población opte por trabajar en las florícolas, quehaceres domésticos y la construcción.

1.1.2. Turismo comunitario en la comunidad.

Desde hace un tiempo atrás la comunidad ha visto que el turismo puede ser una alternativa para generar ingresos y crear empleos para sus moradores, evitando así la migración.

La iniciativa para incursionar en esta actividad provino de la Dirección Cantonal de Turismo de Cayambe y de la Universidad de Especialidades Turísticas (UCT), las que han permitido realizar un estudio, análisis, diagnóstico y evaluación de los componentes del sistema turístico, para de esta forma determinar micro-proyectos y realizar una planificación estratégica con propuestas óptimas para el desarrollo de este ámbito.

El Turismo Comunitario significa turismo responsable, respeto al medio ambiente y a la Comunidad huésped, nace como otra fuente económica de las comunidades campesinas, indígenas, y otros, que generan ingresos extras a las actividades económicas diarias.

El turismo es uno de los ejes sobre los cuales la comunidad trabaja debido a que este tipo de actividad es vistosa para los turistas ya que esto involucra la convivencia, la gastronomía, actividades culturales, agrícolas, ganaderas entre otras. Además se

² Msnm: Metros sobre el nivel del mar.

³ ha: símbolo de hectárea.

caracteriza porque una parte importante de los beneficios son de control de los pobladores de la comunidad.

1.2. EL PROBLEMA.

1.2.1. Antecedentes.

En los últimos tiempos el turismo ha ido evolucionando hasta llegar a ser un medio de ingresos monetarios muy representativo para las comunidades que lo impulsan, ya que para su desarrollo no es necesario disponer de grandes empresas que posean maquinarias que contaminan, ni tampoco de infraestructura costosa, nada más se necesita tener la predisposición de conservar el ecosistema y de explotar los recursos naturales de forma sustentable; la difusión de estos sitios, riquezas naturales, cultura, gastronomía y entretenimiento es ineficaz ya que no son promocionados como se debería, esto solo trae pérdidas económicas y el mal uso de los recursos que se encuentran en el entorno. La actividad turística ha ido creciendo paulatinamente en todo el mundo y en especial en el Ecuador en sus comunidades.

1.2.2. Situación actual del turismo comunitario.

Hoy en día las comunidades indígenas del Cantón Cayambe han visto una buena fuente de ingresos a todo lo referente a la actividad turística, y siendo estas localidades ricas en recursos naturales, culturales, costumbres, gastronomía, entretenimiento se han convertido en potenciales destinos turísticos que tanto turistas nacionales como internacionales han puesto su mirada en ellos, pero hace falta un medio de difusión tanto a nivel local como internacional ya que por el momento solo se promociona de boca en boca y por referencias de personas que ya los visitaron.

La comunidad de Cariacu desde hace unos años atrás ha incursionado en este tipo de actividad. El grupo de ecoturismo “Pakchapi Pakari” conjuntamente con la Dirección de Turismo de Cayambe y las autoridades del gobierno comunitario trabajan por el desarrollo del turismo en la comunidad.

1.2.3. Planteamiento del problema.

Las comunidades indígenas del cantón Cayambe que hacen del turismo comunitario su medio de vida y en especial la comunidad de Cariacu no cuentan con medios eficientes para gestionar la publicidad de su actividad turística, riquezas naturales, cultura, gastronomía, costumbres y entretenimiento tanto a nivel local como internacional, a pesar

de los esfuerzos de las autoridades y del grupo de ecoturismo esta actividad no se ha desarrollado como se esperaba, por lo cual se ve la necesidad de plantear un medio de difusión masivo con la utilización de las nuevas tecnologías informáticas.

1.3. OBJETIVOS.

1.3.1. Objetivo general.

Implementar una aplicación web dinámica para la gestión y promoción de las riquezas turísticas de las comunidades indígenas del cantón Cayambe, en especial de la comunidad de Cariacu.

1.3.2. Objetivos específicos.

- Buscar y recopilar información de los sitios turísticos.
- Investigar las diferentes tecnologías que permiten desarrollar la aplicación.
- Desarrollar e Implantar la aplicación web de gestión turística.
- Incorporar un GIS⁴ dentro del sitio web con información de los principales sitios turísticos existentes en la zona.

1.4. ALCANCE.

El aplicativo desarrollado se encontrará publicado en la web lo cual permitirá brindar toda la información de los recursos turísticos ofertados en cualquier parte del mundo a toda persona con acceso a internet y con deseos de vivir una experiencia nueva.

El sistema tiene como finalidad permitir gestionar y promocionar eficientemente las riquezas turísticas de la comunidad de Cariacu lo cual beneficiara de forma directa e indirectamente a:

- Grupos organizados de mujeres.
- Grupos artísticos de la localidad.
- Grupo de Ecoturismo
- Grupo de jóvenes.
- Pequeños y medianos empresarios.
- A la comunidad en general.
- De alguna manera a los turistas interesados en vivir una experiencia inolvidable.

⁴ GIS: También conocida con SIG, es la abreviatura de Sistemas de Información Geográfica.

Para el buen desempeño y funcionalidad de la aplicación, ésta contará de los siguientes módulos:

- Módulo de Gestión de Usuarios y seguridades.
- Módulo de Contrataciones/Reservaciones.
- Módulo VR (Visitas Virtuales).
- Módulo GIS para localización de los principales sitios turísticos de la zona.
- Módulo de Foros.
- Módulo de Gestión de Información y publicidad turística.

1.5. JUSTIFICACIÓN.

En la actualidad la conservación del medio ambiente es el eje primordial de los seres humanos, debido a esto las comunidades indígenas que son más conscientes de la preservación del entorno donde viven han hecho un gran esfuerzo para optar por la industria sin chimeneas o más conocida como turismo comunitario, ya que es una fuente sustentable de ingresos económicos.

La comunidad de Cariacu está ubicada al Noreste del Cantón Cayambe en la Provincia de Pichincha tiene una población de 750 habitantes en el último censo, esta es una zona dedicada a la agricultura, ganadería, apicultura, que poco a poco busca el reconocimiento local, hoy en día esta comunidad ha visto adecuada la explotación sustentable de sus recursos naturales y culturales a través del turismo comunitario, pero la difusión de esta actividad es ineficiente debido a que la información necesaria no se encuentra disponible para todas aquellas personas que se interesen en visitarlos, lo cual lleva a un esfuerzo casi vano en impulsar esta industria en la localidad.

Para dar un mayor impulso a esta actividad es necesario implementar una aplicación informática que permita gestionar y promocionar tales sitios, en la cual los potenciales turistas puedan visualizar a través de la web toda la información requerida como: mapas, fotografías, videos, costos, datos geográficos, paquetes promocionales, y otros, de todos los puntos turísticos de la zona, y además puedan contactar a las comunidades para posibles contrataciones de los servicios.

Con la implementación de la aplicación mencionada se dará solución al desconocimiento y falta de información que tienen los turistas, lo cual impulsará el crecimiento económico de la población. En el desarrollo de esta aplicación se utilizarán herramientas de software libre que pondrán a prueba los conocimientos académicos adquiridos, las herramientas a utilizar son:

- Base de datos PostgreSQL y versión espacial PostGIS.
- IDE Netbeans 7.0.1 (Java). PHP⁵ y Framework Symfony.
- Servidor de Mapas MapServer.
- Cliente liviano para visualización de información geográfica – pmapper.
- Cliente Pesado para administrar la Base de datos GIS – Quantum GIS.

Se ha escogido estas herramientas y lenguajes de programación debido a la escalabilidad y compatibilidad con la variedad de entornos y plataformas de desarrollo, además por la arquitectura que éstas presentan, la suficiente documentación y por sobre todo por los conocimientos que se tiene sobre estas. Para el desarrollo de la aplicación se utilizará la metodología de desarrollo ágil SCRUM para asegurar la producción de un software de alta calidad que satisfaga la necesidad del usuario final.

Para desarrollo web, la solución más común para la organización de su código es el **patrón de diseño MVC**⁶. El patrón de diseño MVC define una manera de organizar el código de acuerdo a su naturaleza. Este patrón separa el código en **tres capas**: Modelo, Vista y Controlador.

Además él (Ministerio de Turismo, 2008) informó que: “el país ha sido elegido por la exclusiva red de agencias de viajes Virtuoso⁷ como uno de los mejores destinos de turismo de lujo en el 2013”.

De acuerdo a varios asesores de viajes de Virtuoso, las ofertas de turismo del Ecuador colocan al país entre los 20 destinos más populares del mundo. Muestra de ello el ministerio de turismo indica los siguientes datos estadísticos del incremento de entradas de los extranjeros al país.



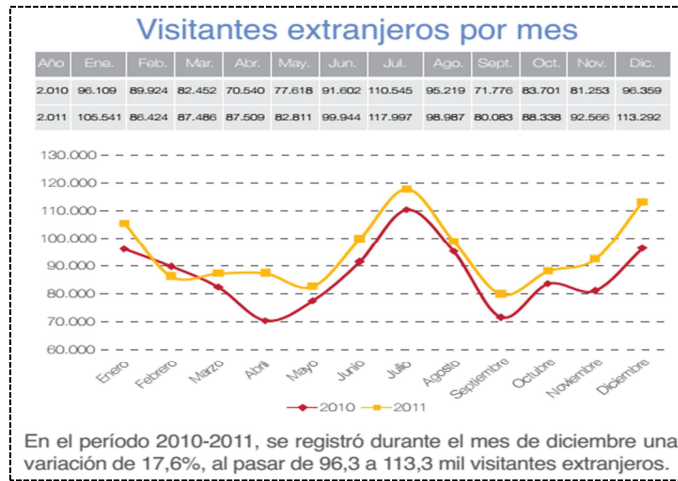
Gráfica 1: Entrada de extranjeros al Ecuador

Fuente: *La experiencia turística en el Ecuador*, Recuperado de: http://servicios.turismo.gob.ec/images/estadisticas/Minis_Turismo_folleto_espanol_completo.pdf, 2013

⁵ **PHP** (PHP Hypertext Pre-processor), lenguaje de programación web.

⁶ **MVC:** patrón de arquitectura de software Modelo, Vista, Controlador.

⁷ **Virtuoso:** es una red exclusiva, líder en la industria de viajes.



Gráfica 2: Visitantes extranjeros por mes.

Fuente: *La experiencia turística en el Ecuador*, Recuperado de: http://servicios.turismo.gob.ec/images/estadisticas/Minis_Turismo_folleto_espanol_completo.pdf, 2013

CAPÍTULO II: MARCO TEÓRICO

2.1. TEORÍA DEL DESARROLLO DEL SISTEMA.

Las Metodologías de Desarrollo de Software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad. Las metodologías se basan en una combinación de los modelos de proceso genéricos ya sean cascada, evolutivo, incremental, espiral entre otros. Además una metodología debe definir con precisión los artefactos, roles y actividades, junto con prácticas y técnicas recomendadas.

2.1.1. Metodología ágil

Un proceso es ágil cuando el desarrollo de software es incremental, es decir, entregas pequeñas de software en ciclos rápidos, se trabaja en cooperación entre el cliente y los desarrolladores, además es sencillo de aprender, modificar, y es adaptable. Entre las metodologías ágiles identificadas encontramos: XP⁸, Scrum, Familia de Metodologías Crystal⁹, FDD¹⁰, DSDM¹¹, ASD¹².

2.1.2. Scrum



Según (Albaladejo), Scrum es: “un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto”.

Según (Díaz, 2009) , define a Scrum como: “una colección de procesos pensada para la gestión de proyectos que permite centrarse en la entrega de valor al cliente y la potenciación del equipo para lograr su máxima eficiencia, dentro de un esquema de mejora continua”. (pp. 41 - 42)

2.1.3. Marco estándar de Scrum.

Una implantación de la metodología estándar de Scrum incluye 3 aspectos, los cuales se lista a continuación:

⁸ **XP (eXtreme Programming)**: Programación extrema, es una metodología ágil de desarrollo de software.

⁹ **Familia de metodologías Crystal**: metodologías ágiles, cada una de ellas está adecuada para un tipo de proyecto.

¹⁰ **FDD (Feature Driven Development)**: es una metodología ágil basada en las funcionalidades.

¹¹ **DSDM (Dynamic Systems Development Method)**: es una metodología ágil para desarrollo de sistemas dinámicos.

¹² **ASD (Adaptive Software Development)**: modelo de implementación de patrones ágiles para desarrollo de software

- **Elementos:** pila del producto, pila del sprint¹³ e incremento.
- **Reuniones:** de inicio del sprint, revisión diaria, revisión del sprint y retrospectiva.
- **Roles:** propietario del producto, equipo, Scrum master y resto de implicados.

2.1.4. Elementos de Scrum

- ✓ **Pila del producto:** (product backlog) lista de requisitos de usuario que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.
- ✓ **Pila del sprint:** (sprint backlog) lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.
- ✓ **Sprint:** nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso por tiempos prefijados.
- ✓ **Incremento:** resultado de cada sprint.

2.1.5. Reuniones de Scrum.

- ✓ **Planificación del sprint:** espacio de tiempo de trabajo precedente al inicio de cada sprint en el que se establece cuál será el trabajo y las metas que se van a alcanzar en la iteración.
- ✓ **Revisión diaria:** revisión diaria muy breve.
- ✓ **Revisión del sprint:** análisis y revisión del incremento desarrollado.
- ✓ **Retrospectiva del sprint:** revisión de lo ocurrido en el Sprint.

2.1.6. Roles de Scrum.

Los roles que intervienen en el proceso de desarrollo de un proyecto con SCRUM son los siguientes:

- ✓ **Propietario del producto:** Persona representante de todos los implicados en el resultado final del producto, es el responsable de definir los objetivos del producto.
- ✓ **Equipo de desarrollo:** grupo de trabajo que implementa el producto.
- ✓ **Scrum Master:** Líder del equipo, vela que todos los integrantes del proyecto cumplan con las reglas y proceso de Scrum.

2.2. ARQUITECTURA DEL SISTEMA.

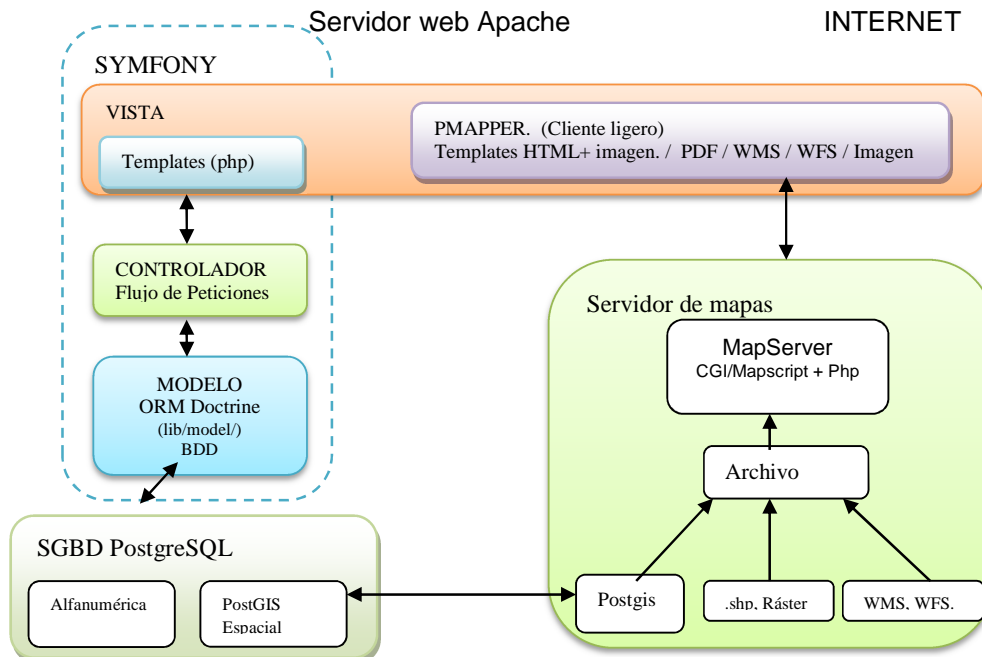
El sistema desarrollado será un aplicativo web que permita acceder a los visitantes a su parte frontal a través de un navegador web por medio del Internet, y su lógica está enfocada en el patrón de diseño MVC, integrada por:

¹³ **Sprint:** en desarrollo ágil de software es un período de tiempo durante el cual un trabajo debe ser completado y preparado para su revisión.

- ✓ La capa **Modelo**, especifica la lógica de negocio. En esta capa se hace uso del ORM¹⁴ Doctrine incluido en Symfony 1.4, que además guarda todas las clases y archivos relacionados con el modelo en el directorio lib/model/. La base de datos pertenece a esta capa
- ✓ La **Vista** es con lo que el usuario interactúa, la parte visual. Para lo cual se hace uso de las plantillas PHP de Symfony. Estas son guardadas en varios directorios templates/ de todos los módulos. Además en el módulo GIS se utilizará el cliente liviano Pmapper para la visualización de los mapas.
- ✓ El **Controlador** es la pieza de código que llama al Modelo para obtener los datos que le pasa a la Vista para la presentación al cliente. Cabe recalcar que será el framework Symfony quien dote de esta funcionalidad.

En relación a los datos, el sistema se alimentará con datos alfanuméricos que se los obtendrá de la base de datos PostgreSQL y de PostGIS para la parte del módulo GIS, además por dicho módulo se implementa otra capa la cual tendrá el servidor de mapas.

A continuación se ilustra la arquitectura del sistema, separando de acuerdo a la funcionalidad en tres niveles: modelo, vista y controlador. Las capas muestran la manera como el aplicativo es segmentado de acuerdo a la estructura del framework de desarrollo.



Gráfica 3: Arquitectura del sistema.

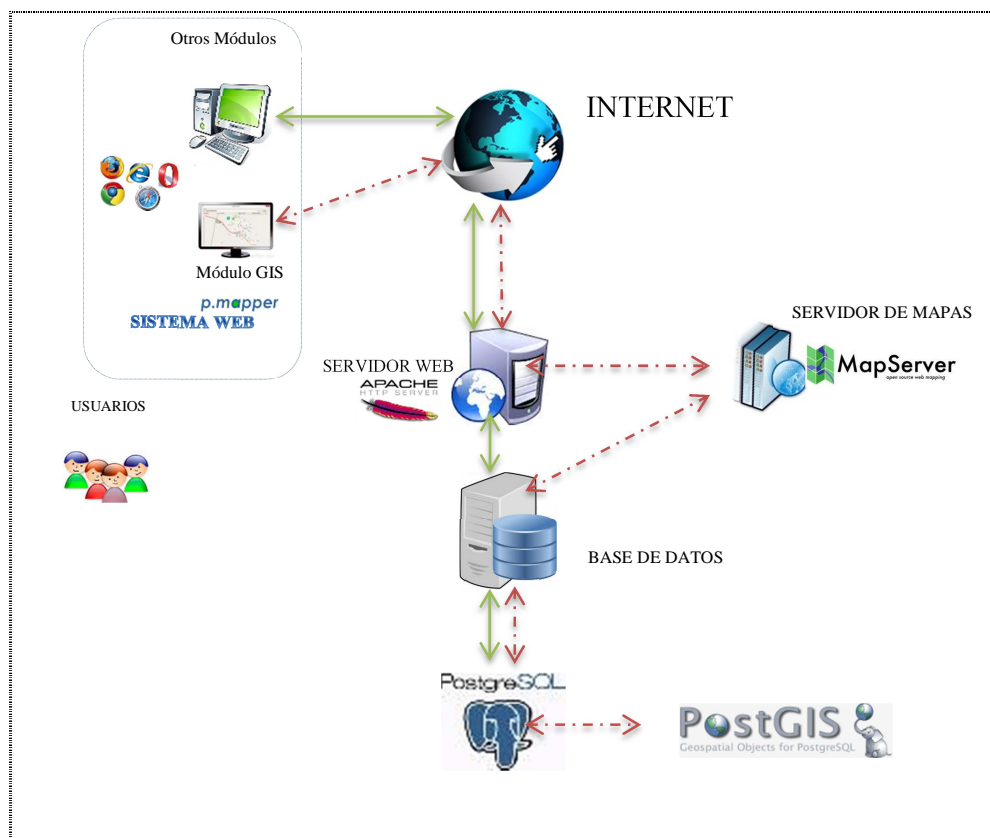
Fuente: Propia.

¹⁴ **ORM:** Es un Mapeador Objeto Relacional que proporciona una capa de persistencia.

El aplicativo, será desarrollado sobre la plataforma Linux, el servidor de aplicaciones Http¹⁵ será Apache sobre el cual correrá el aplicativo y además servirá de plataforma para el Servidor de mapas MapServer, los datos los almacenará PostgreSQL y PostGIS para los datos espaciales.

El flujo del funcionamiento del Sistema se divide en dos partes; la primera responde a las peticiones de los usuarios dentro del módulo GIS, el cual se centra en la publicación y consumo de los servicios de mapas, para lo cual se consultarán los datos espaciales desde las bases de datos alfanuméricas espaciales para luego su generación, y posterior publicación, la segunda parte responde a las solicitudes recibidas del resto de módulos del sistema para los cuales se obtienen información desde la base de datos PostgreSQL para su publicación.

Todo esto se puede utilizar a través de los navegadores como: Firefox, Google Chrome, Internet Explorer, Safari, Opera.



Gráfica 4: Arquitectura funcional del sistema.

Fuente: Propia.

¹⁵ **HTTP** (Hypertext Transfer Protocol), Protocolo de transferencia de Hipertexto.

2.3. SERVIDOR DE APLICACIONES WEB.

2.3.1. Servidor web

El servidor web o también llamado servidor HTTP es un programa que ejecuta toda aplicación al lado del servidor con conexiones bidireccionales y/o unidireccionales. El servidor web permanece en espera de peticiones de ejecución que le hará un cliente o un usuario de Internet, a las cuales le contesta de forma adecuada, entregando información de todo tipo de acuerdo a los comandos requeridos.

2.3.2. Apache

APACHE
HTTP SERVER



El servidor HTTP Apache, es un servidor web de código abierto, para plataformas Unix¹⁶, Microsoft Windows, Macintosh¹⁷ y otras, que implementa el protocolo HTTP/1.1. Es imprescindible indicar que Apache es uno de los mejores y el más utilizado entre los servidores web que existen debido a su gran estabilidad, confiabilidad y al aporte de los voluntarios que lo desarrollan.

2.4. SISTEMA GESTOR DE BASE DE DATOS (SGBD).

Entre los sistemas gestores de bases de datos existentes en la actualidad, PostgreSQL tiene un papel muy importante ya que es un sistema que tiene muchas cualidades que lo hacen ser una muy buena alternativa para instalar sistemas en empresas, universidades.

2.4.1. Definición.

PostgreSQL



PostgreSQL es un sistema gestor de bases de datos relacionales basado en Open Source¹⁸. Es decir que el código fuente del programa está al alcance de cualquier persona, lo cual posibilita a cualquier desarrollador a colaborar con el desarrollo del proyecto.

2.4.2. Herramientas de administración.

La manipulación de los datos de PostgreSQL por consola puede ser algo complicado, por tal motivo existen herramientas que facilitan este trabajo, entre algunas herramientas se encuentran: PgAdmin 3, PgAccess, PhpPgAdmin.

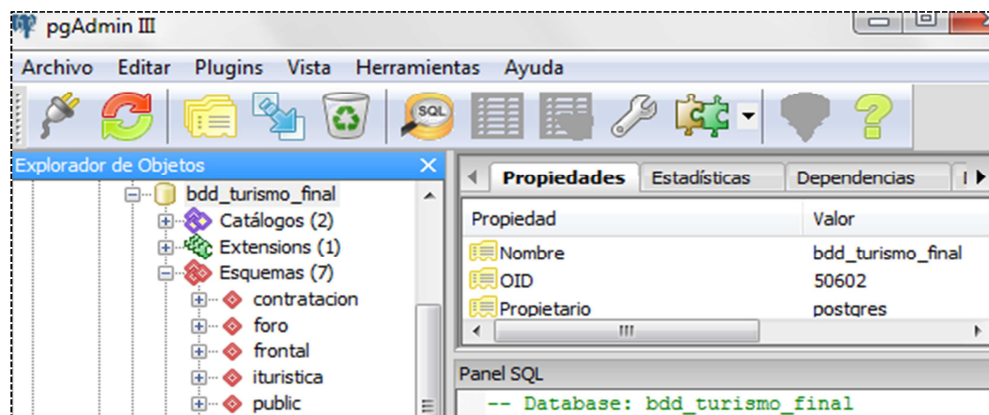
¹⁶ UNIX sistema operativo portable, multitarea y multiusuario

¹⁷ Macintosh, ordenador personal, construido y vendido por la compañía Apple.

¹⁸ Open Source: Código abierto, software distribuido y desarrollado libremente.

2.4.3. PgAdmin 3

Es una interfaz comprensible para el diseño y administración de la base de datos PostgreSQL, diseñada para ejecutarse en la mayoría de los Sistemas Operativos. La aplicación corre bajo GNU/Linux¹⁹, FreeBSD²⁰ y Windows. La interfaz gráfica soporta todas las características de PostgreSQL y facilita su administración.



Gráfica 5: Interfaz gráfica de usuario de PgAdmin 3

Fuente: Propia.

2.5. BASE DE DATOS ESPACIAL.

Las bases de datos espaciales son aquellas que procesan datos existentes en un espacio o datos espaciales, es decir, los datos relacionados a los espacios en el mundo físico.

2.5.1. PostGIS.



PostGIS es una extensión de soporte de datos espaciales para la base de datos objeto relacional PostgreSQL. Al tener soporte a datos espaciales le permite ser utilizado como una base de datos espacial de los GIS y aplicaciones de mapeo web.

2.6. SISTEMAS DE INFORMACIÓN GEOGRÁFICA.

2.6.1. Antecedentes

El inicio de los GIS es en los años 60, pero relacionados con diversos campos de investigación. En los años 80, se usó más frecuentemente gracias al desarrollo de tecnologías informáticas adecuadas, que promovieron la aparición de los productos GIS.

¹⁹ GNU/Linux, términos empleado para referirse a la combinación del núcleo o kernel libre.

²⁰ FreeBSD, avanzado sistema operativo para arquitecturas x86.

En la actualidad, se están realizando grandes inversiones en el desarrollo de herramientas para el tratamiento de información geográfica.

2.6.2. Definición.

Un Sistema de Información Geográfica o GIS es un sistema de información empleado para ingresar, almacenar, recuperar, manipular, analizar y obtener datos referenciados geográficamente o datos geoespaciales, con la finalidad de ofrecer ayuda en la toma de decisiones sobre planificación y manejo del suelo, recursos naturales, medio ambiente, transporte y otros.

2.6.3. Funcionalidad de los SIG.

Existen varios argumentos básicos para la utilización de un SIG, estos son:

- Permite realizar comparaciones entre perspectivas y escalas emulando la capacidad de representación de diferentes lugares al mismo tiempo.
- Permite diferenciar entre cambios cualitativos y cuantitativos.
- Permite tratar gran cantidad de información a varias escalas y proyecciones.
- Integra espacialmente datos tabulares y geográficos con cálculos sobre variables.
- Pone a disposición varias herramientas informáticas estandarizadas.

2.6.4. Componentes.

Los GIS están formados por subsistemas, cada uno de ellos con funciones particulares. A continuación se citan a tres subsistemas fundamentales:

- ✓ **Subsistema de datos.** Encargado de las operaciones de entrada y salida de datos, y la gestión de estos en el GIS. Permite a los otros subsistemas acceder a los datos y ejecutar sus funciones en base a estos.
- ✓ **Subsistema de visualización y creación cartográfica.** Encargada de crear representaciones a partir de los datos. Además agrega la capacidad de edición.
- ✓ **Subsistema de análisis.** Contiene métodos y procesos para el análisis de los datos geográficos.

Para comprender un GIS es necesario conocer los elementos básicos que lo componen. Los elementos principales que se contemplan son los siguientes:

- ✓ **Datos.** Los datos son la materia prima para el trabajo en un GIS.
- ✓ **Métodos.** Conjunto de fórmulas y metodologías a aplicar sobre los datos.

- ✓ **Software.** Aplicación informática que trate los datos y desarrolle los métodos.
- ✓ **Hardware.** El equipo para ejecutar el software.
- ✓ **Personas.** Son el motor del GIS, encargadas de diseñar y utilizar el software.



Gráfica 6: Componentes básicos de un SIG.
Fuente: Propia.

2.6.5. Cartografía.

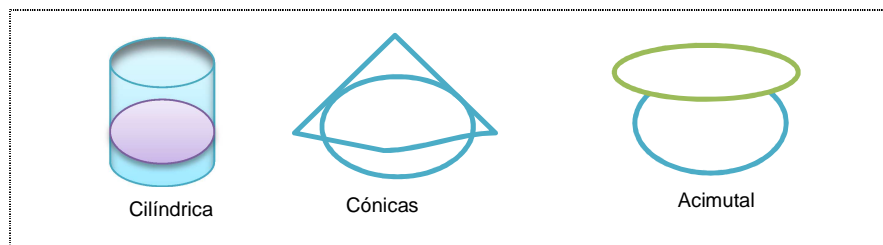
La cartografía es una ciencia, arte y técnica integradas que permiten representar la Tierra en una representación cartográfica o mapa. La Tierra tiene una forma que se denomina geoide y para representarla se la realiza a través de proyecciones.

2.6.6. Mapa

Un mapa es una representación lo más aproximado a la realidad, para esto se utiliza unas convenciones. En primer lugar la realidad a representar es generalmente volumétrica y por lo tanto implica un cambio de tres dimensiones a dos.

2.6.7. Proyecciones

Las proyecciones sirven para representar sobre un plano la superficie esférica de la Tierra con la menor deformación posible, utilizando para ello una red de meridianos y paralelos. Existen cientos de proyecciones, las cuales se agrupan en tres sistemas básicos: cilíndricas, cónicas y acimutales o polares. Las primeras utilizan como plano de proyección un cilindro tangente a la superficie de la Tierra. El segundo es un cono tangente o secante, y el último la proyección es tangente a un solo punto.



Gráfica 7: Tipos de Proyección
Fuente: Propia.

2.6.8. Escala.

La escala de un mapa es la relación matemática que existe entre éste y las dimensiones reales. Si la realidad es una escala 1:1, un mapa a escala 1:50.000 representaría: 500 m. de una carretera en 1 cm. La escala de un mapa establece el nivel de información que muestra el mismo.

2.6.9. Modelos para la información geográfica.

Los datos son una parte imprescindible de los GIS, ya que sin ellos, las aplicaciones GIS y otros elementos que se encuentran en su entorno no tendrían ninguna utilidad. Para hacer un estudio sobre un área geográfica es necesario disponer de datos sobre dicha área. No obstante, el proceso de convertir esa área geográfica y la información acerca de ella en un dato para un GIS no es una tarea fácil.

Este proceso requiere del desarrollo de un modelo de dato geográfico, que represente la realidad y sirva para conocer esta con claridad. El camino que va desde la realidad hasta el conjunto de valores numéricos pasa por tres niveles:

- **Establecimiento de un modelo geográfico.** Un modelo conceptual de la realidad geográfica y su conducta.
- **Establecimiento de un modelo de representación.** Forma de almacenar el modelo conceptual anterior y sus características, reduciéndolo a una serie de elementos.
- **Establecimiento de un modelo de almacenamiento.** Esquema para almacenar los distintos elementos del modelo de representación.

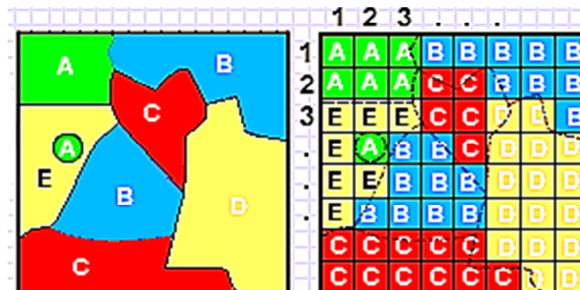
2.6.10. Modelos geográficos.

Inicialmente se conceptualiza el espacio estudiado. Este modelo geográfico es un esquema mental que constituye una forma particular de entender el hecho geográfico, pero que aún no incorpora elementos relativos a su representación o almacenamiento. Ejemplo: el espacio como campos, y como entidades discretas

2.6.11. Modelos de representación.

A partir de los modelos geográficos, se reducen sus propiedades a un conjunto finito de elementos, de tal manera que el registro de esos elementos sirva para almacenar la realidad que los modelos geográficos describen. Para esto se emplean los modelos de representación, también denominados modelos de datos. Se tienen los siguientes:

Modelo ráster: La zona de estudio se divide de forma sistemática en una serie de unidades mínimas denominadas celdas, y para cada una de estas se almacena la información pertinente que la describe.

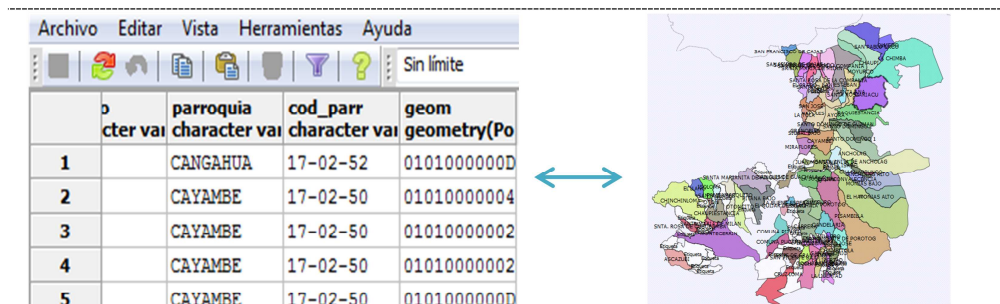


Gráfica 8: Modelo ráster.

Fuente: *Modelo Ráster*, Recuperado de:

http://www.cartovirtual.es/aprendizaje/cursoTIG/modelo_raster.html, 2013.

Modelo vectorial: Sus características geográficas se representan como vectores, manteniendo las características geométricas. Cada uno de estos vectores está vinculado a una fila en una base de datos que describe sus atributos.



Gráfica 9: Modelo vectorial SIG.

Fuente: Propia

Este modelo organiza el espacio geográfico mediante una serie de primitivas geométricas que contienen los elementos más destacados de dicho espacio. Estas primitivas son de tres tipos: puntos, líneas y polígonos.

2.6.12. Modelos de almacenamiento.

Este modelo es la última etapa en el proceso que llevan desde la realidad hasta el conjunto de valores numéricos que modelan esa realidad por medio de un GIS. Los modelos de almacenamiento implantan un esquema de cómo convertir las unidades que proponen los modelos anteriores en valores numéricos. Es decir, que especifica cómo escribir los datos en un soporte digital o guardarlos en un computador.

Modelos para representaciones ráster: Para almacenar una capa ráster simplemente se guardan sus valores uno a uno, en una estructura similar a la capa.

Modelos para representaciones vectoriales: Existen diferentes alternativas para guardar los elementos que componen una capa, entre ellas se establecen 4 alternativas: modelo spaghetti, diccionario de vértices, estructura arco-nodo, estructura TIN²¹.

2.7. SERVIDOR DE MAPAS.

Un Servidor de Mapas se compone de un software y un hardware que permite publicar datos GIS en Internet o en la Intranet, permitiendo al cliente interactuar con ellos en forma dinámica por medio de un navegador web.

2.7.1. MapServer.



MapServer es una plataforma de procesamiento de datos geográficos Open Source desarrollado en lenguaje C²², que permite crear aplicaciones cartográficas interactivas en la web.

Es un programa CGI²³ que se encuentra inactivo en el servidor web en espera de solicitudes.

2.7.2. PHP – MapScript.

Este es un módulo PHP que hace disponible las funcionalidades de MapScript en MapServer con una librería PHP de carga dinámica. En términos simples, este módulo le permitirá utilizar el potente lenguaje de script PHP para crear dinámicamente y modificar imágenes de mapa de MapServer.

2.7.3. Mapfile.

Este es el corazón del servidor de mapas MapServer, ya que en este se definen las relaciones entre los objetos, puntos y otros, además permite especificar los recursos que son utilizados en la aplicación, así como también contiene la información para plasmar el mapa, mostrar la leyenda.

Mapfile es un archivo de configuración con extensión “.map”. Está constituido por secciones las cuales tienen una funcionalidad específica.

²¹ **TIN** (Triangulated Irregular Network), estructura de dato digital usado en GIS.

²² **C**, Lenguaje de programación.

²³ **CGI**, especifica un estándar para transferir datos entre el cliente y el programa.

2.7.4. OGC.

Open Geospatial Consortium (OGC), creado en 1994 y agrupa a 372 organizaciones públicas y privadas. Define estándares abiertos e interoperables dentro de los sistemas de información geográfica.

La función que cumple esta organización es implementar acuerdos, estándares entre las empresas de desarrollo de herramientas GIS para que permitan la interoperación de sus herramientas de geo-procesamiento y además facilita el intercambio de la información geográfica para que los usuarios la puedan usar fácilmente. Anteriormente esta organización era conocida como Open GIS Consortium.

2.7.5. Shape.

El Shapefile es un formato de archivo de almacenamiento digital de representación vectorial sencillo diseñado por la ESRI²⁴ para sus aplicaciones GIS.

Este formato shapefile almacena digitalmente la localización de los elementos geográficos y la información de sus atributos en varios archivos:

- Archivo .shp: contiene la geometría de la entidad.
- Archivo .shx: contiene el índice de la geometría de la entidad.
- Archivo .dbf: tabla dBase, almacena la información de los atributos de las entidades.

2.7.6. WFS

Web Feature Service o Servicio de entidades vectoriales, es un servicio estándar, que suministra operaciones web en una interfaz de comunicación e interacción con los mapas servidos, lo cual permite realizar la consulta y edición de los datos geográficos. Este estándar permite:

- Mostrar las entidades geográficas disponibles (GetCapabilities).
- Agregar, modificar, eliminar entidades (Transaction), entre otras opciones.

2.7.7. WCS

Web Coverage Service o Servicio de Cobertura Web de OGC, es un servicio de datos que especifica un estándar que permite el acceso interoperable a “coberturas” geoespaciales a través de la web.

²⁴ **ESRI (Environmental Systems Research Institute):** empresa dedicada a desarrollar software para GIS.

Este estándar proporciona un servicio abierto de acceso a datos de coberturas, como por ejemplo imágenes satelitales de elevaciones, que se obtengan por el protocolo HTTP en la web. Un servicio WCS devuelve datos en un formato apropiado para que se pueda utilizar como entrada para el análisis y modelado de un GIS.

2.7.8. WMS

Web Map Service o Servicio de mapas en la web, es un servicio que proporciona “mapas” bajo demanda en formatos de imagen (jpeg, png, gif) que pueden ser invocados desde cualquier visor GIS o navegador web.

Los servicios WMS permiten la consulta y publicación de mapas de forma dinámica a partir de datos geográficos, estos mapas pueden contener una o varias capas ráster o vectoriales en un orden específico.

2.7.9. WMC

Web Map Context es una especificación de la OGC y define un formato para almacenar el contexto de la cartografía del mapa.

El WMC se define como un documento de esquema XML²⁵ que describe la configuración para que los visores de mapas puedan reconstruir los aspectos de los mapas.

El contexto almacena toda la información necesaria sobre las capas que constituyen el mapa representado por el usuario, la región que cubre el mapa, los servidores de donde se los obtiene.

2.8. CLIENTES LIVIANOS GIS.

2.8.1. Definición.

Un cliente liviano o cliente ligero se define como un computador cliente o un software cliente dentro de una arquitectura de red cliente - servidor, el cual depende principalmente del servidor central para realizar las tareas de procesamiento, por lo tanto, este se centra en el transporte de la entrada y la salida entre el usuario y el servidor remoto.

Los clientes ligeros web para GIS son aplicaciones en el internet que se encargan de visualizar información geográfica y permiten su manipulación a través de herramientas básicas de navegación y análisis.


²⁵ XML (eXtensible Markup Language), lenguaje de marcas extensible.

2.8.2. Características.

Las características básicas de los clientes livianos son los siguientes:

- ✓ No implementa nada relacionado a la lógica de la aplicación.
- ✓ Actúa como intermediario entre el usuario y el servidor
 - Atiende las entradas y las envía al servidor.
 - Muestra datos y resultados que devuelve el servidor.
- ✓ Los recursos de hardware son mínimos.
- ✓ Aumenta la complejidad del servidor.

2.8.3. Pmapper.

 Es un framework basado en MapServer PHP/MapScript, desarrollado por Argim Burger. Este framework permite publicar mapas en MapServer, para lo cual dispone de una gran variedad de funcionalidades y múltiples configuraciones.

Entre sus características se destacan, las funciones de búsqueda, visualización de resultados de consultas con bases de datos, impresión en HTML²⁶ y PDF²⁷, y la configuración que cambian la forma de mostrar los mapas.

2.9. CLIENTES PESADOS GIS.

2.9.1. Definición.

Un cliente pesado GIS es una aplicación que se ejecuta en el ordenador del usuario para proporcionar las funcionalidades de búsqueda, visualización, consulta y análisis de datos basadas en los servicios OGC disponibles.

2.9.2. Características.

Las características fundamentales de los clientes pesados GIS son las siguientes:

- ✓ Realiza la mayor parte de la lógica de aplicación.
- ✓ Procesa los datos del usuario antes de enviar al servidor.
- ✓ Demanda equipos con buena capacidad de proceso y almacenamiento de datos.

²⁶ **HTML** (HyperText Markup Language), lenguaje de marcas de hipertexto.

²⁷ **PDF** (Portable Document Format), formato de documento portátil.

2.9.3. Quantum GIS.



También llamado QGIS, es un Sistema de Información Geográfica de Código Abierto con licencia bajo GNU/GPL. Además es un proyecto oficial de OSGeo²⁸. Se ejecuta sobre Windows, Mac OSX, Linux, Unix y Android²⁹, soporta numerosos formatos y funcionalidades de datos vector, datos ráster y bases de datos.

2.10. REALIDAD VIRTUAL.

2.10.1. Antecedentes.

A fines de los años 80, los gráficos creados por computador entraron en una nueva época. Las imágenes tridimensionales empezaron a reemplazar a las bidimensionales, también surgió la necesidad de un lugar de trabajo interactivo concebido por medio de la tecnología.

En 1989, Jaron Lanier³⁰, popularizó el término "Realidad Virtual". En los años siguientes, los sistemas de realidad virtual se vieron enriquecidos con impresiones del mundo real por medio de estímulos auditivos, visuales.

2.10.2. Definición.

Realidad virtual es una ciencia que emplea los computadores y otros dispositivos, con la finalidad de reproducir una apariencia de la realidad que permita al usuario tener la sensación de estar presente en ella.

2.10.3. Visita virtual

Una visita virtual es una simulación de un espacio real, esta se forma de varias imágenes unidas mediante un software, creando una imagen interactiva que el visitante puede visualizar en 360° mediante el ordenador.

2.10.4. Características de la visita virtual

- Las visitas virtuales consisten en una serie de fotografías de alta definición, que son tratadas de forma individual, y una vez unidas de forma inteligente, forman una sola fotografía panorámica de hasta 360°.

²⁸ **OSGeo** (Open Source Geospatial Foundation), organización que da soporte y promueve el desarrollo de tecnologías geoespaciales.

²⁹ **Android**, sistema operativo basado en Linux

³⁰ **Jaron Lanier**: informático, artista visual y autor. Pionero que popularizó el término "Realidad Virtual".

- Tienen la facultad de moverse en toda dirección de forma manual o automática.
- Se puede acercar y alejar sobre cualquier zona de la panorámica.
- Se pueden incluir botones de control que hagan aún más fácil el manejo.
- Tiene la capacidad de aumentar los detalles fotográficos.
- El tamaño de la presentación en pantalla puede ser incluso a pantalla completa, si bien para este modo se aconseja una conexión de alta velocidad.
- Para las visitas virtuales, se cuenta con tecnología digital y los medios más avanzados en el desarrollo de aplicaciones multimedia para internet.

2.10.5. Beneficios de las visitas virtuales

- Incrementa la confianza en los clientes ya que muestra lo real sobre el lugar.
- Aumentan las visitas al lugar donde se ha realizado la visita virtual. Mejora la opinión del cliente sobre el sitio que se observa.
- Acorta el miedo que los visitantes tienen sobre lo desconocido y agranda la confianza al permitirle desplazarse libremente por las visitas.
- Muestra con realismo el lugar a visitar. Una imagen vale más que mil palabras
- Muestra los sitios con mucho detalle, sin necesidad de leer grandes catálogos.

2.10.6. Fotografía panorámica

Una fotografía panorámica es una imagen cuyo tamaño es mayor en horizontal que en vertical, conocida como fotografía de amplio formato. Es una técnica de la fotografía, en la cual se emplea equipos especializados que capturan imágenes con puntos de vista panorámicos.

2.10.7. Fotografía 360°

Imágenes especiales ya que el borde derecho se une con el izquierdo formando una imagen continua. Para crear estas imágenes se unen fotografías en horizontal hasta que se haya girado 360 grados, de manera que la última foto solape a la primera.

2.10.8. Fotografía cilíndrica

Las fotografías cilíndricas de 360 grados son aquellas imágenes cuyo ancho abarca toda la longitud del horizonte mientras que su alto no alcanza a cubrir todo el espacio.

Expresado en grados, son fotografías que miden 360 grados de ancho por menos de 180 grados de alto. Se proyectan como si estuviesen dentro de un cilindro.

2.10.9. Fotografía esférica

Se la conoce como imágenes de formato equi - rectangular, y se formulan en 360° de ancho por 180° de alto, este tipo de imagen se proyectan como si estuviesen dentro de una esfera.

2.10.10. Fotografía cúbica

Las fotografías cúbicas son la composición de 6 imágenes que representan cada lado del cubo, permiten que los observadores puedan mirar hacia arriba y abajo al piso, se las puede representar en formato flash³¹ o Quicktime³².

2.11. LENGUAJE DE PROGRAMACIÓN PHP.

2.11.1. Antecedentes

PHP fue creado por Rasmus Lerdorf³³ en 1995. En la Actualidad este lenguaje sigue siendo desarrollado con nuevas funciones. Este lenguaje está publicado bajo la licencia PHP de software libre.

2.11.2. Definición.



PHP (Hypertext Preprocessor) es un lenguaje de programación interpretado de script de código abierto, que se ejecuta en el servidor, creado para la implementación de sitios web interactivos y dinámicos, el código generado se incrusta en las páginas HTML.

PHP permite crear una infinidad de aplicaciones web debido a la gran cantidad de librerías de funciones y módulos que posee. Entre sus capacidades se puede mencionar: compatibilidad de conexión con las bases de datos, subida de archivos al servidor, envío de correos, creación de imágenes entre otras.

2.12. FRAMEWORK SYMFONY.

2.12.1. Definición

Según (Potencier & Zaninotto, 2007), Symfony dice que es:

³¹ **Flash:** Tecnología para realizar animaciones gráficas vectoriales

³² **Quicktime:** framework multimedia implementado por Apple, compuesto por librerías y un reproductor multimedia.

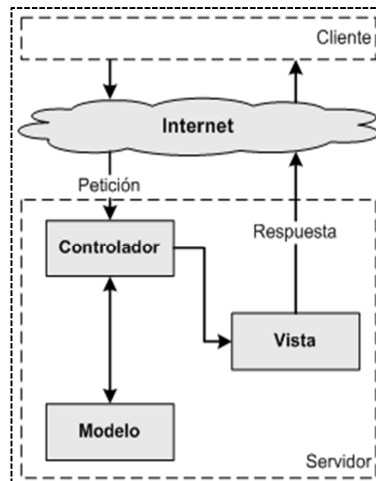
³³ **Rasmus Lerdorf:** programador informático creador de la primera versión del lenguaje de programación PHP.

Un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja.

2.12.2. MVC en Symfony

Symfony se basa en el patrón de diseño web MVC que está formado por tres niveles:

- El Modelo, representa la lógica de negocio.
- La Vista, convierte el modelo en una página web para que interactúe el usuario.
- El Controlador, procesa las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.



Gráfica 10: Patrón MVC de Symfony.

Fuente: *El patrón MVC*, Recuperado de:

http://librosweb.es/symfony_1_4/capitulo_2/el_patron_mvc.html, 2013

2.12.3. ORM

Un ORM (Object Relational Mapper) Mapeador Objeto Relacional, es una técnica de programación en la que se transforman en clases, atributos y tipo de datos, todas las tablas, columnas y relaciones de una base de datos, dentro de un lenguaje orientado a objetos, de tal manera que se crea una base de datos virtual. En general el ORM se encarga de convertir de manera automática los registros de una tabla en objetos de una clase para procesarlos y al momento de almacenarlos realiza lo contrario, es decir, convierte los objetos en registros de una tabla.

2.12.4. Doctrine.

La librería Doctrine está inspirada en Hibernate, y es un ORM que proporciona una capa de persistencia para objetos PHP. Provee una capa de abstracción que se ubica sobre un SGBD³⁴.

Se caracteriza ya que posee su propio lenguaje para el acceso a datos, llamado DQL, el cual permite trabajar con las operaciones básicas de las bases de datos: crear, eliminar, modificar, seleccionar.

2.12.5. DQL.

DQL (Doctrine Query Language) Lenguaje de Consulta Doctrine, es un lenguaje para consultas propio del ORM Doctrine. La sintaxis se asemeja a la utilizada en SQL³⁵, pero se emplea para obtener objetos en vez de filas de tablas.

2.12.6. YAML.

“YAML Ain't Another Markup Language”, YAML no es otro lenguaje de marcado. Es un formato para serialización de archivos diseñado en torno a los tipos de datos nativos comunes de los lenguajes de programación ágiles.

Este es un lenguaje de serialización de datos diseñado para ser amable con el humano y trabajar de manera correcta con los lenguajes de programación modernos, como XML, C, Python³⁶, Perl³⁷.

2.13. COMERCIO ELECTRÓNICO.

2.13.1. Antecedentes

Años atrás se iniciaron las ventas por catálogo en los Estados Unidos. Este sistema de venta, revolucionario para la época, consistió en un catálogo por internet con fotos de los productos a vender, que permitió a las empresas captar nuevos segmentos de mercado que no estaban siendo atendidos.

Un punto importante de esta forma de vender es que los compradores pueden seleccionar los productos en la tranquilidad de sus hogares, sin la asistencia o presión del vendedor.

³⁴ **SGBD**, Sistema de Gestión de Base de Datos.

³⁵ **SQL** (structured query language), lenguaje de consulta estructurado.

³⁶ **Python**, es un lenguaje de programación interpretado.

³⁷ **Perl** (Practical Extraction and Report Language), Lenguaje de programación Práctico para la Extracción e Informe.

2.13.2. Definición.

El comercio electrónico, o también conocido como “e-commerce”, es una modalidad para distribuir, vender y/o adquirir todo tipo de producto o servicio con el uso de cualquier medio electrónico, como las redes de información o el internet. Es una herramienta que emplean las empresas para la reducción de costos y mejorar las relaciones comerciales.

2.13.3. PayPal.

PayPal[™] PayPal es una empresa estadounidense que pertenece al sector del comercio electrónico por internet, que brinda servicio para pagar y recibir pagos en línea de la forma más rápida y segura, a través de un dispositivo móvil. Este servicio da a los clientes formas más sencillas para enviar dinero sin compartir su información financiera, y con la flexibilidad de pagar con los saldos de sus cuentas, cuentas bancarias, tarjetas de crédito.

2.14. WEB 2.0.

2.14.1. Antecedentes.

En los inicios de la web, se disponían de sitios web en las que los usuarios solamente eran simples espectadores de su contenido ya que estas eran estáticas, pero con el avance de la tecnología, la actitud, y las redes sociales se establecieron nuevas formas de crear aplicaciones dinámicas en la web, las cuales permitan a los usuarios crear contenido colaborativo y mantener una interacción entre ellos.

2.14.2. Definición.



Web 2.0 es un concepto que se acuña en el año 2003 y es el producto de un fenómeno social que se produjo desde el surgimiento varias aplicaciones en Internet. Este concepto concibe la diferencia entre la primera época de la Web, y la revolución y el auge de los sitios web pensados en el usuario, los blogs, las redes sociales y otras.

CAPÍTULO III: HERRAMIENTAS DE DESARROLLO

3.1. INTRODUCCIÓN GENERAL.

Este capítulo está dedicado al análisis de las herramientas de desarrollo planteadas en el capítulo 1. En los siguientes párrafos se hace un análisis de los lenguajes de programación a utilizar, así como también de las herramientas de software a emplear en el diseño de la solución. En los últimos párrafos de este capítulo se realiza un planteamiento del diseño arquitectónico del sistema, y finalmente se describe a detalle la metodología a seguir para el desarrollo del software.

3.2. LENGUAJE DE PROGRAMACIÓN.

Actualmente existen diferentes lenguajes de programación para el desarrollo web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. En las siguientes líneas se pretende mostrar las ventajas y desventajas de los lenguajes seleccionados para el desarrollo del proyecto.

3.2.1. Lenguaje HTML



Es un lenguaje de programación estático para el desarrollo de aplicaciones web. Desarrollado por el W3C³⁸. Los archivos pueden tener las extensiones (htm, html). La sintaxis es la siguiente:

```
<html>(Inicio del documento HTML)
<head> ( Cabecera ) </head>
<body>
  ( Cuerpo )
</body>
</html> (fin del documento HTML)
```

Ventajas:

- Fácil y sencillo para representar hipertexto.
- Presenta el código en forma vistosa, organizada y estructurada.
- Se dispone de editores de páginas web o WYSIWYG³⁹, por lo tanto no es necesario tener mucho conocimiento de programación.
- Los ficheros que se generan son pequeños.
- Facilita un rápido despliegue.
- Tiene mucha documentación por lo tanto es fácil de aprender.
- Todos los navegadores web admiten este lenguaje.

³⁸W3C, es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

³⁹WYSIWYG (What You See Is What You Get), lo que ves es lo que obtienes. Procesadores y editores de texto HTML.

Desventajas:

- El lenguaje generado es estático.
- Los browsers o navegadores pueden interpretar el código de forma diferente.
- Guarda etiquetas que se convierten en “basura” y obstaculizan la corrección.
- El diseño de sitios web son más lentos.

3.2.2. Lenguaje JavaScript



Es un lenguaje de programación interpretado. Fue creado por Brendan Eich⁴⁰ en la empresa Netscape Communications⁴¹. Este lenguaje es semejante a Java⁴², con la diferencia de que no está orientado a objetos.

Este código es introducido dentro de las páginas web, para ejecutar funciones programadas. Debido a las incompatibilidades la W3C diseño el estándar DOM⁴³. Su sintaxis es la siguiente:

```
<script type="text/javascript">  
    Contenido  
</script>
```

Ventajas:

- Lenguaje fiable.
- Poseen capacidades limitados por motivo de seguridad.
- Se ejecuta al lado del cliente.

Desventajas:

- Código visible para todo usuario.
- Código se descarga totalmente.
- El problema XSS⁴⁴, puede colocar en peligro la seguridad del sitio.

3.2.3. Lenguaje PHP



Es un lenguaje de programación para crear sitios web dinámicos. PHP significa “PHP Hypertext Pre-processor”, inicialmente se llamó Personal Home Page. Apareció en 1995, desarrollado por PHP Group.

⁴⁰ **Brendan Eich** programador de computadoras, más conocido por inventar el lenguaje de programación JavaScript.

⁴¹ **Netscape Communications Corporation**, empresa de software creadora del navegador web Netscape Navigator.

⁴² **Java** es un lenguaje de programación por objetos.

⁴³ **DOM** (Document Object Model), Modelo de Objetos del Documento.

⁴⁴ **XSS (Cross-site scripting)**, es un tipo de inseguridad informática o agujero de seguridad típico de las aplicaciones Web.

Este se utiliza para crear código para páginas web dinámicas, dicho código se incrusta en páginas HTML y estas se ejecutan en el servidor. Para que se ejecute PHP es necesario tener instalado un servidor web con las librerías de PHP. Los archivos cuentan con la extensión (.php).

Sintaxis:

```
<?php //inicio
    $echo "Hola mundo";
?> //fin
```

Ventajas:

- Fácil de aprender, documentación que incluye descripción y ejemplos.
- Soporta la orientación a objetos. Clases y herencia.
- Lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con los gestores de base de datos más utilizados: MySQL⁴⁵, PostgreSQL, Oracle⁴⁶, MS SQL Server⁴⁷, entre otras.
- Facultad de expandir su funcionalidad con el uso módulos o extensiones.
- Es libre, lo que favorece su uso.
- Posee cientos de funciones. No necesita la definición de tipos de variables.

Desventajas:

- Dependiente de un servidor web instalado y corriendo.
- El servidor es el encargado de ejecutar todo el trabajo y no delega nada al cliente. Lo que resulta ineficiente a medida que las solicitudes aumenten.
- Puede ser complicado leer el código al mezclar sentencias HTML y PHP.
- En aplicaciones grandes la programación orientada a objetos puede ser ineficiente.
- Dificulta la organización de la aplicación por capas.

Seguridad:

PHP es un fuerte lenguaje, ya sea como un módulo del servidor web o ejecutado como un binario CGI independiente, tiene la facultad de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor, lo cual resulta inseguro por naturaleza. Por tal motivo, PHP está diseñado para ser más seguro al momento de escribir programas, ya que permite configurar su conducta de compilación y ejecución.

⁴⁵ **MySQL**, sistema de gestión de bases de datos relacional, multi-hilo y multiusuario

⁴⁶ **Oracle**, es un sistema de gestión de base de datos objeto-relacional.

⁴⁷ **MS SQL Server**, es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relaciona

3.2.4. XML.



XML (eXtensible Markup Language), Lenguaje extensible de marcas, es un lenguaje formado por un conjunto de reglas que sirven para definir etiquetas semánticas para organizar un documento.

XML es un metalenguaje, lo que permite diseñar un propio lenguaje de etiquetas.

```
<?xml version='1.0' encoding='UTF-8'?>
<Persona>
  <nombre>Carlos</nombre>
  <apellido>Caluguillin</apellido>
</Persona>
```

Ventajas:

- De fácil interpretación por humanos y por software, y es gratuito.
- Separa radicalmente el contenido de su presentación o formato.
- Puede ser utilizado con cualquier lenguaje o alfabeto. Estructura jerárquica
- El análisis sintáctico es fácil por las reglas de composición del documento.
- Poderosos enlaces (XLL⁴⁸).

Desventajas:

- Con el tiempo la mayoría de las ventajas que disponibles se pueden volver sus desventajas.
- Requiere más espacio en disco, ya que un archivo de texto es de mayor tamaño que un binario.

3.2.5. CSS



CSS (Cascading Style Sheets), Hojas de estilo en cascada, es una herramienta impresionante para agregar estilos de presentación a los sitios web. Ahorra tiempo y permite diseñar sitios web de un modo totalmente nuevo. Es imprescindible para los que trabajan en el diseño web.

La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios:

⁴⁸ XLL (eXtensible Linking Language): Define el modo de enlace entre diferentes enlaces.

| Navegador | Motor | CSS 1 | CSS 2.1 | CSS 3 |
|-------------------|---------|---|----------|--|
| Google Chrome | WebKit | Completo desde la versión 85 del motor | Completo | Todos los selectores, pseudo-clases y muchas propiedades |
| Internet Explorer | Trident | Completo desde la versión 7.0 del navegador | Completo | Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador |
| Firefox | Gecko | Completo desde la versión 1.0 del navegador | Completo | Todos los selectores, pseudo-clases y muchas propiedades |
| Safari | WebKit | Completo desde la versión 85 del motor | Completo | Todos los selectores, pseudo-clases y muchas propiedades |
| Opera | Presto | Completo desde la versión 1.0 del navegador | Completo | Todos los selectores, pseudo-clases y muchas propiedades |

Tabla 1: Soporte de CSS en los navegadores.

Fuente: Soporte de CSS en los diferentes navegadores, Recuperado de, http://librosweb.es/css/capitulo_1/soporte_de_css_en_los_navegadores.html, 2013

Ventajas:

- Aplicación de varios estilos a diferentes tipos de medios (impresión, pantalla, y otros). Posee técnicas sofisticadas y avanzadas.
- Con una sola hoja de estilo es posible alterar la presentación de cada elemento sin manipular el código HTML, ahorrando esfuerzo y tiempo de edición.
- El lenguaje de CSS ofrece una gran cantidad de herramientas de composición.
- Evita recurrir a trucos para obtener efectos. Potente, sencillo y fácil de aprender.
- Los documentos que usan CSS resultan más compactos.
- Se puede usar con otros lenguajes de programación para obtener efectos dinámicos en las páginas.
- Especificar hojas de estilo para diferentes navegadores y tipos de medios.

Desventajas

- El navegador aplica un formato y el trabajo resulta casi inútil.
- Varias propiedades de CSS ocasionan que partes de la página resulten inaccesibles para algunos navegadores si no son empleados correctamente.

3.2.6. AJAX

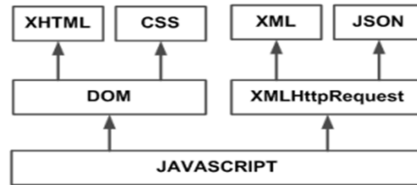


AJAX es el acrónimo de Asynchronous JavaScript and XML, e indica una mezcla de tecnologías que proveen el diseño de aplicaciones web con algunas características avanzadas.

Según (Eguiluz Perez, 2008), dice que Ajax no es: “una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes”.

Ajax está formado por varias tecnologías:

- XHTML⁴⁹ y CSS, crea presentaciones basadas en estándares.
- DOM, permite la manipulación dinámica de la presentación.
- XML, XSLT⁵⁰ y JSON⁵¹, para el intercambio y manipulación de información.
- XMLHttpRequest⁵², para el intercambio asíncrono de información.
- JavaScript, para unir todas las tecnologías involucradas.



Gráfica 11: Tecnologías agrupadas bajo el concepto AJAX.

Fuente: Capítulo 1. *Introducción a AJAX*, Recuperado de: http://librosweb.es/ajax/capitulo_1.html, 2013

Ventajas

- Utiliza tecnologías existentes.
- La mayoría de navegadores la soportan.
- Interactividad. El usuario no esperan hasta que lleguen los datos del servidor.
- Portabilidad (sin plug-in Flash⁵³ o Applet de Java⁵⁴)
- Mayor velocidad, porque no retorna toda la página nuevamente.
- La página se parece a una aplicación de escritorio.

Desventajas

- No existe el concepto de volver a la página anterior.
- La mezcla de páginas con AJAX y otras sin ella desorientan al usuario.
- Problemas con navegadores antiguos que no implementan esta tecnología.
- No se ejecuta si el usuario desactiva JavaScript en el navegador.
- Requiere programadores que conozcan las tecnologías que intervienen.
- La carga del servidor es demorosa lo cual aturde al visitante.

3.2.7. YAML.



YAML es un lenguaje de marcado ligero que permite especificar de forma sencilla estructuras como arrays con menos caracteres que XML.

⁴⁹ **XHTML** (eXtensible HyperText Markup Language), es HTML expresado como XML válido.

⁵⁰ **XSLT** (Extensible Stylesheet Language Transformations), lenguaje de transformación de documentos XML.

⁵¹ **JSON** (JavaScript Object Notation), formato ligero para el intercambio de datos.

⁵² **XMLHttpRequest** (Extensible Markup Language - Hypertext Transfer Protocol Request).

⁵³ **Plug-in flash**, programa que permite cargar animaciones multimedia por internet.

⁵⁴ **Applet Java**, componente escrito en lenguaje java.

YAML es usado en frameworks de PHP como Symfony dada su facilidad de uso y lo amigable para la edición y lectura por los humanos.

Características

Las características de YAML son:

- Legible para humanos.
- Datos portables entre lenguajes de programación.
- Concuerda con las estructuras nativas de datos de lenguas ágiles.
- Tiene un modelo coherente para apoyar herramientas genéricas.
- Procesamiento de una sola pasada.
- Expresivo y extensible. Fácil de implementar y utilizar.

Convenciones básicas de YAML

YAML fue creado para que los datos sean representados adecuadamente como combinaciones de listas, hashes y datos escalares. La sintaxis está diseñada para que sea legible, y fácilmente mapeable a los tipos de datos de los lenguajes de alto nivel.

- El contenido de YAML se escribe usando el conjunto de caracteres imprimibles de Unicode⁵⁵, bien en UTF-8⁵⁶ o UTF-16⁵⁷.
- La estructura del documento se denota indentando con espacios en blanco; sin embargo no se permite la tabulación para indentar⁵⁸.

Archivos YAML en Symfony

El framework Symfony utiliza por defecto el formato YAML para la configuración. Este formato indica su estructura mediante la tabulación y es muy rápido de escribir. Sin embargo, se deben tener en cuenta las convenciones al escribir estos archivos. A continuación un ejemplo de un archivo de configuración de Symfony en YAML.

`databases.yml`: (`miproyecto/config/databases.yml`), contiene la definición de los accesos a bases de datos y las opciones de conexión de cada acceso.

```
doctrine:
  class: sfDoctrineDatabase
  param:
    dsn:      postgresql:host=localhost;dbname=bdd_turismo
    username: postgres
    password: postgres
```

⁵⁵ **Unicode**, estándar de codificación de caracteres.

⁵⁶ **UTF-8** (8-bit Unicode Transformation Format), formato de codificación de caracteres Unicode e ISO 10646.

⁵⁷ **UTF-16** (UCS Transformation Format for 16 Planes of Group 00), ISO/IEC 10646:2003, codifica caracteres UCS y Unicode.

⁵⁸ **Indentar**, uso en informática que significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores.

3.2.8. DQL.

DQL (Doctrine Query Language), Lenguaje de consulta doctrine es un lenguaje de consulta de objetos, creado para colaborar con los usuarios en la recuperación de objetos complejos. Lenguaje de consulta utilizado por el ORM de Symfony.

Tipos de consultas DQL

DQL tiene consultas SELECT, UPDATE y DELETE. Las instrucciones INSERT no se admiten en DQL, debido a que las entidades y sus relaciones se introducen en el contexto de la persistencia para garantizar la coherencia de los objetos del modelo.

Consultas Select: forma poderosa de recuperar partes del modelo de objetos. Además se puede recuperar entidades y sus asociaciones con una única consulta SQL lo que aumenta el rendimiento de la aplicación.

En el siguiente ejemplo se realiza una consulta DQL select básica de una tabla:

```
$query = Doctrine_Query::create()
    ->select('u.*')
    ->from('Usuario u')
    ->execute();
```

El ejemplo anterior en SQL puro se realizaría de la siguiente manera:

```
SELECT u.id AS u__id,
       u.nombre AS u__nombre,
       u.apellido AS u__apellido,
       u.direccion AS u__direccion
FROM usuario u
```

Consultas Update y Delete: permiten ejecutar cambios masivos en las entidades del modelo de objetos. Esto, es necesario cuando no se carga en memoria todas las entidades afectadas en una actualización masiva. En los siguientes ejemplos se realiza una consulta DQL Update y una Delete:

```
$query = Doctrine_Query::create()
    ->update('Usuario')
    ->set('direccion', 'El Olivo')
    ->where('id = ?', 2);

$query2 = Doctrine_Query::create()
    ->delete('Usuario u')
    ->where('u.id = ?', 2);
```

Ventajas

Al comparar con el uso de SQL crudo, DQL tiene varias ventajas:

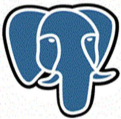
- Diseñado para recuperar los objetos no filas del conjunto de resultados.
- Interpretan las relaciones por lo que no se escribe manualmente JOIN⁵⁹ SQL.
- Portátil en diferentes bases de datos.
- Algoritmos complejos incorporados, para ayudar a desarrolladores a recuperar objetos de forma eficiente.
- Compatible con varias funciones que ahorran tiempo cuando se trata de datos relacionados de uno – a – muchos, muchos – a – muchos.

Desventajas.

- DQL es insensible a mayúsculas y minúsculas, salvo en nombres de clase, campos y espacios de nombres.

3.3. BASES DE DATOS POSTGRESQL

3.3.1. Introducción

 PostgreSQL es un sistema gestor de bases de datos objeto relacional, bajo licencia BSD⁶⁰, de código fuente libre. Utiliza un modelo cliente - servidor y usa multiprocesos en vez de multi hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afecta al resto y el sistema continúa funcionando.

3.3.2. Características principales.

Sus características la hacen una de las bases de datos más potentes y robustas del mercado. Para su creación los desarrolladores tomaron en cuenta la estabilidad, potencia, robustez, facilidad de administración e implementación de estándares. PostgreSQL funciona bastante bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez a él sistema.

A continuación las características más importantes soportadas por PostgreSQL:

⁵⁹ **JOIN**, sentencia SQL que permite combinar registros de dos o más tablas en una base de datos relacional.

⁶⁰ **BSD** (Berkeley Software Distribution), licencia de distribución de software Berkeley.

| Generales |
|---|
| Es una base de datos 100% ACID ⁶¹ . |
| Soporta distintos tipos de datos. También permite la creación de tipos propios. |
| Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales. |
| Copias de seguridad en caliente (Online/hot backups) |
| Unicode |
| Juegos de caracteres internacionales. |
| Regionalización por columna. |
| Multi-Version Concurrency Control (MVCC ⁶²) |
| Múltiples métodos de autenticación |
| Acceso encriptado vía SSL ⁶³ |
| SE-postgres ⁶⁴ |
| Completa documentación |
| Licencia BSD |
| Disponible para Linux, UNIX, Mac OS X, Solaris y Windows 32/64bit. |

Tabla 2: Características generales de SGBD PostgreSQL.

Fuente: Propia

3.3.1. Arquitectura de PostgreSQL.

Los componentes más importantes en un sistema PostgreSQL son:

- **Demonio postmaster:** Proceso principal. Encargado de atender las conexiones entrantes de los clientes a través de un puerto, crea procesos hijos.
- **Procesos hijos PostgreSQL:** Encargados de autenticar a los clientes, gestionar las consultas y enviar resultados a las aplicaciones clientes.
- **Ficheros de configuración:** Ficheros de configuración utilizados por PostgreSQL: postgresql.conf, pg_ident.conf, pg_hba.conf.
- **PostgreSQL share buffer cache:** Memoria compartida empleada por PostgreSQL para recopilar información en caché.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos.
- **Kernel disk buffer cache:** Caché de disco del sistema operativo.
- **Aplicación cliente:** Aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ser vía TCP/IP⁶⁵ o sockets⁶⁶ locales.
- **Disco:** Disco duro físico donde se guardan los datos y toda la información necesaria que sirve para que PostgreSQL se ejecute.

⁶¹ **ACID** (Atomicity, Consistency, Isolation, Durability), Atomicidad, Consistencia, Aislamiento y Durabilidad,

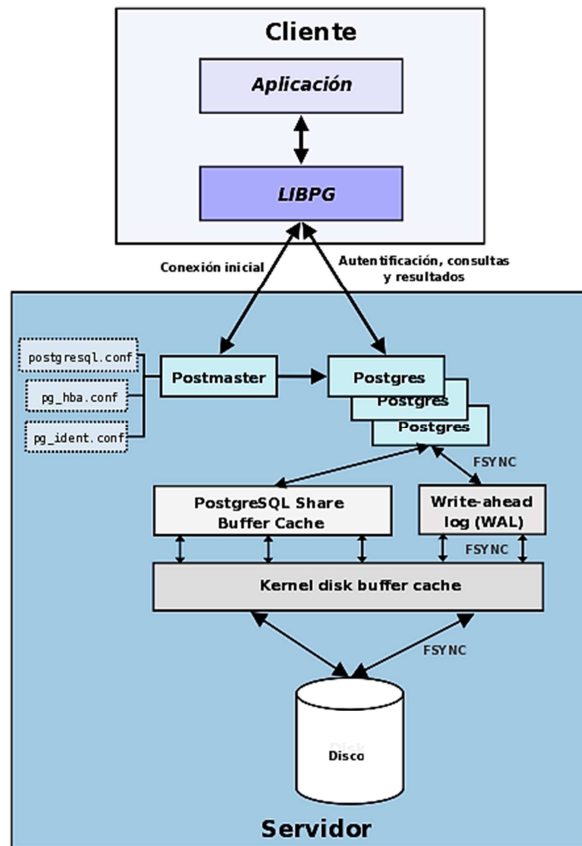
⁶² **MVCC** (Multi-Version Concurrency Control), control de concurrencia mediante versiones múltiples.

⁶³ **SSL** (Secure Sockets Layer), capa de conexión segura.

⁶⁴ **SE-Postgresql** (Security-Enhanced PostgreSQL), mejora de políticas de seguridad incorporada de PostgreSQL.

⁶⁵ **TCP/IP** (Protocolo de control de transmisión/Protocolo de Internet), conjunto de protocolos de internet.

⁶⁶ **Socket**, método de comunicación entre un programa del cliente y el servidor en una red



Gráfica 12: Componentes importantes de SGBD PostgreSQL.
Fuente: *Componentes Principales de SGBD PostgreSQL*, Recuperado de:
http://www.postgresql.org.es/sobre_postgresql, 2013.

3.3.2. Requerimientos hardware.

PostgreSQL fue diseñado e implementado para arrancar en máquinas lentas que posean pocos megabytes de memoria RAM. Esto no implica que PostgreSQL no funcione en computadores de gran capacidad. La velocidad del procesador es de lo depende su ejecución. Además es compatible con varios microprocesadores y sistemas operativos como Linux, Solaris, y Windows.

3.3.3. Requerimientos software.

PostgreSQL es compatible con Microsoft Windows, pero la mayoría de usuarios lo ejecutan sobre plataformas UNIX. UNIX ofrece varias ventajas sobre Windows.

3.3.4. Limitaciones.

A continuación las limitaciones principales de PostgreSQL:

| Límite | Valor |
|-------------------------------------|---|
| Máximo tamaño base de dato | Ilimitado (Depende de tu sistema de almacenamiento) |
| Máximo tamaño de tabla | 32 TB ⁶⁷ |
| Máximo tamaño de fila | 1.6 TB |
| Máximo tamaño de campo | 1 GB |
| Máximo número de filas por tabla | Ilimitado |
| Máximo número de columnas por tabla | 250 - 1600 (dependiendo del tipo) |
| Máximo número de índices por tabla | Ilimitado |

Tabla 3: Limitaciones de PostgreSQL.

Fuente: *Componentes Principales de SGBD PostgreSQL*, Recuperado de: http://www.postgresql.org.es/sobre_postgresql, 2013

3.3.5. Ventajas.

Entre las principales ventajas de PostgreSQL tenemos las siguientes:

| Ventajas |
|--|
| Ampliamente popular - Ideal para tecnologías Web. |
| Fácil de Administrar. |
| Su sintaxis SQL es estándar y fácil de aprender. |
| Footprint bajo de memoria, bastante poderoso con una configuración adecuada. |
| Capacidades de replicación de datos. |
| Soporte empresarial disponible. |
| Estabilidad y confiabilidad legendarias |
| Multiplataforma y Extensible. |
| Diseñado para ambientes de alto volumen |

Tabla 4: Ventajas de PostgreSQL.

Fuente: Propia.

3.3.6. Desventajas

Las desventajas más claras se enumeran a continuación:


| Desventajas |
|---|
| Comparado con MySQL el proceso de actualización e inserción es más lento, debido a que PostgreSQL posee cabeceras de intersección, las cuales MySQL no posee. |
| El soporte en línea es a través de los foros oficiales, pero la ayuda no siempre es ofrecida. |
| Requiere de más recursos que MySQL. |
| Sintaxis de varias sentencias y comandos no son instintivas. |

Tabla 5: Desventajas de PostgreSQL.

Fuente: Propia

3.4. MAPSERVER

3.4.1. Introducción.

 Es una plataforma de procesamiento de datos geográficos de software abierto desarrollado en lenguaje C, surge como un proyecto llamado ForNet de la Universidad de Minnesota, el Departamento de Recursos Naturales de Minnesota y en colaboración con la NASA⁶⁸. Utiliza su propio lenguaje para el despliegue de capas de información geográfica, leyendas, escalas, simbología y la

⁶⁷ TB Terabyte, unidad de almacenamiento informático,

⁶⁸ NASA (National Aeronautics and Space Administration), Administración Nacional de la Aeronáutica y del Espacio

configuración de ellas tanto en ancho líneas, color, etiquetado, y otras, las cuales se definen en el archivo de configuración Mapfile.

3.4.2. Requerimientos hardware.

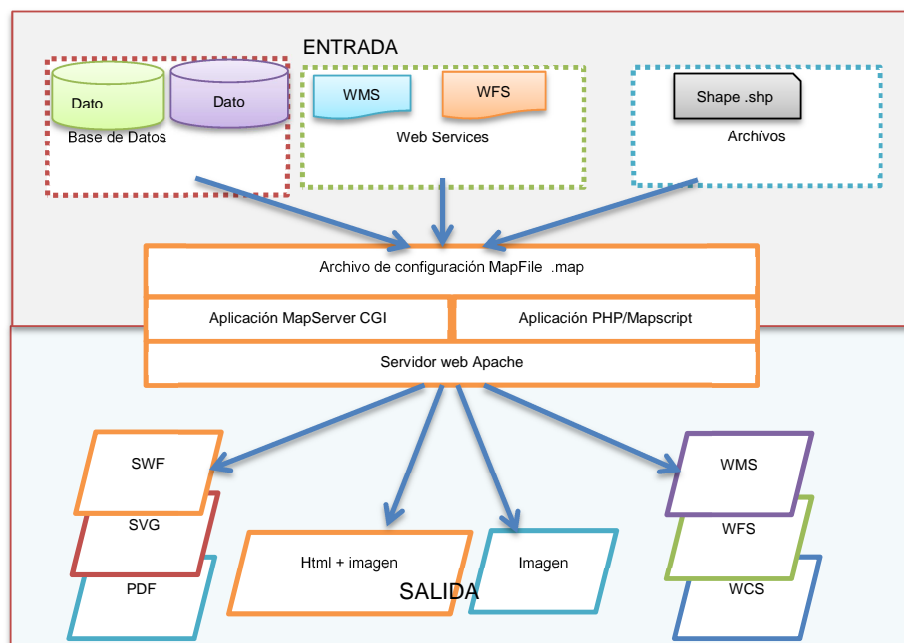
MapServer se ejecuta en Linux, Windows, Mac OS X, Solaris, entre otros. Para compilar o instalar los programas requeridos, es necesario tener derechos de administrador en la máquina. Los requerimientos de hardware para fines de desarrollo y aprendizaje son mínimos, pero para la implementación, los requerimientos son más altos ya que se tiene que optimizar todos los recursos.

3.4.3. Requerimientos software.

Se necesita tener configurado correctamente un sitio de trabajo y un servidor Web (HTTP), como Apache o Microsoft IIS⁶⁹, en la máquina en la que va a instalar MapServer.

3.4.4. Arquitectura básica de MapServer.

El servidor de mapas MapServer generalmente se ejecuta como una aplicación CGI, que permite establecer comunicación entre un servidor web y un programa, de tal modo que este último pueda interactuar con internet y corre dentro de un servidor http. La arquitectura básica de MapServer consta de los siguientes componentes:



Gráfica 13: Arquitectura de MapServer.

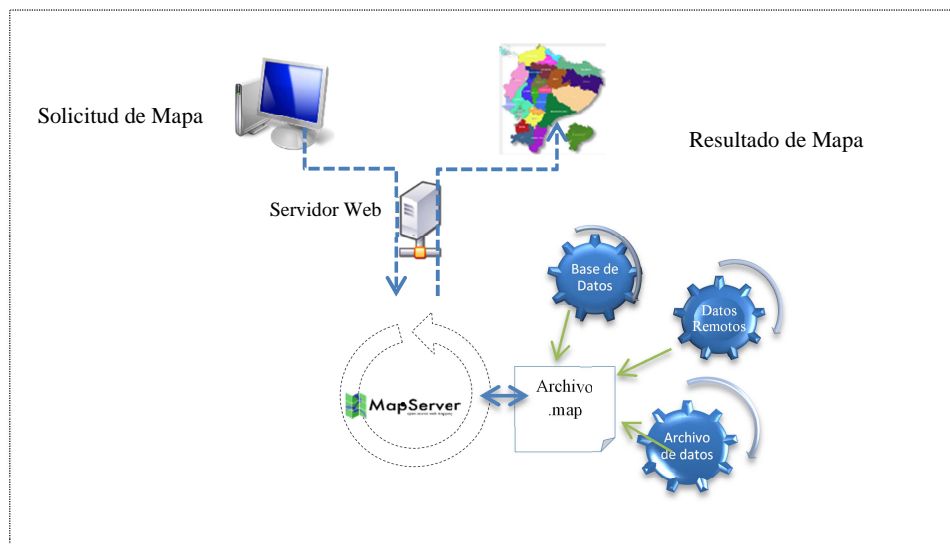
Fuente: Propia

⁶⁹ IIS, Internet Information Services, servidor web y servicios para Microsoft Windows.

- ✓ **Mapfile:** Corazón de MapServer ya que es su archivo principal de configuración. En este fichero se puntualiza el área del mapa, las capas, la localización y fuente de los datos, proyecciones y simbología. Extensión .map.
- ✓ **Datos geográficos:** MapServer puede utilizar diferentes tipos de fuentes de datos geográficos. Formato por defecto es ESRI Shape.
- ✓ **Páginas Html:** interfaz entre el usuario y MapServer.
- ✓ **MapServer CGI:** Binario o ejecutable que atiende las solicitudes y retorna imágenes, datos, y otros. Se ubica en el directorio cgi-bin o scripts del servidor web. El programa se llama mapserv.
- ✓ **Servidor web Http – Apache:** Interpreta la solicitud del usuario desde un navegador y retorna una página html.

3.4.5. Funcionamiento de MapServer

Su funcionamiento está configurado en un archivo de texto estructurado con extensión “.map”, este compone el mapa que está estructurado en capas, las capas se dividen en clases. El resultado que devuelve MapServer, obedece a la solicitud, este puede ser un gráfico con “mapa, leyenda, escala, métricas, visión” o alfanumérico “resultado de una solicitud de datos espaciales o alfanuméricos”. A continuación se muestra gráficamente el flujo de funcionamiento de MapServer:



Gráfica 14: Funcionamiento de MapServer.

Fuente: Propia

3.4.6. Características principales.

A continuación se detallan las características principales de MapServer:

| CARACTERÍSTICAS DE MAPSERVER |
|---|
| Se ejecuta bajo plataformas Linux/Apache y Windows (MS4W). |
| Formatos vectoriales soportados: ESRI shapefiles, PostGIS, ESRI ArcSDE, GML ⁷⁰ y otros. |
| Formatos ráster soportados: JPG, PNG, GIF, TIFF/GeoTIFF, EPPL7 ⁷¹ y otros vía GDAL/OGR ⁷² . |
| Fuentes: TrueType. |
| Configuración rápida vía parámetros GET pasados por URL ⁷³ . |
| MapScript facilita API para acceder a las funcionalidades de MapServer a través de lenguajes de programación como PHP, Java, Perl, Python, Ruby o C#. |
| Soporte a estándares interoperables como WMS, WFS, WCS. |

Tabla 6: Características de servidor de mapas MapServer.

Fuente: Propia

3.4.7. Ventajas.

Entre las ventajas más relevantes de MapServer se describen las siguientes:

- ✓ Permite mostrar los datos a otros usuarios empleando Internet.
- ✓ Soluciones a los proyectos con mapas en múltiples formatos GDAL/OGR.
- ✓ Acceso de datos y performance GDAL/OGR PostGIS.
- ✓ Portabilidad y Costo \$0.00.

3.4.8. Desventajas.

- ✓ Dificultad en la compilación del paquete.
- ✓ No incluye componente de metadatos.

3.5. IDE DE DESARROLLO – NETBEANS 7.0.1

3.5.1. Introducción

Los Entornos de Desarrollo Integrado o IDE (Integrated Development Environment), son unos programas compuestos por un conjunto de herramientas para un programador. Puede dedicarse a un solo lenguaje de programación o a varios de ellos.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI⁷⁴).

⁷⁰ **GML** (Geography Markup Language), Lenguaje de Marcado Geográfico.

⁷¹ **EPP7** (Environmental Planning and Programming Language). Formato basado en ráster.

⁷² **GDAL/OGR** (Geospatial Data Abstraction Library), Herramientas para tratamiento de datos Geoespaciales

⁷³ **URL** (Uniform Resource Locator), localizador de recursos uniforme.

Para el lenguaje de programación PHP se disponen varios IDEs, entre los cuales se puede destacar: Eclipse, Aptana Studio, Netbeans, Php Designer, Open Komodo Project. Este proyecto se centra en el IDE de desarrollo Netbeans.

3.5.2. Definición.



NetBeans 7.0.1 es un entorno de desarrollo integrado disponible para Windows, Mac, Linux y Solaris⁷⁵. El proyecto Netbeans consiste es un IDE de código abierto y una plataforma de aplicaciones que permite a los desarrolladores crear rápidamente aplicaciones web, de escritorio y aplicaciones móviles utilizando la plataforma Java. El proyecto de NetBeans está apoyado por una comunidad de desarrolladores y ofrece una gran documentación, recursos de capacitación, y plugins de terceros.

3.5.3. Requerimientos de software.

El IDE Netbeans está diseñado para correr sobre los siguientes sistemas operativos: Windows, Linux (x86/x64), Solaris (x86/x64), Solaris (sparc), MAC OSX. Además requiere el JDK⁷⁶, que consta del JRE⁷⁷.

3.5.4. Requerimientos de hardware.

- ✓ Intel Pentium III o superior
- ✓ Espacio en disco: mínimo 750 Mbytes
- ✓ Memoria: mínimo 521 Mbytes RAM⁷⁸

3.5.5. Características.

Actualmente está integrado con servidores de aplicaciones GlassFish, Apache Tomcat y utiliza bases de Datos MySQL, PostgreSQL y cualquiera que se conecte con JDBC⁷⁹ como Oracle, SQL Server. Además tiene soporte de lenguajes para el desarrollo de la especificación de Java SE 7 con características del lenguaje JDK7. Incluye soporte de edición de HTML 5, mejoras en el editor Java, soporte para PHP 5.3. Adicionalmente tiene soporte para el framework Symfony. Disponible en inglés, portugués, japonés, ruso y chino. En la siguiente gráfica se muestra la descarga:

⁷⁴ **GUI** (Graphical User Interface), Interfaz Gráfica de Usuario

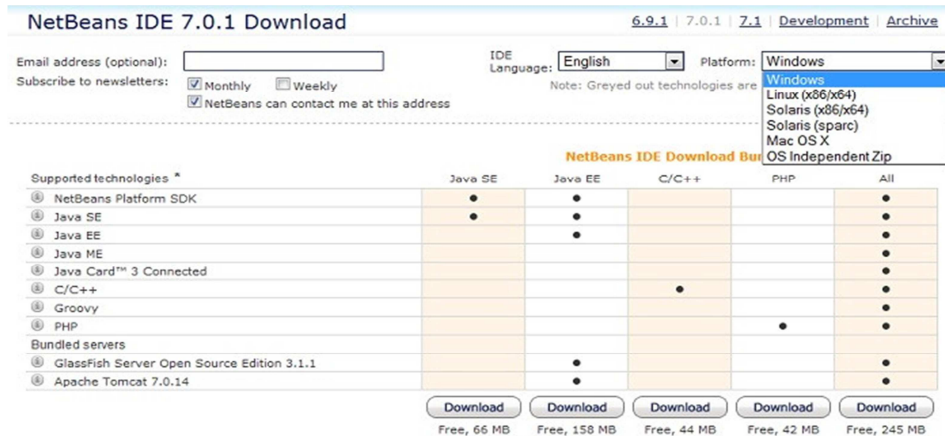
⁷⁵ **Solaris**, es un sistema operativo de tipo Unix

⁷⁶ **JDK** (Java Development Kit), conjunto de herramientas que permiten desarrollar programas en lenguaje Java

⁷⁷ **JRE** (Java Runtime Environment), conjunto de utilidades que permite la ejecución de programas Java.

⁷⁸ **RAM** (Random Access Memory), Memoria de Acceso Aleatorio.

⁷⁹ **JDBC** (Java Database Connectivity), permite la ejecución de operaciones sobre bases de datos desde el lenguaje Java.



Gráfica 15: Características de IDE Netbeans 7.0.1.

Fuente: *Netbeans 7.0.1 download*, Recuperado de:

<https://netbeans.org/downloads/7.0.1/index.html>, 2013

3.5.6. Ventajas.

- ✓ Facilita las tareas en proyectos grandes.
- ✓ Provee un asiste parcial para la escritura de código.
- ✓ Ayuda en la navegación de las clases pre-definidas en la plataforma.
- ✓ Grandes beneficios en su uso.

3.5.7. Desventajas

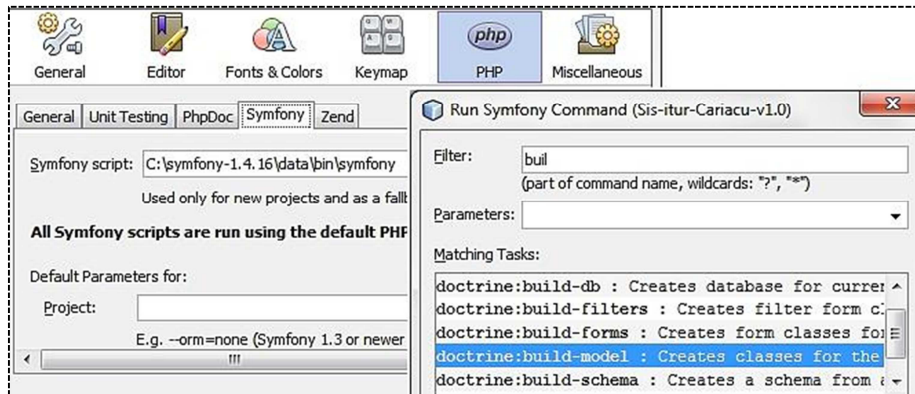
- ✓ Gran consume de CPU⁸⁰ y RAM.
- ✓ No se cuenta en idioma español para esta versión.

3.5.8. Integración con Symfony

En septiembre de 2008 inició una campaña para que NetBeans eligiera Symfony como su primer framework. Luego de 13 meses después NetBeans 6.8 completó el soporte de Symfony.

Esta integración permite desarrollar aplicaciones de forma más sencilla y productiva. En primer lugar, permite crear nuevos proyectos directamente desde el IDE. Además permite ejecutar todas las tareas de Symfony, incluso pasándole argumentos y opciones, sin la necesidad de utilizar una consola de comandos externa. Además en los archivo de una vista se tiene acceso al autocompletado de variables, incluso de los objetos del núcleo de Symfony. NetBeans también permite saltar de una acción al archivo de su vista asociada y viceversa. Para esto es necesario disponer de Symfony 1.x.

⁸⁰ CPU (Central Processing Unit), Unidad Central de procesamiento o procesador.




Gráfica 16: Integración de Symfony con Netbeans.

Fuente: Propia

3.6. FRAMEWORK SYMFONY 1.4.16

3.6.1. Introducción

 En el proceso de desarrollo del software hay que tomar en cuenta otro aspecto muy importante como lo es la forma de crear código, es decir, si se programa en crudo o con la ayuda de algún framework.

Un framework, por lo general, se define como aquella aplicación o conjunto de módulos que permiten, o tienen por objetivo, el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas para que los programadores las usen directamente.

Symfony es un framework PHP de software libre que permite crear aplicaciones y sitios web rápidos y seguros de forma profesional, está construido con varios componentes independientes creados por el proyecto Symfony.

3.6.2. Pre requisitos para instalación.

Primeramente se debe contar con un entorno de desarrollo web funcionando: un servidor web, un motor de bases de datos (MySQL, PostgreSQL, SQLite⁸¹ o cualquier otra base de datos que soporte PDO⁸²), y PHP 5.2.4 o superior.

3.6.3. Características de Symfony

Symfony se diseñó para que se ajuste a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de plataformas.

⁸¹ SQLite, es un sistema de gestión de bases de datos relacional compatible con ACID.

⁸² PDO (PHP Data Objects), extensión Objetos de Datos PHP.

- ✓ Es independiente de la bases de datos.
- ✓ Fácil de extender con el uso de librerías creadas por terceros.
- ✓ Sencillo de usar, y adaptable a los procesos más complicados.
- ✓ El programador solo tiene que configurar lo que no es convencional.
- ✓ Utiliza las mejores prácticas y patrones de diseño para la web.
- ✓ Idóneo para aplicaciones empresariales, y flexible a las arquitecturas de cada empresa, además es estable para crear aplicativos a largo plazo
- ✓ Código fácil de interpretar y mantener por los comentarios de phpDocumentor⁸³.

3.6.4. Implementación MVC de Symfony.

Symfony implementa lo mejor del patrón de diseño MVC en la creación de aplicaciones rápidas y sencillas. Para la creación de una aplicación sencilla se necesita de los siguientes componentes:



Gráfica 17: Componentes de aplicación básica en Symfony.

Fuente: Propia.

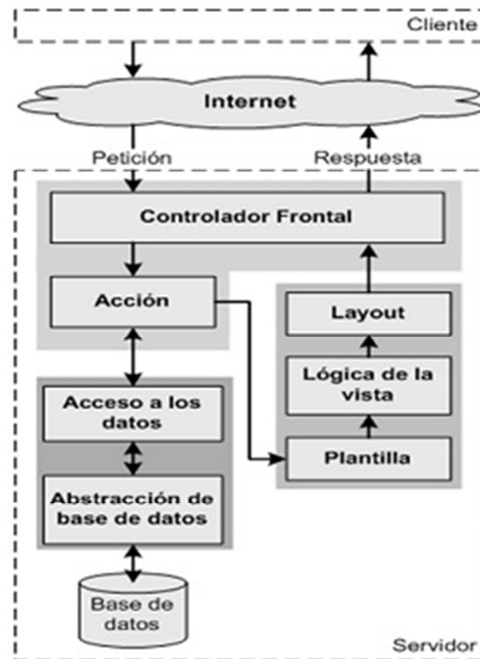
A primera vista son muchos scripts los que se necesitarían para crear una aplicación sencilla. Pero hay que tener en cuenta varios aspectos que Symfony emplea para su optimización.

- ✓ El **controlador frontal** y el **layout** son comunes para todas las acciones del aplicativo, lo cual simplifica la generación de código. Además el controlador frontal es código relativo MVC que lo genera automáticamente Symfony.
- ✓ El contenido en la **capa de modelo** se genera automáticamente, de acuerdo a la estructura de los datos de la aplicación.
- ✓ El **ORM** genera la estructura básica de las clases con todo el código necesario.
- ✓ La **abstracción de la base de datos** se realiza de forma transparente para el desarrollador, ya que se realiza a través de PDO, lo cual implica que en cualquier

⁸³ **phpDocumentor**, es un generador de documentación de código abierto escrito en PHP

momento se puede cambiar de gestor de base datos sin escribir código adicional, lo único que se realizará es cambiar los parámetros de configuración.

- ✓ La lógica de la **vista** no es necesario programarla, se puede cambiar en un archivo de configuración sencillo.



Gráfica 18: Flujo de trabajo de Symfony.

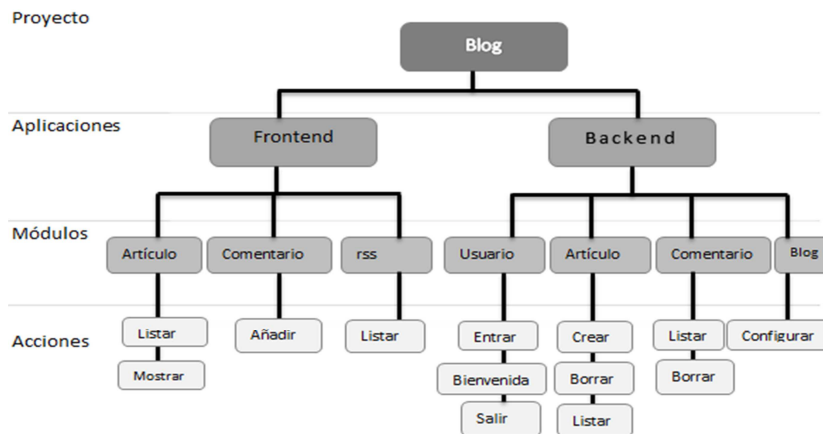
Fuente web: *El patrón MVC*, Recuperado de:

http://librosweb.es/symfony_1_4/capitulo_2/el_patron_mvc.html, 2013.

3.6.5. Estructura del proyecto.

El framework Symfony define un proyecto como "un conjunto de servicios y operaciones disponibles bajo un determinado nombre de dominio y que comparten el mismo modelo de objetos".

- Dentro de un **proyecto**, las operaciones se agrupan en aplicaciones.
- Cada, **aplicación** se ejecuta de manera independiente respecto de otras aplicaciones que tenga el mismo proyecto. Por lo general, un proyecto contiene dos aplicaciones: una para la parte pública y otra para la parte de administración.
- Cada aplicación está formada por uno o más módulos. Un **módulo** puede representar a una página web o a un grupo de páginas relacionadas.
- Los módulos almacenan las **acciones**, que representan cada una de los procedimientos que se puede realizar en un módulo.



Gráfica 19: Organización del código en Symfony.
Fuente: Propia.

3.6.6. Estructura del árbol de archivos.

Un proyecto en el framework Symfony está estructurado en directorios que organizan de forma lógica todo su contenido, esto se genera automáticamente cuando se crea un nuevo proyecto. Esta estructura es completamente personalizable, para reorganizar los archivos y directorios o de acuerdo con las exigencias del cliente.

| DIRECTORIO | DESCRIPCIÓN |
|--------------------|--|
| Proyecto | Directorio global del proyecto |
| apps/ | Contiene los directorios de las aplicaciones. |
| frontend/ | Es el directorio de la aplicación frontal. |
| backend/ | Es el directorio de la aplicación de administración. |
| cache/ | Contiene la versión cacheada de la configuración y (si está activada) la versión cacheada de las acciones y plantillas del proyecto. |
| config/ | Almacena las configuraciones generales del proyecto. |
| data/ | Directorio que almacenan los archivos relacionados con los datos. |
| sql/ | Contiene el esquema de la base de datos en SQL. |
| doc/ | Contiene la documentación del proyecto. |
| lib/ | Almacena librerías y clases externas |
| model/ | Contiene el modelo de objetos del proyecto. |
| log/ | Guarda los archivos logs que se generan. |
| plugins/ | Almacena los plugins instalados en el proyecto. |
| test/ | Almacena las pruebas unitarias y funcionales escritas en PHP y compatibles con las pruebas de Symfony |
| bootstrap/ | Contiene archivos de las pruebas unitarias y funcionales. |
| unit/ | Pruebas unitarias del proyecto. |
| functional/ | Pruebas funcionales del proyecto. |
| web/ | Es la raíz del servidor web, contiene los archivos accesibles desde Internet. |
| css/ | Contiene archivos de hojas de estilos. |
| images/ | Contiene imágenes. |
| js/ | Contiene archivos javascript. |
| uploads/ | Contiene los archivos subidos al servidor. |

Tabla 7: Directorios de proyecto Symfony.
Fuente: Propia

La estructura del directorio de archivos de cada aplicación de un proyecto en Symfony se describe en la siguiente tabla:

| DIRECTORIO | DESCRIPCIÓN |
|---------------------|---|
| apps/ | Contiene las aplicaciones. |
| [nombre aplicación] | Es el nombre de la aplicación. |
| config/ | Contiene archivos de configuración de la aplicación. |
| i18n/ | Contiene archivos para internacionalización de la aplicación. |
| lib/ | Almacena clases y librerías utilizadas en la aplicación. |
| modules/ | Contiene los módulos de la aplicación. |
| templates/ | Contiene plantillas |
| layout.php | Plantilla principal de la aplicación |

Tabla 8: Directorio de aplicación Symfony.

Fuente: Propia

Las aplicaciones contienen uno o más módulos, de acuerdo a las necesidades del proyecto. Los módulos tienen su propio subdirectorio, el cual se describe en la siguiente tabla:

| DIRECTORIO | DESCRIPCIÓN |
|---------------------|---|
| apps/ | Contiene las aplicaciones. |
| [nombre aplicación] | Es el nombre de la aplicación. |
| modules/ | Directorio que contiene todos los módulos de la aplicación. |
| nombre módulo/ | Nombre del módulo. |
| actions/ | Almacena la clase de acción del módulo. |
| action.class.php | Es la clase acción del módulo. |
| config/ | Contiene configuraciones adicionales del módulo. |
| lib/ | Contiene las clases y librerías utilizadas solo en el módulo. |
| templates/ | Contiene las plantillas correspondientes a cada acción. |
| indexSuccess.php | Plantilla de inicio de módulo. |

Tabla 9: Directorio de módulo de Symfony.

Fuente: Propia.

3.6.7. Los entornos.

En el directorio web/, del proyecto se encuentran los controladores frontales de las aplicaciones, estos controladores son el archivo index.php y frontend_dev.php, dependiendo del nombre de la aplicación. Los dos archivos atienden todas las peticiones que se generan en la aplicación, además estos archivos apuntan a la misma aplicación pero se utilizan en diferentes entornos. En Symfony se pueden contar con varios entornos:

- ✓ **Entorno de desarrollo (dev):** entorno para los programadores web para corregir errores y agregar características.
- ✓ **Entorno de pruebas (test):** entorno utilizado para ejecutar automáticamente las pruebas unitarias.
- ✓ **Entorno intermedio:** entorno utilizado por el cliente para probar la aplicación, informar errores, características que le faltan a la aplicación.
- ✓ **Entorno de producción (prod):** entorno en el que utilizan los usuarios finales.

3.6.8. Sistema de configuraciones.

Symfony define varias normas para su configuración y uso. Los archivos de configuración son sencillos y muy potentes para personalizar cualquier aspecto.

- ✓ La configuración de Symfony se guarda en archivos escritos con YAML.
- ✓ Tiene archivos de configuración a nivel de proyecto, aplicación y de módulo.
- ✓ Permite definir conjuntos de opciones de configuración (entorno).
- ✓ Se puede acceder a los valores establecidos en los ficheros de configuración desde cualquier parte del código de la aplicación.
- ✓ Se puede incorporar código PHP dentro de los archivos YAML

Configuración del proyecto

Symfony crea por defecto algunos archivos de configuración relacionados con el proyecto. El directorio `proyecto/config/` contiene los siguientes archivos:

- ✓ `ProjectConfiguration.class.php`: Contiene la ruta a los archivos del framework.
- ✓ `databases.yml`: accesos a bases de datos y opciones de conexión de cada acceso.
- ✓ `properties.ini`: parámetros que utiliza la herramienta de línea de comandos.
- ✓ `rsync_exclude.txt`, `rsync_include.txt`: archivos y directorios que se excluyen/incluyen durante la sincronización entre servidores.
- ✓ `schema.yml`: representación del modelo de datos relacional del proyecto. Si usa Doctrine, este archivo se encuentra en el directorio `config/doctrine/`

Configuración de la aplicación

La configuración de la aplicación es muy importante al momento de configurar. Esta configuración se realiza en el controlador frontal del directorio **web/** y el directorio **config/** de la aplicación que contiene varios archivos YAML.

- ✓ Configuración del Controlador Frontal (`web/`): configurar el controlador frontal (`web/index.php`), es el primer script que se ejecuta con cada petición.

Archivos de configuración en el directorio `proyecto/apps/frontend/config/`:

- ✓ `app.yml`: configuración específica de la aplicación; variables globales.
- ✓ `frontendConfiguration.class.php`: esta clase inicia la ejecución de la aplicación
- ✓ `factories.yml`: clases adicionales para el manejo de la vista, peticiones, y otros.
- ✓ `filters.yml`: En este archivo se definen los filtros que se van a procesar.
- ✓ `routing.yml`: almacena las reglas de enrutamiento.
- ✓ `settings.yml`: principales opciones de configuración de una aplicación.

- ✓ `view.yml`: estructura de la vista por defecto: layout, CSS, JavaScript.

Configuración de los módulos

Al crear un módulo no tienen ninguna configuración propia, pero es posible modificar estas opciones. Los archivos de configuración se encuentran en el directorio `proyecto/apps/frontend/modules/modulo/config/`.

- ✓ `generator.yml`: para módulos generados automáticamente desde una tabla de la base de datos.
- ✓ `module.yml`: configuración de la acción y otros parámetros del módulo.
- ✓ `security.yml`: restringir el acceso a determinadas acciones del módulo.
- ✓ `view.yml`: configurar las vistas de una o de todas las acciones del módulo.

3.6.9. El Modelo (DOCTRINE)

En proyectos desarrollados en Symfony, el acceso y la modificación de los datos guardados en la base de datos se realiza mediante objetos. Este medio permite un gran nivel de abstracción y su portabilidad. El componente que facilita esto en Symfony es el ORM: Propel y Doctrine.

Esquema de base de datos de Symfony: Para generar el modelo de objetos de datos, se debe realizar un mapeo de la base de datos, para eso el ORM necesita una descripción del modelo relacional o esquema. El archivo `schema.yml` se encuentra en el directorio `proyecto/config/doctrine`.

Las clases del modelo: Para generar las clases del modelo se debe ejecutar la tarea de `symfony doctrine:build-model`. Al ejecutar este comando, el framework examina el esquema y crea las clases base del modelo, que se guardan en el directorio `lib/model/doctrine/base`. Además, crea las clases en el directorio `lib/model/doctrine`.

3.6.10. Ventajas.

- ✓ Basado en PHP 5, por lo tanto, soporta el lenguaje orientado a objetos.
- ✓ Independiente de la base de datos usada. La capa de abstracción y el ORM facilita el cambio de base de datos sin ningún problema.
- ✓ Fácil de instalar en cualquier plataforma.
- ✓ Rápido para el desarrollo, conveniente para aplicaciones grandes.
- ✓ Usa los patrones y las mejores prácticas en el desarrollo web.
- ✓ Muy estable, escalable⁸⁴ y LTS⁸⁵.

⁸⁴ **Escalable**, es la propiedad deseable de un sistema, para reaccionar y adaptarse a cambios.

- ✓ Código fácil de leer y de mantener. Soporte y desarrollo creciente.

3.6.11. Desventajas.

- ✓ Tiempo de aprendizaje extenso, no se puede dominar en un par de días.
- ✓ Depende de la versión de PHP.

Comparativa entre frameworks PHP

A continuación se ilustra una tabla comparativa con los aspectos más relevantes de algunos frameworks PHP:

| | Cake PHP | Symfony | Zend Framework | Code Igniter |
|--------------------------|---------------|---------------|----------------|---------------|
| Durabilidad | 80,00% | 80,00% | 90,00% | 63,33% |
| Solución industrializada | 50,00% | 67,65% | 64,71% | 47,06% |
| Adaptabilidad | 83,33% | 100,00% | 100,00% | 83,33% |
| Estrategia | 50,00% | 50,00% | 41,67% | 41,67% |
| Media: | 65,83% | 74,41% | 74,10% | 58,85% |

Tabla 10: Evaluación por el método QSOs⁸⁶.

Fuente web: *Libro blanco sobre frameworks php para empresas*, Recuperado de: <http://fr.clever-age.com/veille/publications/livres-blancs/livre-blanc-%EF%BB%BF%EF%BB%BFframeworks-php.html>, 2013.

3.7. POSTGIS.

3.7.1. Introducción.



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. PostGIS es estable, rápido, compatible con estándares, con cientos de funciones espaciales y actualmente es la base de datos espacial de código abierto más utilizada. La administración de la base de datos es posible a través de pgAdmin y phpPgAdmin, entre otros.

Es posible importar y exportar datos mediante herramientas en línea de comandos o a través de clientes SIG de escritorio o web.

3.7.2. Requerimientos.

El principal requerimientos de instalación de PostGIS es tener instalado y configurado PostgreSQL 8.4 o superior. Además son necesarios los paquetes GEOS, PROJ, que permiten el uso de funciones espaciales y la re proyección. Corre bajo sistemas operativos Windows, Linux, Unix, Mac.

⁸⁵ **LTS** (Long Term Support), Soporte Técnico Extendido.

⁸⁶ **QSOs** (Qualification and Selection of Open Source software), calificación y selección de software de código abierto.

3.7.3. Características únicas de PostGIS.

El hecho de que está construido sobre PostgreSQL, PostGIS hereda de forma automática sus características, así como los estándares abiertos. A continuación un listado de algunas características que le hacen único:

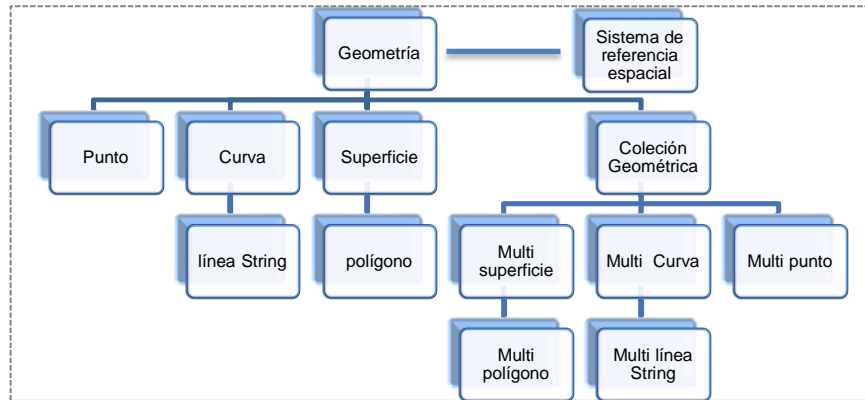
- ✓ PostGIS es software libre, tiene licencia GNU/GPL.
- ✓ Es compatible con los estándares de OGC.
- ✓ Soporta tipos de datos espaciales, índices espaciales y tiene cientos de funciones espaciales.
- ✓ Permite importar y exportar datos a través de las herramientas conversoras (shp2pgsql, pgsq2shp, ogr2ogr, dxf2postgis).
- ✓ Alternativa al software propietario que lo supera en estabilidad y rapidez.
- ✓ Buen número de clientes GIS de escritorio para visualizar datos PostGIS: uDig, QGIS, mezoGIS, OpenJUMP, SpatialKit para ArcGIS, gvSIG, GRASS, ArcGIS, Manifold, GeoConcept, MapInfo, autoCAD Map 3D.
- ✓ Servidores de mapas web: MapServer, GeoServer, MapGuide, ArGIS Server.
- ✓ Es la base de datos espacial de código abierto más utilizada.

3.7.4. Objetos SIG soportados.

Una base de datos común tiene cadenas, números y fechas, mientras que una base de datos espacial agrega otros tipos de datos para representar características geográficas. Estos tipos de datos espaciales resumen y encapsulan estructuras espaciales como límite y dimensión. Los objetos GIS soportados por PostGIS los define OpenGIS. A continuación ejemplos de la representación de objetos en modo texto:

- ✓ POINT(0 0 0)
- ✓ LINESTRING(0 0,1 1,1 2)
- ✓ POLYGON((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))
- ✓ MULTIPOINT(0 0 0,1 2 1)
- ✓ MULTILINESTRING((0 0 0,1 1 0,1 2 1),(2 3 1,3 2 1,5 4 1))
- ✓ MULTIPOLYGON(((0 0 0,4 0 0,4 4 0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0)),((-1 -1 0,-1 -2 0,-2 -2 0,-2 -1 0,-1 -1 0)))
- ✓ GEOMETRYCOLLECTION(POINT(2 3 9),LINESTRING((2 3 4,3 4 5))

Los tipos de datos espaciales se organizan en una jerarquía de tipos. Cada subtipo hereda la estructura (atributos) y comportamiento de su tipo súper.



Gráfica 20: Jerarquía de tipo de datos espaciales.

Fuente web: *Introducción PostGIS*, Recuperado de: <http://workshops.boundlessgeo.com/postgis-intro/introduction.html>, 2013.

3.7.5. Aplicaciones que soportan PostGIS.

PostGIS es una de las bases de datos espaciales más utilizadas, existen varios programas de otros fabricantes que soportan para el almacenamiento y recuperación de datos. La siguiente tabla muestra algunos de los programas que soportan PostGIS:

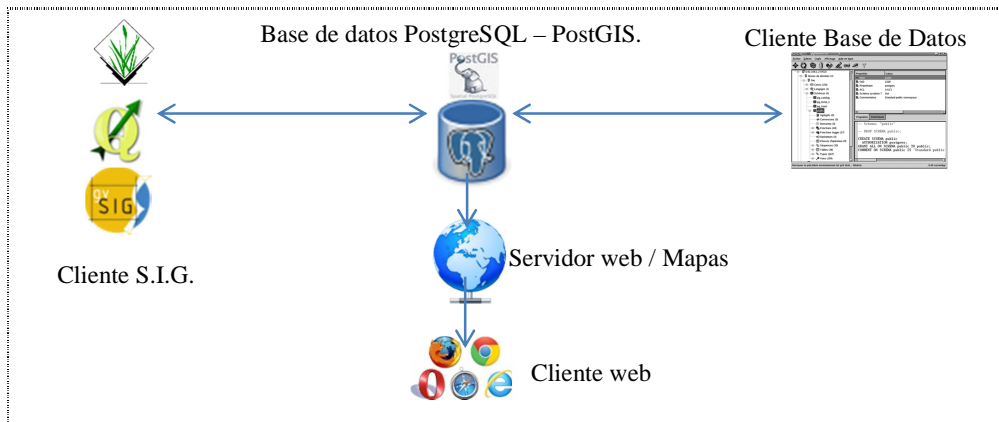
| Abierto – Libre | Cerrado – Propietario |
|---|---|
| <ul style="list-style-type: none"> • Carga/Extracción <ul style="list-style-type: none"> • Shp2Pgsql, ogr2ogr, Dxf2PostGIS | <ul style="list-style-type: none"> • Carga/Extracción <ul style="list-style-type: none"> • Safe FME Desktop Translator/Converter |
| <ul style="list-style-type: none"> • Servidor de Mapas <ul style="list-style-type: none"> • Mapserver • GeoServer • SharpMap SDK - para ASP.NET 2.0 • MapGuide Open Source (mediante FDO) | <ul style="list-style-type: none"> • Servidor de Mapas <ul style="list-style-type: none"> • Ionic Red Spider (ahora ERDAS) • Cadcorp GeognoSIS • Iwan Mapserver, MapDotNet Server • MapGuide Enterprise (mediante FDO) • ESRI ArcGIS Server 9.3+ |
| <ul style="list-style-type: none"> • Cliente de Escritorio S.I.G. <ul style="list-style-type: none"> • uDig, QGIS, mezoGIS • OpenJUMP • OpenEV • SharpMap SDK para Microsoft.NET 2.0 • ZigGIS para ArcGIS/ArcObjects.NET • GvSIG, GRASS | <ul style="list-style-type: none"> • Cliente de Escritorio S.I.G. <ul style="list-style-type: none"> • Cadcorp SIS • Microimages TNTmips GIS • ESRI ArcGIS 9.3+ • Manifold • GeoConcept • MapInfo (v10) • AutoCAD Map 3D (using FDO) |

Tabla 11: Aplicaciones soportadas por PostGIS.

Fuente web: *Introduction to PostGIS*, Recuperado de: <http://workshops.boundlessgeo.com/postgis-intro/introduction.html>, 2013.

3.7.6. Arquitectura

PostGIS obtiene todas las características de PostgreSQL, por lo tanto, también la arquitectura de PostGIS es del tipo cliente – servidor.



Gráfica 21: Arquitectura de PostGIS

Fuente: Propia.

A continuación una matriz comparativa entre PostGIS con la extensión de MySQL.

| Parámetro comparación | PostGIS | MySQL Spatial Extensions |
|----------------------------|---|--|
| Referencia | Boston Geographic Information Systems. | Boston Geographic Information System. |
| Licencia | GPL Open Source. | Comercial Open Source (COSS). |
| Plataforma. | Windows, Linux, Unix, Mac. | Windows, Linux, Unix, Mac. |
| País de origen | Estados Unidos | Suecia |
| Lenguaje desarrollo | C | C, C++ |
| Componentes Metadatos | Si | Si |
| Clientes S.I.G. libres | OpenJump, QGIS, gvSIG, uDig, GRASS. | GvSIG |
| Clientes S.I.G. privativos | ESRI ArGIS 9.x Server, ZigGIS, MapInfo 10+, AutoCAD FDO. | FME |
| Mapeo Web | ArGIS, UMN MapServer, GeoServer MapGuide, Manifold, Feature Server, MapDotNet. | UMN Mapserver, Geoserver, MapGuide. |
| Funciones Espaciales | Más de 800 funciones y operadores. | Pocas funciones espaciales |
| Tipos que soporta. | Geometry | Geography |
| | Poligon, Point, LineString, MultiPoint, MultiPoligon, MultiLineString, GeometryCollection, CircularString, CompoundCurve, CurvePoligon, MultiCurve, MultiSurface. | Soporte OGC Point, LineString, MultiPoint, MultiPoligon, MultiLineString, GeometryCollection. |

Tabla 12: Comparativa entre PostGIS y MySQL Spatial Extensions.

Fuente: Propia.

3.8. PMAPPER.

3.8.1. Introducción.

Los clientes livianos son proyectos para acceder a la información geoespacial en ambientes web. Este tipo de software se utiliza para construir sencillos visores web hasta completos geo portales, e incluso herramientas de edición vía web.

p.mapper Pmapper es un framework basado en MapServer PHP/MapScript desarrollado por Argim Burger, para la publicación de mapas, dispone de muchas funcionalidades y múltiples configuraciones que facilitan la personalización de aplicaciones de MapServer. Es software libre, se redistribuye o modifica bajo los términos de la GNU/GPL.

3.8.2. Características.

El framework Pmapper cuenta con las siguientes características:

- ✓ DHTML (DOM) Interfaz zoom/pan
- ✓ Navegadores soportados: Mozilla 2, Netscape 7, IE 5/7, Opera 6.
- ✓ Zoom/pan via teclas de teclado, rueda del ratón, mapa de referencia.
- ✓ Funciones de consulta (identificar, seleccionar, buscar).
- ✓ Ver resultados de consultas con enlaces de bases de datos e hipervínculos.
- ✓ Configuración de visualización de consultas vía JavaScript
- ✓ Funciones de impresión: HTML y PDF. Función para medir distancias y áreas
- ✓ Varios tipos de visualización de capas y leyendas.
- ✓ Descarga de imágenes de mapas visualizados en varios formatos y resoluciones.

3.8.3. Estructura de archivos.

La estructura de los directorios de p.mapper es la siguiente:

- ✓ Config: Directorio de los archivos de configuración.
- ✓ Images: Contiene todas las imágenes utilizadas en la interfaz de usuario.
- ✓ Incphp: Directorio que contiene los archivos PHP con las funciones generales.
- ✓ Javascript: Contiene los archivos JavaScript de la aplicación.
- ✓ Plugins: Contiene plugins especiales que dan funcionalidades adicionales.
- ✓ Templates: Contiene las plantillas css de la aplicación.

3.8.4. Cuadros comparativos.

| Cliente | Licencia | Desarrollo | Documentación | Descripción |
|-------------|-------------|---------------|--|---|
| Mapbender | GNU GPL | CCGIS | Idiomas: alemán, Inglés Formatos: PDF Niveles: Usuarios | Framework. Provee interfaz de administración para gestión de seguridad, usuarios con una BD en MySQL o PostgreSQL. |
| Mapfish | GNU GPL v 3 | Camptocamp SA | Idiomas: Francés. Formatos: Html Niveles: Usuarios | Framework. Basado en Pylons. A lado del cliente utiliza y extiende OpenLayers, GeoExt y ExtJS. |
| Pmapper | GNU GPL | Armin Burger | Idiomas: Inglés Formatos: PDF, Trac Niveles: Usuarios | Framework. Basado en MapServer y PHP/MapScript. Provee un conjunto de herramientas listas para usar. API de plugins para agregar funcionalidades. |
| Open Layers | BSD-style | Metacarta | Idiomas: Francés, Inglés, Portugués Formatos: Html. Niveles: Usuarios | No tiene dependencias en el servidor. Sirve de base para varios proyectos en la web. Edición en línea. |

Tabla 13: Comparación clientes ligeros web gis – Descripción General.

Fuente web: *Comparación clientes ligeros web*, Recuperado de: <http://geotux.tuxfamily.org/index.php/es/component/k2/item/291-comparacion-clientes-web-v6>, 2013.

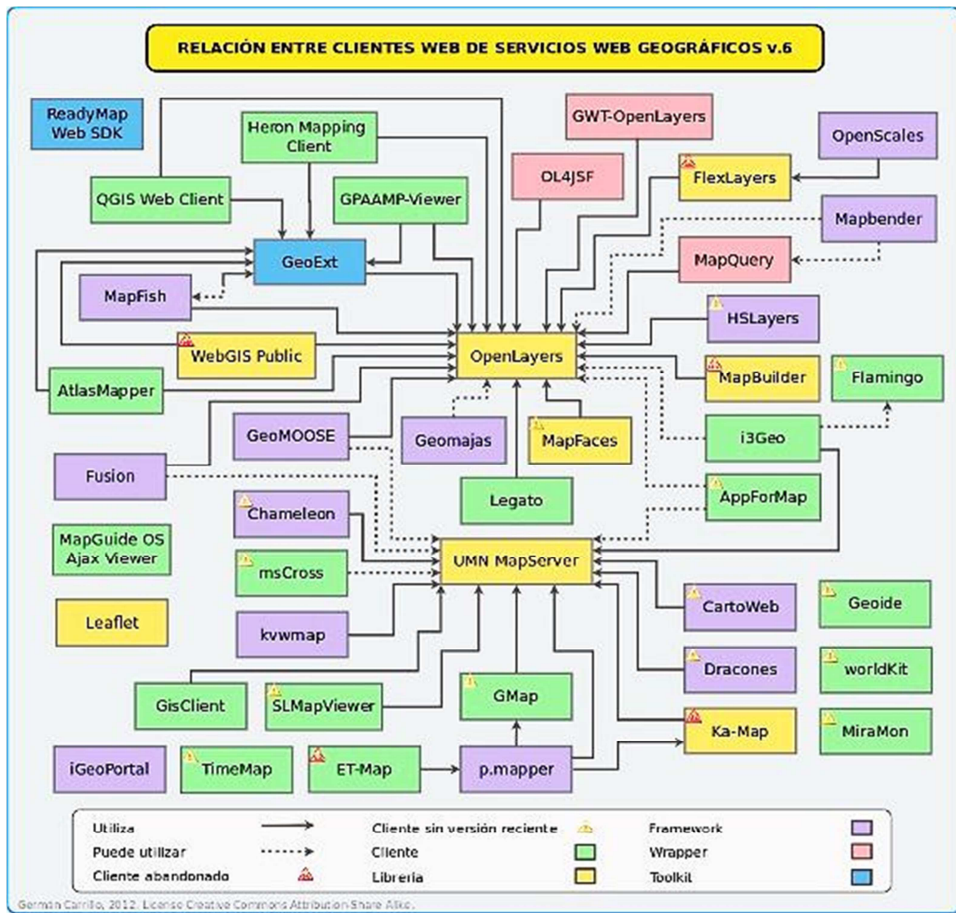
| Ciente | Lenguaje en el que está escrito | Lenguaje programación que admite | Servicios OGC que consume | Depende de servidor de mapas | plug-ins privados | ¿Incluye componente de metadatos? | Listas de correo |
|-------------|---------------------------------|----------------------------------|---------------------------|------------------------------|-------------------|-----------------------------------|------------------|
| Mapbender | Javascript; PHP | PHP | WMS, WFS, WFS-T | No | no | Si (ISO-19119) | Si (Usuarios) |
| Mapfish | Javascript; Python | Java, Javascript, PHP, Python | WMS, WFS | No | no | No | Si (Usuarios) |
| Pmapper | Javascript; PHP | Javascript; PHP | WMS; WFS | Si MapServer | no | No | Si (Usuarios) |
| Open Layers | Javascript | Javascript | WMS; WFS | No | No | No | Si |

Tabla 14: Características técnicas de clientes ligeros web SIG.

Fuente web: *Comparación clientes ligeros web*, Recuperado de: <http://geotux.tuxfamily.org/index.php/es/component/k2/item/291-comparacion-clientes-web-v6>, 2013.

3.8.5. Relación entre clientes web.

A continuación la gráfica de clientes de servicios web geográficos de software libre.



Gráfica 22: Relación entre clientes de servicios web.

Fuente web: *Comparación clientes ligeros web*, Recuperado de: <http://geotux.tuxfamily.org/index.php/es/component/k2/item/291-comparacion-clientes-web-v6>, 2013.

3.9. QUANTUM GIS

3.9.1. Introducción.



Quantum GIS (QGIS) es un sistema de información geográfica de código abierto. Este software surgió en mayo de 2002 y se instauró en junio del mismo año como un proyecto en SourceForge. Se ejecuta bajo plataformas GNU/Linux, Unix, Mac OS y Windows. Distribuido bajo la licencia GNU GPL, por lo que puede ser modificado libremente para que realice diferentes y más especializadas funciones.

3.9.2. Características.

QGIS proporciona varias funcionalidades SIG dotadas por los plugins que lo forman.

Mostrar datos: Obtener y superponer datos vectoriales y ráster en diferentes formatos y proyecciones, sin hacer una conversión a un formato interno o común.

Datos soportados:

- ✓ PostgreSQL para el uso de tablas PostGIS, formatos vectoriales con el soporte de la librería de instalación OGR, incluyendo archivos de formato ESRI,
- ✓ MapInfo, SDTS y GML.
- ✓ Ráster y soporte de formatos de imágenes por la librería GDAL, tales como GeoTiff, Erdas Img., ArcInfo Ascii Grid, JPEG, PNG.
- ✓ Base de datos Spatialite.
- ✓ GRASS ráster y los datos vectoriales de bases de datos de GRASS
- ✓ Datos en línea como OGC-compliant en WMS o WFS.
- ✓ Datos OpenStreetMap.

Explora los datos y componer mapas: Puede diseñar mapas y explorar interactivamente los datos espaciales en la interfaz de usuario. Entre las herramientas útiles de la GUI se encuentran:

- ✓ Proyección de vuelo.
- ✓ Diseño de mapas.
- ✓ Panel de descripción.
- ✓ Marcadores espaciales.

- ✓ Identificación, selección de características.
- ✓ Editar y ver atributos de búsqueda.
- ✓ Características de etiquetado.
- ✓ Cambio de simbología vectorial y ráster.
- ✓ Añadir una capa de cuadrícula.
- ✓ Decorar el mapa con una flecha norte, escala y etiqueta copyright.
- ✓ Guardar y restaurar proyectos.

Crear, editar, manejar y exportar datos: Puede crear, editar, manejar y exportar mapas vectoriales en varios formatos. QGIS ofrece lo siguiente:

- ✓ Herramientas de digitalización, soporte a formatos OGR y capas vectoriales.
- ✓ Crear y editar shapefiles y capas vectoriales GRASS.
- ✓ Herramientas GPS para importar y exportar en formato GPX, y convertir otros formatos GPS a GPX, bajar y subir directamente a una unidad GPS.
- ✓ Visualizar y editar los datos OpenStreetMap.
- ✓ Crear capas PostGIS desde shapefiles con el plugin SPIT.
- ✓ Manejo mejorado de tablas PostGIS.
- ✓ Manejo de atributos de tablas vectoriales.
- ✓ Guardar capturas de pantalla como imágenes geo referenciadas.

Analizar los datos: Puede realizar análisis de datos espaciales de PostGIS y otros formatos admitidos por OGR usando el plugin de Python fTools. Ofrece análisis vectorial, muestreo, geo procesamiento, geometría y herramientas de gestión de bases de datos. También puede utilizar las herramientas de GRASS integradas, que incluyen la funcionalidad completa de GRASS con más de 400 módulos.

Publicar mapas en internet: Puede ser utilizado para exportar datos a un archivo de mapa y publicarlos en el internet usando el servidor web UMN MapServer. También se pueden utilizar como un cliente WMS, WMS-C o CMA y CMA-T, y como WMS o WFS.

3.9.3. Principales plugins.

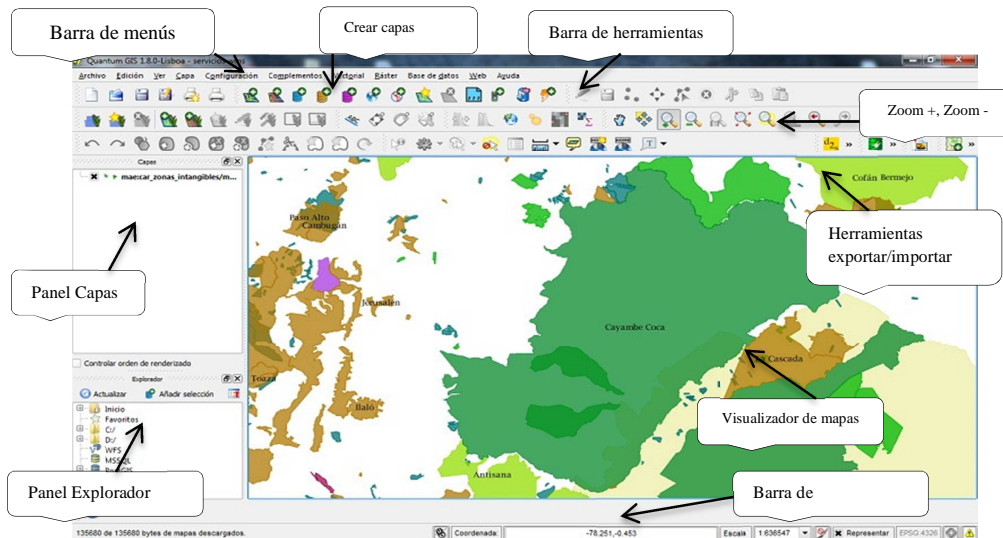
QGIS adquiere su funcionalidad de acuerdo a los plugins que tenga instalado.

| Plugin | Descripción |
|--------------------------|--|
| Add Delimited Text Layer | Carga y muestra los archivos de texto delimitados que tienen coordenadas x, y. |
| Coordinate Capture | Captura de coordenadas con el mouse. |
| DB Manager | Editar y visualizar capas y tablas, ejecutar consultas SQL. |
| Diagram Overlay | Colocar diagramas de capas vectoriales. |
| Dxf2Shp Converter | Convertidor DXF. |
| GPS Tools | Carga e importación de datos GPS. |
| GRASS | Integración GIS GRASS. |
| GDALTools | Integrar herramientas GDAL en QGIS. |
| Georeferencer GDAL | Agregar de información de la proyección ráster usando GDAL. |
| Heatmap tool | Generación de mapas ráster de calor a partir de datos puntuales. |
| Interpolation plugin | Interpolación basada en vértices de una capa vector. |
| Mapserver Export | Exportar archivo de proyecto QGIS a un mapa de MapServer. |
| Offline Editing | Permitir la edición offline y la sincronización con la base de datos. |
| OpenStreetMap plugin | Visor y editor de datos de OpenStreetMap. |
| Oracle Spatial GeoRaster | Soporte Geo Ráster. |
| Plugin Installer | Descargar e instalar plugins. |
| Raster terrain analysis | Análisis del terreno basado en la trama. |
| Road graph plugin | Análisis la ruta más corta de la red. |
| SPIT | Importar Shapefile a PostgreSQL / PostGIS. |
| SQL Anywhere Plugin | Almacenar capas vectoriales dentro de la Base de datos SQL. |
| Zonal statistics plugin | Calcular suma, media de trama para cada polígono de una capa vector. |
| Spatial Query plugin | Hace consultas espaciales en capas vectoriales. |
| eVIS | Herramientas de visualización de eventos. |
| fTools | Herramientas para el análisis y gestión de datos vectoriales. |

Tabla 15: Plugins principales de Quantum GIS 1.8.

Fuente: Propia.

3.9.4. GUI de Quantum GIS



Gráfica 23: GUI de Quantum GIS

Fuente: Propia.

3.9.5. Exportar mapa a MapServer.

Quantum GIS incluye la función de exportar mapas generados ya sea por archivos .shp, tablas de alguna base de datos o servicio web. Para esto es necesario contar con el plugin de Mapserver Export. Lo cual generará un archivo .map que posteriormente puede ser utilizado en el servidor de mapas MapServer y el cliente ligero pmapper.

Criterio de selección de tecnologías y herramientas de software a utilizar.

Elegir plataformas o tecnologías es un paso esencial cuando se inicia un proyecto desde cero. Cometer errores en las decisiones tomadas puede salir caro. Todos los cambios que se tenga que hacer a posterior van a tener su costo tanto en lo económico como en el tiempo. Para realizar el estudio de las herramientas de software de desarrollo y tecnologías, se evalúan varios elementos: Plataforma de desarrollo, lenguaje de programación, framework, otros lenguajes y estándares de desarrollo y la herramienta de la base de datos. Para la selección de las herramientas se tomaron los siguientes criterios:

- ✓ La modalidad de software libre.
- ✓ Funcionalidad, dependencia e integración.
- ✓ La documentación disponible.
- ✓ Conocimientos previos adquiridos.

| ÍTEM | Tecnología/Plataforma | MOTIVO |
|---------------------------------|------------------------------|--|
| Lenguaje de Programación | PHP, HTML, Javascript, XML | Por la facilidad de codificación. |
| Plataforma de desarrollo | Netbeans 7.0.1 | Este IDE viene integrado con el framework Symfony. |
| Framework de desarrollo | Symfony 1.4.x | Por agilizar el desarrollo debido a su Patrón de diseño MVC. |
| Bases de Datos | PostgreSQL | Esta BDD es potente y confiable y además por ser de acceso libre. |
| Base de dato Espacial | PostGIS | Versión espacial compatible. |
| Servidor de Mapas | MapServer | Acceso a BDD y archivos .shp. |
| Ciente Ligero GIS | Pmapper | Ciente ligero compatible con servidor de mapa. |
| Ciente Pesado GIS | Quantum GIS | Permite crear mapas .map para utilizar en el cliente ligero pmapper. |

Tabla 16: Definición de tecnologías y herramientas.

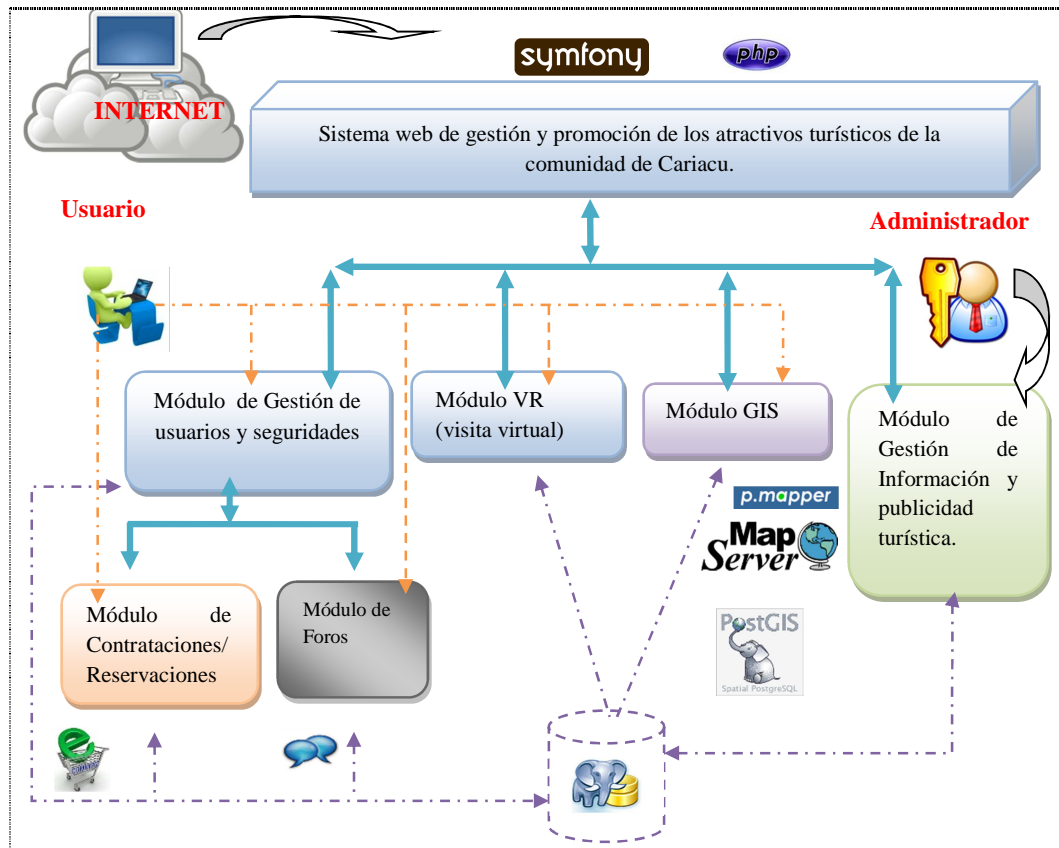
Fuente: Propia.

3.10. PLANTEAMIENTO DEL DISEÑO ARQUITECTÓNICO DEL SISTEMA.

Los sistemas siempre se descomponen en subsistemas o módulos que suministran algún conjunto relacionado de servicios. El proceso de diseño inicial para identificar estos subsistemas y establecer un marco de trabajo para el control y comunicación de los subsistemas se llama diseño arquitectónico y lo que produce este proceso de diseño es una descripción de la arquitectura de software. El diseño arquitectónico es a menudo el punto inicial para la especificación de diversas partes del sistema.

El proceso de diseño arquitectónico comprende el establecimiento de un marco de trabajo estructural básico para un sistema. Esto implica identificar los componentes

principales del sistema y la comunicación entre ellos. Dentro de este aspecto se plantea el siguiente diseño arquitectónico del aplicativo web:



Gráfica 24: Diseño arquitectónico del sistema.

Fuente: Propia.

3.11. APORTE DE GIS AL TURISMO.

En la rama del turístico los GIS también aportan grandes beneficios, tanto para los que ofrecen los servicios como para los visitantes, la información que ofrecen los GIS son de gran aporte para la buena toma de decisión, son fundamentales porque permiten contar con datos de las poblaciones, de las zonas permitidas y todo aquello que sirva para consolidar un negocio y ofrecer un buen servicio tanto para los pobladores como para los visitantes.

La tecnología GIS brinda grandes oportunidades para la creación de modernas y dinámicas aplicaciones turísticas que utilicen mapas web. En la actualidad se puede apreciar varias empresas que utilizan los GIS como un medio de difusión de su actividad turística.

| Empresa | Descripción | Sitio Web |
|---|---|---|
| Crown of the Continent Catalunya | Sitio donde se puede planear viajes, y visualizar los sitios turísticos en un mapa. | http://www.crownofthecontinent.net/map.php |
| Municipio de Gualeguaychu | Contiene los recursos y servicios turísticos de Cataluña desde el principal repositorio de datos turísticos geo referenciados de la Administración. | http://www.catalunya.com/en-el-mapa |
| Turismo de Córdoba | Sitio de Municipio de Gualeguaychú – Argentina. | http://www.geomat-maps.com.ar/mgchu/map.phtml?me=5629346.8,6343437.8,5641792.8,6351200.7&config=default |
| Alcandía Mayor de Cartagena | El aplicativo está basado en tecnología SIG que permite posicionar ubicaciones y realizar consultas interactivas. | http://www.e-sig.info/cordobaturismo/ |
| | Sitio web que contiene mapas Gis de la Alcandía de Cartagena – Colombia | http://midas.cartagena.gov.co/ |

Tabla 17: Uso de tecnología GIS en el turismo.

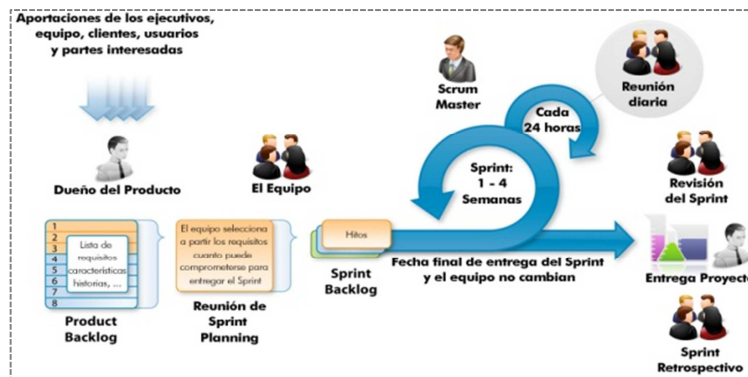
Fuente: Propia.

3.12. METODOLOGÍA SCRUM

3.12.1. Introducción a la metodología.

La metodología Scrum es sencilla, pero no es fácil de establecer, debido a que su funcionamiento no se centra en la ejecución de procesos y seguimiento de un plan, sino en un ajuste continuo a las circunstancias de la evolución del proyecto.

Dicho esto, el presente proyecto se adapta a estos requerimientos con la finalidad de mantener un control continuo del proceso de desarrollo del aplicativo. Se inicia con una visión general del aplicativo, y a partir de esta se especifica y da detalle a las funcionalidades o partes que tienen mayor prioridad, y que pueden llevarse a cabo en un período de tiempo breve que en este caso será entre 20 a 31 días. Cada iteración finaliza con la entrega de un módulo del sistema. Se establecen reuniones diarias breves donde el desarrollador junto al dueño del software (Grupo de ecoturismo) revisa el trabajo realizado el día anterior y lo que se plantea para el siguiente. Esta reunión diaria se estableció en 15 minutos máximo, se realiza en las instalaciones de grupo turístico con el uso de una pizarra con información de las tareas realizadas y a realizar.



Gráfica 25: Proceso metodología SCRUM.

Fuente web: *Metodología Scrum*, Recuperado de:

<http://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM>, 2013.

3.12.2. Gestión de la evolución del proyecto.

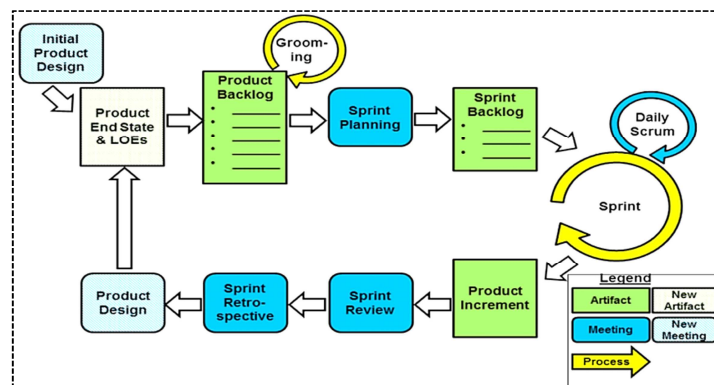
Scrum maneja de forma real la evolución del proyecto con las siguientes tácticas:

Revisión de las Iteraciones: Al culminar cada iteración se realiza una revisión funcional del módulo desarrollado, con los implicados en el proyecto (Desarrollador, Grupo de ecoturismo). Por tal motivo las correcciones o aclaraciones se realizan en el transcurso de la iteración.

Desarrollo incremental: El desarrollo incremental implica que al final de cada iteración se cuenta con un módulo totalmente operativo, que se puede usar y evaluar.

Auto-organización: El equipo de trabajo es auto-organizado, por lo cual se puede establecer un margen de maniobra para tomar las decisiones.

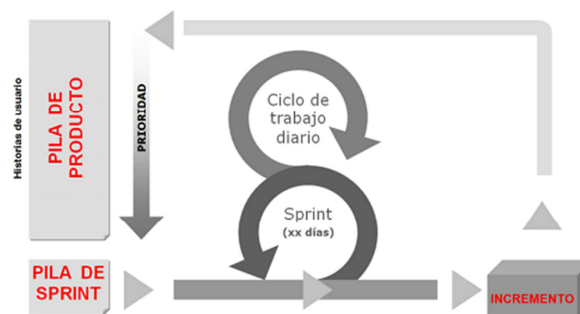
Colaboración: Este componente es muy importante y necesario, ya que todos los miembros del equipo colaboran entre sí, según las capacidades y no por su cargo.



Gráfica 26: Marco Scrum estándar.

Fuente web: *Marco estándar SCRUM*, Recuperado de: <http://www.scrummanager.net/blog/2014/05/t-scrum/>, 2013.

3.12.3. Elementos de Scrum:



Gráfica 27: Diagrama de ciclo iterativo de SCRUM.

Fuente: Propia.

Pila del producto: los requisitos del cliente.

La pila del producto es una lista de funcionalidades que se generan luego de establecer las necesidades y requerimientos del cliente (Grupo de ecoturismo). Representa todo lo que el cliente manifestó en la reunión de definición de requerimientos.

Pila del Sprint

La pila del sprint es la descomposición de las funcionalidades enlistadas en la pila de producto en las tareas más pequeñas y fáciles de dar seguimiento. Esta pila es la que posteriormente se sigue para el desarrollo de los módulos del sistema turístico.

El incremento.

En este caso el incremento es cada módulo que se desarrollará en la iteración. Algo muy importante es que para que el módulo sea considerado como un incremento debe estar completamente terminado y funcional.

3.12.4. Reuniones de Scrum.

Planificación del sprint. Cada cierto tiempo se realizan las reuniones de planificación, en este proyecto las reuniones se establecieron al final de cada iteración donde se hace la presentación del módulo desarrollado y se establece los requerimientos del siguiente módulo a implementar. Esta reunión la dirige el Scrum Master a la cual asisten el propietario del producto (Grupo de ecoturismo) y el desarrollador

Revisión diaria: seguimiento del sprint. Se establece un lapso de tiempo que no exceda los 15 minutos, para realizar la reunión diaria en las instalaciones del cliente (Grupo ecoturismo) en la que el desarrollador informa al resto de asistentes sobre las tareas que está realizando, los problemas que encontró, y actualiza en la pila del sprint las tareas terminadas.

Revisión del Sprint: Fin de Sprint. Esta reunión es informativa y se realiza al final del sprint, con una duración máxima de 2 horas, las cuales se realizan en las instalaciones del cliente en horas de la tarde, en la cual el desarrollador presenta al propietario del producto y usuarios, el módulo desarrollado en el sprint. El objetivo es observar el módulo: terminado y probado.

Retrospectiva. Al mismo tiempo que se realiza la reunión de fin de sprint se aprovecha para hacer un análisis del estado actual del proyecto con el fin de mantener una mejora continua de la agilidad de la institución y del proyecto.

3.12.5. Roles de Scrum.

Las personas involucradas de forma directa o indirecta con el desarrollo del proyecto, se clasifican en dos grupos: los comprometidos y los implicados. Scrum llama a los primeros *cerdos* y a los segundos *gallinas*. El origen de estos nombres surge de la siguiente metáfora que diferencia entre **compromiso** e **implicación** con el proyecto:

- Una gallina y un cerdo paseaban por la carretera. La gallina preguntó al cerdo: “¿Quieres abrir un restaurante conmigo?”.
- El cerdo consideró la propuesta y respondió: “Sí, me gustaría. ¿Y cómo lo llamaríamos?”.
- La gallina respondió: “jamón con huevos”.
- El cerdo se detuvo, hizo una pausa y contestó: “pensándolo mejor, creo que no voy a abrir un restaurante contigo.
- Yo estaría realmente comprometido, mientras que tu estarías sólo implicada”.

| COMPROMETIDOS (CERDOS) | IMPLICADOS (GALLINAS) |
|---------------------------------------|--|
| Propietario del Producto. | Otros interesados (Dirección, Gerencias, etc.) |
| Miembros del equipo. Desarrollador | Scrum Master. |

Tabla 18: Roles de Scrum.

Fuente: Propia.

El propietario del producto. El propietario o dueño del producto o *product owner* es el “grupo de Ecoturismo de la comunidad de Cariacu”. Una sola persona debe encargarse de la gestión el producto por lo cual se establece que debe ser el presidente de dicha organización, ya que él tendrá la última palabra en cómo será el producto final, y el orden de desarrollando de los sucesivos módulos. Además se compromete en la gestión para la financiación del proyecto.

El equipo. Es el grupo de personas que desarrollan el producto de manera conjunta. En este caso el equipo está distribuido entre el desarrollador y el resto de miembros del grupo de ecoturismo.

Scrum master. Es la persona responsable del funcionamiento de la metodología Scrum en el proyecto. Además es el encargado de asegurar que el resto del equipo no tenga problemas en cumplir con sus funciones y tareas. En otras palabras, este rol ayuda al equipo a mantenerse activo y productivo.

3.12.6. Fases de Scrum

Fase de planificación. En esta fase se define todo el trabajo a realizar descrito por el cliente en las historias de usuarios, se trasladan a la pila de producto donde se priorizan las actividades y en función a los requerimientos del cliente se eligen las tareas a incluir en la pila de Sprint.

Fase de desarrollo: es la parte ágil, donde el sistema se desarrolla en Sprints.

Fase de finalización: incluye integración, pruebas de aceptación y documentación. Indica la finalización de todos los requerimientos establecidos, quedando la pila de producto vacía.

CAPÍTULO IV: IMPLEMENTACIÓN

En este capítulo se procede a desarrollar el aplicativo, siguiendo las especificaciones de la metodología de desarrollo de software SCRUM.

Primeramente se inicia con la propuesta del proyecto de software, para luego asignar los roles a las personas involucradas, posteriormente se realiza la primera reunión para realizar el análisis de requerimientos y funcionalidades del aplicativo.

La metodología plantea un desarrollo iterativo con 3 fases; la primera fase es la de PLANIFICACIÓN, en esta fase se realiza un análisis global, en donde se plantea los requerimientos y funcionalidades del aplicativo, todo esto recopilado en artefactos como actas de reuniones, historias de usuarios, pila de producto y pila de sprint.

La segunda fase es la de DESARROLLO, esta es de diseño, implementación y pruebas de lo planteado en la fase anterior, se generan los incrementos o entregables.

La tercera fase es la de FINALIZACIÓN, esta es para las pruebas de aceptación y liberación o integración del incremento, esto recopilado en las actas de entrega - recepción.

La etapa FINAL es subir a entorno de producción todo el aplicativo, completamente probado, se genera el acta de entrega - recepción del proyecto.

Es indispensable mencionar que el aplicativo es realizado en SPRINTs o iteraciones, en cada uno de los cuales se construye un módulo del aplicativo.

4.1. Proyecto de software.

El proyecto de software “Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu con la utilización de software libre”, nace de la necesidad de dar a conocer la actividad turística de la comunidad de Cariacu.

A continuación se muestra un extracto del proyecto de software.

| PROYECTO INFORMÁTICO | | | |
|--|---------------------------------------|--|-------------|
| TÍTULO | | | |
| Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu con la utilización de software libre. | | | |
| OBJETIVO | | | |
| Implementar una aplicación web dinámica para la gestión y promoción de las riquezas turísticas de las comunidades indígenas del cantón Cayambe, principalmente de la comunidad de Cariacu. | | | |
| JUSTIFICACION | | | |
| La comunidad de Cariacu está ubicada al noreste del cantón Cayambe en la provincia de Pichincha, esta es una zona dedicada a la agricultura, ganadería, apicultura, hoy en día esta comunidad ha visto adecuada la explotación sustentable de sus recursos naturales y culturales a través del turismo comunitario, pero la difusión de esta actividad es ineficiente debido a que la información necesaria no se encuentra disponible para las personas que se interesen en visitarlos. | | | |
| DESCRIPCIÓN | | | |
| De manera general el aplicativo permite gestionar la actividad turística, a través de sus diferentes módulos. Gestión de usuarios, Gestión de la información, Foros de discusión, Visitas virtuales, Catálogo de productos y contrataciones, GIS. | | | |
| ALCANCE | | | |
| El sistema estará publicado en la web lo cual permitirá brindar la información en cualquier parte del mundo a toda persona con acceso a internet y con deseos de vivir una experiencia nueva. | | | |
| BENEFICIARIOS | | | |
| Grupo de Ecoturismo, Grupos organizados de mujeres, Grupos artísticos de la comunidad, Grupo de jóvenes, Pequeños y medianos empresarios, La comunidad en general y los turistas interesados en visitar la localidad | | | |
| SOPORTE TECNOLÓGICO | | | |
| El aplicativo será desarrollado con el patrón de diseño Modelo, Vista, Controlador, para lo cual se utilizará: lenguaje de programación PHP 5, HTML, CSS, JavaScript, XML, con base de datos PostgreSQL 9 y la versión espacial PostGIS, servidor de mapas MapServer, cliente ligero Pmapper y framework Symfony 1.4.16. | | | |
| RESPONSABLE PROYECTO | | | |
| Nombre Responsable | | Cargo / Función | |
| Grupo ecoturismo Pakchapi Pakari | | Dueño del aplicativo | |
| Sr. Germánico Guacán | | Presidente grupo "Pakchapi Pakari" – representante | |
| Sr. Carlos Caluguillín | | Tesisista – programador | |
| PLANIFICACION | | | |
| Nro. | Actividad | Fecha Inicio | Fecha Final |
| 1 | Asignación de roles | 30/06/2013 | 30/06/2013 |
| 2 | Análisis de requerimientos | 30/06/2013 | 01/07/2013 |
| 3 | Sprint 0: Módulo visita virtual | 02/07/2013 | 21/07/2013 |
| 4 | Sprint 1: Módulo gestión usuarios | 22/07/2013 | 10/08/2013 |
| 5 | Sprint 2: Módulo foro | 11/08/2013 | 01/09/2013 |
| 6 | Sprint 3: Módulo contrataciones | 02/09/2013 | 21/09/2013 |
| 7 | Sprint 4: Módulo gestión información. | 22/09/2013 | 22/10/2013 |
| 8 | Sprint 5: Módulo GIS | 23/10/2013 | 23/11/2013 |
| 9 | Integración – Entrega | 24/11/2013 | 10/12/2013 |
| 10 | Capacitación | | |

Tabla 19: Proyecto de software.

Fuente: Propia

Para iniciar con el proyecto se realizaron reuniones con el grupo de ecoturismo, para socializar el proyecto, definir roles y definir requerimientos del sistema.

4.2. Asignación de roles.

La metodología asigna los roles a cada persona involucrada en el desarrollo del software, a continuación se muestra las personas involucradas en el proyecto y su rol en el mismo.

| Nombre | Cargo | Función |
|--------------------------------------|---|--|
| Sr. Germánico Guacán | Propietario del producto (Product Owner). | Representante del grupo de ecoturismo. |
| Sr. Carlos Caluguillín. | Equipo de SCRUM. | Programador. |
| Sr. Carlos Caluguillín. | SCRUM Manager. | Cumplir metodología. |
| Comunidad de Cariacu. | Involucrado. | Facilita información. |
| Dirección de turismo cantón Cayambe. | Involucrado. | Promociona actividad turística en el cantón. |

Tabla 20: Asignación de roles SCRUM.
Fuente: Propia

4.3. Análisis inicial de requerimientos.

El proyecto busca la gestión de la publicidad de la actividad turística que realiza el grupo de ecoturismo de la comunidad de Cariacu, para lo cual requiere de una aplicación informática que esté al alcance de cualquier persona a través del internet, por tal motivo, se realiza el análisis de los requerimientos y funcionalidades del aplicativo. Fruto de ello se plantean los siguientes componentes:

| Módulo | Descripción / Función |
|---|---|
| VR (Visita virtual) | Visitas virtuales a los atractivos turísticos, a través de imágenes panorámicas 360°. |
| Gestión de usuarios y seguridad | Acceso a varios módulos con privilegios. |
| Foros | Temas de discusión y comentarios sobre la actividad turística. |
| Contrataciones y reservaciones | Catálogo de productos y servicios, compra en línea. |
| Gestión de la información y publicidad turística. | Gestionar la información turística y promocionarla desde el aplicativo. |
| GIS | Visualizar principales sitios turísticos en un GIS. |

Tabla 21: Módulos del sistema.
Fuente: Propia.

| Requerimiento | Ítem | Observación |
|------------------------------------|--|--|
| Lenguaje Programación. | PHP 5, Html, Css, Javascript, Xml y otros. | Por ser una aplicación web. |
| Framework Desarrollo. | Symfony 1.4.16 | Para codificación estructurada y rápida del aplicativo |
| Acceso a Datos. | Base de datos PostgreSQL y PostGIS. | Para acceso a datos y GIS |
| Sistema de Información Geográfico. | MapServer, Pmapper, Quantum GIS | Para gestionar el GIS del aplicativo. |

Tabla 22: Requerimientos tecnológicos del aplicativo.
Fuente: Propia.

Es importante recalcar que los requerimientos son seleccionados bajo la premisa de software libre, lo cual facilita en gran medida la creación del sistema. Luego del análisis y definición de componentes y requerimientos se procede al desarrollo del proyecto, para lo cual se da paso al primer sprint.

4.4. SPRINT 0: Desarrollo de módulo de visita virtual.

El primer Sprint tiene por objetivo el desarrollo del módulo de VR (Visita virtual), que permitirá la visualización de imágenes panorámicas en 360° de los sitios turísticos de las comunidades.

Planificación.

✓ Historias de usuario.

| HISTORIA DE USUARIO Nro. 1 | | | |
|---|--|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu con la utilización de software libre. | | |
| Nombre de la historia: | Diseño de Base de Datos. | | |
| ID de historia: | 1 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 1 | Dependencia: | Ninguna. |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 01 – 07 – 2013 |
| Descripción | | | |
| Se debe crear la base de datos para el sistema | | | |
| Crear las tablas para el módulo de visitas virtuales. | | | |
| Pruebas de Aceptación | | | |
| La base de datos se creará en PostgreSQL. | | | |
| Las Tablas deben de estar normalizadas y deben de tener todos los campos necesarios. | | | |
| Debe haber integridad referencial entre los datos. | | | |
| Permitir ingresar, actualizar, eliminar y seleccionar datos sin ningún inconveniente. | | | |
| Observaciones | | | |
| Normalizar a lo más apto posible. | | | |

Tabla 23: Historia de usuario N° 1.

Fuente: Propia.

| HISTORIA DE USUARIO Nro. 2 | | | |
|---|--|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu con la utilización de software libre. | | |
| Nombre de la historia: | Análisis y desarrollo del Módulo de visita virtual | | |
| ID de historia: | 2 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 1 | Dependencia: | Ninguna. |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 01 – 07 – 2013 |
| Descripción | | | |
| Se debe desarrollar el módulo de visita virtual, donde se describirán los atractivos turísticos de la zona. | | | |
| Se podrá visualizar los sitios turísticos de la comunidad desde un navegador. | | | |
| Las visitas deben realizarse con fotografías panorámicas en 360°. | | | |
| Debe permitir crear, modificar, eliminar las visitas virtuales. | | | |
| Pruebas de Aceptación | | | |
| La visita virtual debe ser idéntica a la realidad. | | | |
| Debe permitir navegar por todos los sitios. | | | |
| Funcionalidad al visualizar, crear, modificar y eliminar. | | | |
| Observaciones | | | |
| Para crear, modificar, actualizar y eliminar las visitas virtuales se realizarán desde el módulo de gestión de información desde la parte administrativa. | | | |

Tabla 24: Historia de usuario N° 2.

Fuente: Propia.

✓ Pila de producto.

Una vez recopiladas las Historias de Usuarios se procede a generar la pila del producto, en la cual se describen las funcionalidades a realizar para el desarrollo del módulo VR.

| PILA DE PRODUCTO | | |
|------------------|---|-----------|
| ID | Nombre Historia de usuario | Prioridad |
| 1 | Diseño de base de datos. | ALTA |
| 2 | Análisis y desarrollo del módulo de visita virtual. | ALTA |

Tabla 25: Pila de producto N°1.

Fuente: Propia.

✓ **Pila de Sprint.**

Seguidamente se dividen las funcionalidades de la pila de producto en tareas pequeñas para facilitar su desarrollo.

| PILA DE SPRINT | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|-----------------------------------|--|--------------|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| ID | Tareas | Tipo | Estado | Esfuerzo estimado (horas). | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Diseño de Base de Datos | | | | | | | | | | | | | | | | | | | |
| 1 | Creación BDD. | Codificación | Terminada | 1 | | | | | | | | | | | | | | | | | | | |
| 2 | Análisis de tablas | Análisis | Terminada | 1 | | | | | | | | | | | | | | | | | | | |
| 3 | Creación de tablas. | Codificación | Terminada | 1 | | | | | | | | | | | | | | | | | | | |
| 4 | Datos de prueba. | Prueba | Terminada | | 1 | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Análisis y desarrollo del módulo de visitas virtuales. | | | | | | | | | | | | | | | | | | | |
| 5 | Análisis de requerimientos | Análisis | Terminada | | 2 | 1 | | | | | | | | | | | | | | | | | |
| 6 | Estudio de herramientas | Análisis | Terminada | | 4 | 2 | 3 | 3 | 2 | | | | | | | | | | | | | | |
| 7 | Crear proyecto web-aplicación frontend | Codificación | Terminada | | | 1 | | | | | | | | | | | | | | | | | |
| 8 | Configurar Symfony en aplicación | Codificación | Terminada | | | 1 | | | | | | | | | | | | | | | | | |
| 9 | Creación de módulo VR | Codificación | Terminada | | | 1 | | | | | | | | | | | | | | | | | |
| 10 | Creación de plantillas | Codificación | Terminada | | | 3 | 2 | 1 | 1 | 1 | | | | | | | | | | | | | |
| 11 | Creación de visitas virtuales | Codificación | Terminada | | | 2 | 3 | 3 | 4 | 5 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 |
| 12 | Pruebas funcionalidad | Prueba | Terminada | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 13 | Pruebas de aceptación. | Prueba | Terminada | | | | | | | | | | | | | | | | | | | | 2 |
| 14 | Integración | Codificación | Terminada | | | | | | | | | | | | | | | | | | | | 1 |

Tabla 26: Pila de sprint N° 1.

Fuente: Propia.

Desarrollo.

Definidas las tareas, se inicia el proceso de desarrollo y codificación del módulo.

Requerimientos.

- ❖ **Instalar servidor web apache.** Apache se encuentra disponible para todas las plataformas en su página oficial <http://www.apache.org>, de donde se la puede descargar e instalar. En Linux se puede instalar a través de líneas de comando: `$apt-get install apache2`. Además es necesario instalar soporte para php 5: `$apt-get install php5 libapache-mod-php5 php5-cli php5-pear php5-pdo php5-pgsql php5-gd php5-mbstring php5-mcrypt php5-openssl php5-apc php5-curl`.
- ❖ **Instalar Base de datos PostgreSQL.** PostgreSQL está disponible para descargar en su página oficial <http://www.postgresql.org/download/>.

Instalar en linux:

Instalar: `$ apt-get install postgresql postgresql-client`

Cambiar la contraseña del usuario: `$ passwd postgres`

- ❖ **Descargar e instalar Netbeans.** Netbeans está disponible en sus diferentes versiones en la página oficial: <https://netbeans.org/downloads/>, de donde se puede descargar e instalar en cualquier plataforma, previamente es necesario instalar la máquina virtual de java, que está disponible para todas las plataformas en su sitio oficial: <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>.
- ❖ **Descargar framework Symfony.** Symfony está disponible en su sitio oficial <http://symfony.com/download>. La versión necesaria es la 1.4, en la actualidad se encuentra en la versión 2.3.7

Tarea 1: Creación de base de datos. La primera tarea es crear la base de datos para la aplicación, para lo cual se ejecutan los siguientes comandos en Linux:

Crear base de datos:

- `$ su postgres`
- `bash-4.2$ psql`
- `postgres=# CREATE DATABASE bdd_turismo WITH OWNER = postgres ENCODING='UTF8' TABLESPACE=pg_default CONNECTION LIMIT=-1;`

Tarea 2, 3, 4: Análisis de tablas, Creación de tablas, Datos de prueba. Se analiza y crea la tabla de la base de datos para almacenar la información sobre las visitas virtuales, con sus respectivos datos de prueba.

| sitios_vv | |
|--------------|------------------|
| id_sitio: | INTEGER [PK] |
| nombre: | text(2147483647) |
| descripcion: | text(2147483647) |
| upload_url: | text(2147483647) |
| publicado: | BOOLEAN |
| id_empresa: | INTEGER |
| lugar: | text(2147483647) |
| miniatura: | text(2147483647) |

Gráfica 28: Diagrama entidad - relación de la tabla del módulo VR.
Fuente: Propia.

Tarea 5: Análisis de requerimientos. En el análisis de requerimientos para el módulo se hace énfasis en la utilización del framework Symfony 1.4, php 5, html, css, xml, flash, jquery.

Se usan las librerías jquery.prettyPhoto.js, jquery.1.6.4.min.js, chapter_one.js, para dar el efecto de imagen flotante, el archivo AC_RunActiveContent.js para la ejecución de archivos flash, el archivo RyubinPanoPlayer5.js para visualizar las imágenes como panoramas en 360°, el archivo PanoramaCtrl07.swf para la barra de navegación de la

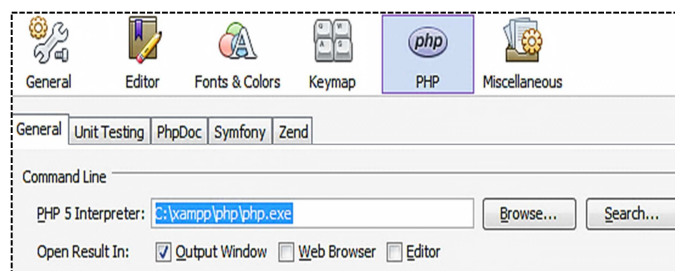
visita. Con respecto a las imágenes panorámicas, se aceptan en formatos: jpg, png, gif, con un tamaño máximo de 3Mb y de máximo 8000px de ancho por 1500px de alto.

Tarea 6: Estudio de herramientas.

- Framework Symfony
 - Estructura y funcionamiento.
 - Framework PHP con patrón de diseño MVC.
 - Soporte con IDE Netbeans.
 - Agrega una capa de abstracción de datos con el ORM Doctrine.
- IDE Netbeans
 - Funcionamiento
 - Permite crear aplicaciones web PHP.
 - Fácil integrar con Symfony.
- Base de datos PostgreSQL.
 - Fácil de administrar.
 - Soporte con framework Symfony.

Tarea 7: Crear proyecto web – aplicación frontend. Para la creación del proyecto web en Netbeans con el framework Symfony es necesario lo siguiente:

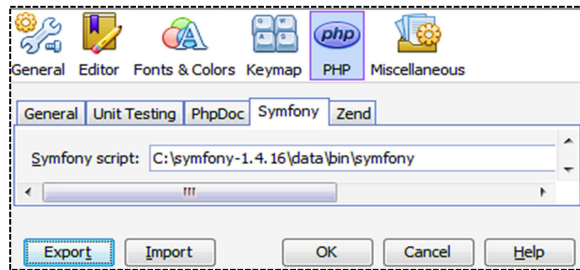
- Configuración de php en Netbeans
 - Ejecutar Netbeans
 - Seleccionar el intérprete php: *Tools* → *Options* → *PHP* → *General* → *PHP 5 Interpreter*.



Gráfica 29: Configurar PHP en Netbeans.

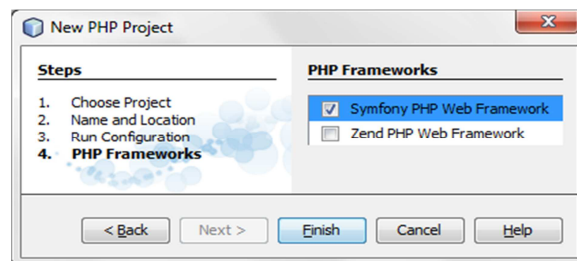
Fuente: Propia

- Configuración de Symfony en Netbeans.
 - Copiar el paquete de Symfony en algún directorio del servidor.
 - Configurar script de Symfony, el cual se encuentra dentro del directorio `symfony.1.4/data/bin/symfony`: *Tools* → *Options* → *PHP* → *Symfony* → *Symfony script*.

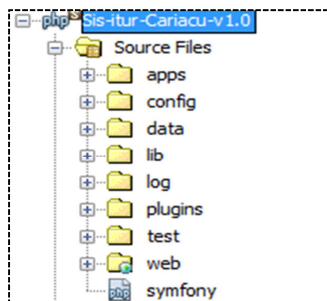


Gráfica 30: Configurar Symfony en Netbeans.
Fuente: Propia.

- Crear proyecto web
 - Crear web PHP: *File* → *New Project* → *PHP* → *PHP Application* → *Next* → *Sis-itur-Cariacu-v1.0* → *Next* → *Next* → *Symfony PHP Web Framework* → *Finish*.

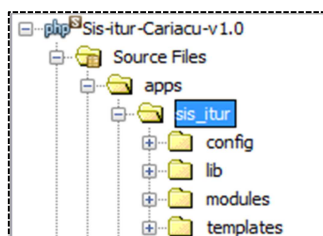


Gráfica 31: Crear nuevo proyecto web PHP con Symfony.
Fuente: Propia.



Gráfica 32: Estructura de directorio de nuevo proyecto.
Fuente: Propia.

- Crear aplicación frontend:
 - Clic derecho sobre nombre de *proyecto* → *Symfony* → *Run Command* → *symfony generate:app sis_itur*



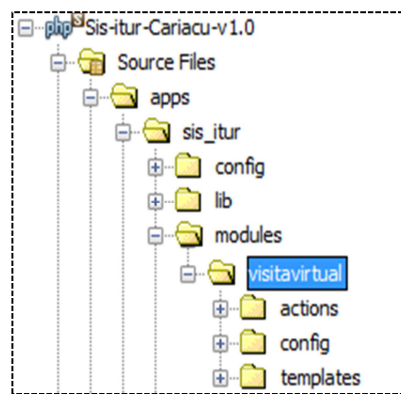
Gráfica 33: Creación de aplicación frontend - sis_itur.
Fuente: Propia.

Tarea 8: Configurar Symfony en aplicación:

- Configurar la conexión a base de datos en el archivo *Sis-itur-Cariacu-v1.0/config/database.yml*.
- Para que el proyecto pueda ser portable es necesario crear la carpeta vendor dentro del directorio *Sis-itur-Cariacu-v1.0/lib/* y copiar Symfony dentro de ella, *Sis-itur-Cariacu-v1.0/lib/vendor/symfony1.4*, y configurarlo en el archivo *Sis-itur-Cariacu-v1.0/config/ProjectConfiguration.class.php*.
- Finalmente generar el esquema de la base de datos, el modelo y los formularios dentro del proyecto: clic derecho sobre *proyecto* → *Symfony* → *Run Command* →

```
$symfony doctrine:build-schema  
$symfony doctrine:build-model  
$symfony doctrine:build-forms
```

Tarea 9: Creación de módulo VR. Para crear el módulo VR (visita virtual) dentro del aplicativo web se sigue los siguientes pasos: clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony generate:module sis_itur visitavirtual* → *run*.



Gráfica 34: Creación de módulo visitavirtual.
Fuente: Propia.

Tarea 10: Creación de Plantillas. La plantilla de inicio de las visitas virtuales:



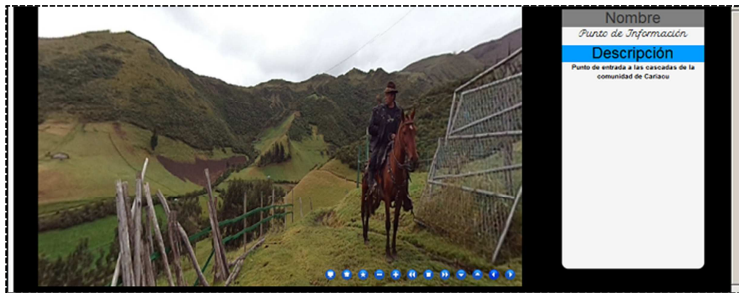
Gráfica 35: Interfaz de ingreso a módulo VR.
Fuente: Propia.

La pantalla donde se muestran las visitas disponibles es la siguiente:



Gráfica 36: Pantalla de lista de visitas.
Fuente: Propia

Al seleccionar una visita, esta se muestra en una ventana emergente:



Gráfica 37: Pantalla de visita virtual.
Fuente: Propia

Tarea 11: Creación de visitas virtuales. En la creación de las visitas virtuales se utilizaron varias clases .php que fueron generadas por Symfony, las cuales interactúan con las solicitudes de las plantillas:

| TARJETA CRC | | | |
|----------------------|---|---------------|--|
| Nombre Clase: | BaseSitiosVv | | |
| Descripción: | Es la clase de abstracción de datos de la tabla sitios_vv generada por Symfony. Acceso a base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Hace referencia a las claves foráneas | | |

Tabla 27: Tarjeta CRC de la clase BaseSitiosVv.
Fuente: Propia.

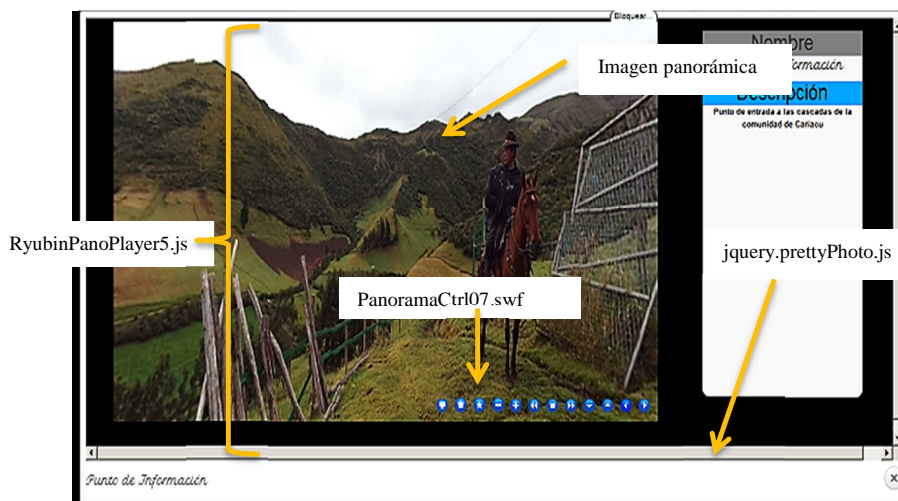
| TARJETA CRC | | | |
|-----------------------------|--|---------------|---|
| Nombre Clase: | SitiosVvTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a la tabla sitios_vv de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getFindBySitiosPublicados() | Retorna visitas virtuales disponibles. | | |
| getFindByIdSito(\$id) | Retorna visita virtual por id. | | |
| getFindAllVisitas() | Retorna todas la visitas virtuales | | |
| getBusqueda(\$frase) | Retorna visita virtual por etiqueta o nombre | | |

Tabla 28: Tarjeta CRC de la clase SitiosVvTable.
Fuente: Propia.

| TARJETA CRC | | | |
|--------------------|--|---------------|--|
| Nombre Clase: | VisitavirtualActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la visualización de las visitas virtuales. Capa Controlador. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Ejecuta la pantalla de inicio de las visitas virtuales. | Súper clase: | sfActions (Controlador genérico, clase de Symfony) |
| executeListPano() | Muestra la pantalla con la lista de visitas virtuales | | |
| executeView2Pano() | Muestra la visita virtual seleccionada. | | |

Tabla 29: Tarjeta CRC de las clase SitiosVv
Fuente: Propia.

Además se empleó código XML dentro de PHP para enlazar la imagen panorámica con los componentes flash y JavaScript que permiten visualizar las visitas virtuales.



Gráfica 38: Componentes interfaz de la visita virtual.
Fuente: Propia.

Tarea 12: Pruebas funcionalidad.

Existen dos formas de realizar las pruebas funcionales; la primera es manual y consiste en probar el módulo desde un navegador web y observar su funcionamiento, la segunda es la utilización de la herramienta de Symfony, la cual sirve para examinar las aplicaciones de inicio a fin, es decir, desde la solicitud realizada en el navegador hasta la respuesta que retorna del servidor. Comprueban las capas de enrutamiento, las acciones, el modelo, y las plantillas. Se ejecutan por medio de un navegador especial que proporciona la clase *sfBrowser*, y la clase *sfTestFunctional* que suministra métodos de pruebas.

Para hacer esta prueba seguir los siguientes pasos: clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony test:functional sis_itur visitavirtualActions* → *Run*

```

Output - Sis-itur-Cariacu-v1.0 (test:funcional sis_itur visitavirtualActions)
# get /visitavirtual/listPano
ok 1 - request parameter module is visitavirtual
ok 2 - request parameter action is listPano
ok 3 - status code is 200
ok 4 - response selector body does not match regex /This

```

Gráfica 39: Prueba funcional del módulo VR.

Fuente: Propia.

Finalización.

Tarea 13: Pruebas de aceptación. La siguiente es la prueba de aceptación de la creación de la base de datos del sistema.

| Prueba funcional | | |
|--|---|-----------------------|
| Número Prueba: 1 | Nro. Historia de Usuario: 1 | |
| Nombre de la Prueba: | Crear base de datos para el sistema. | |
| Descripción: | Se crea la base de datos bb_turismo en PostgreSQL | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| creación de la base de datos, desde la consola de administración de PostgreSQL | Base de datos creada correctamente. | Satisfactoria |

Tabla 30: Prueba de aceptación de creación de base de datos.

Fuente: Propia.

A continuación se muestra la prueba de aceptación de la creación de la tabla para el módulo VR (Visitas virtuales)

| Prueba funcional | | |
|----------------------------------|--|-----------------------|
| Número Prueba: 2 | Nro. Historia de Usuario: 1 | |
| Nombre de la Prueba: | Crear las tablas para el módulo VR(Visita virtual) | |
| Descripción: | Se analiza y diseña las tablas de la base de datos para las visitas virtuales. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Crear la tabla sitios_vv | Tabla normalizada, con integridad referencial. Contiene campos necesarios. | Satisfactoria |

Tabla 31: Prueba de aceptación de creación de tablas de módulo VR.

Fuente: Propia.

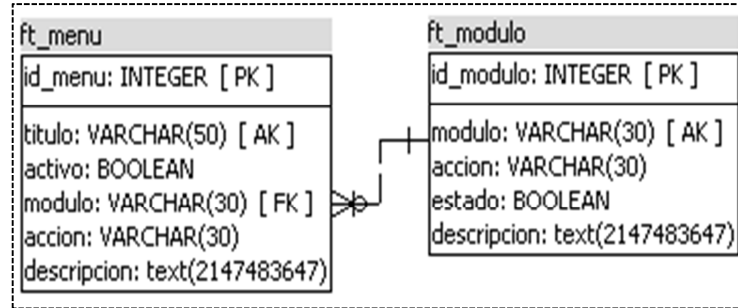
| Prueba funcional | | |
|---|---|-----------------------|
| Número Prueba: 3 | Nro. Historia de Usuario: 2 | |
| Nombre de la Prueba: | Crear el módulo VR(Visita virtual) | |
| Descripción: | Se analiza e implementa el módulo para la visualización de las visitas virtuales. Los usuarios al acceder al módulo VR pueden navegar por todas las visitas virtuales creadas, cada una de las visitas muestran una imagen panorámica con sus datos descriptivos. | |
| Condiciones de ejecución: | Base de datos y tabla creada. Herramientas configuradas. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| El usuario ingresa a módulo. | Visualizar pantalla de inicio | Satisfactoria. |
| Ingresar a las visitas virtuales disponibles al dar clic en botón "Haga clic para Entrar" | Visualizar visitas virtuales disponibles. | Satisfactoria. |
| Seleccionar visita virtual al dar clic en botón "Ver visita virtual" | Desplegar imagen panorámica en forma de visita virtual con datos informativos y una barra de navegación | Satisfactoria. |

Tabla 32: Prueba de aceptación de la implementación del módulo VR (Visita virtual).

Fuente: Propia.

Tarea 14: Integración.

El módulo es el primero que se implementa en el sistema por lo tanto no es necesario integrarlo con otros módulos, no obstante se diseñan dos tablas en la base de datos que proporcionan acceso y funcionalidad de todos los módulos dentro del sistema.



Gráfica 40: Diagrama entidad relación de tablas ft_modulo, ft_menu.
Fuente: Propia.

4.5. SPRINT 1: Desarrollo de módulo gestión de usuarios.

Finalizado el primer sprint se continúa con el desarrollo del módulo gestión de usuarios y seguridades, el cual tiene como finalidad proporcionar acceso privilegiado a ciertas funcionalidades del sistema. Además este interactuará con varios de los módulos que se van a implementar.

Planificación.

✓ Historias de usuario.

| HISTORIA DE USUARIO N° 3 | | | |
|--|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Diseño de tablas para el módulo de Gestión de Usuarios. | | |
| ID de historia: | 3 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 2 | Dependencia: | Ninguna. |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 22 – 07 – 2013 |
| Descripción | | | |
| Diseñar las tablas para el módulo de gestión de usuarios, tomando en cuenta todas las relaciones que posteriormente puede tener con otros módulos. | | | |
| Pruebas de Aceptación | | | |
| Las tablas deben de estar normalizadas en la forma más adecuada. Debe permitir ingresar datos de prueba con integridad referencial | | | |
| Observaciones | | | |
| | | | |

Tabla 33: Historia de usuario N° 3.
Fuente: Propia.

| HISTORIA DE USUARIO N° 4 | | | |
|---|---|---------------------|---|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Desarrollo del módulo de gestión de usuarios | | |
| ID de historia: | 4 | Usuario: | Sr. Germánico Guacán – Presidente grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 2 | Dependencia: | Ninguna. |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 22 – 07 – 2013 |
| Descripción | | | |
| Se debe desarrollar el módulo de gestión de usuarios. | | | |
| El módulo debe permitir que los visitantes puedan crearse una cuenta en el aplicativo, registrar la información solicitada previa validación. | | | |
| Además debe permitir que los usuarios puedan modificar su información. | | | |
| El módulo debe permitir la autenticación de los usuarios. | | | |
| Pruebas de Aceptación | | | |
| Este módulo debe permitir crear y modificar cuentas de usuarios del sistema. | | | |
| Debe permitir la autenticación correcta de los usuarios. | | | |
| Debe registrar datos validados. | | | |
| Observaciones | | | |
| Varios módulos dependerán de este. | | | |

Tabla 34: Historia de usuario N° 4.

Fuente: Propia.

✓ **Pila de producto.**

| PILA DE PRODUCTO | | |
|------------------|--|-----------|
| ID | Nombre Historia de Usuario | Prioridad |
| 3 | Diseño de tablas para módulo de Gestión de Usuarios. | ALTA |
| 4 | Desarrollo del módulo de gestión de usuarios. | ALTA |

Tabla 35: Pila de producto N° 2.

Fuente: Propia.

✓ **Pila de sprint.**

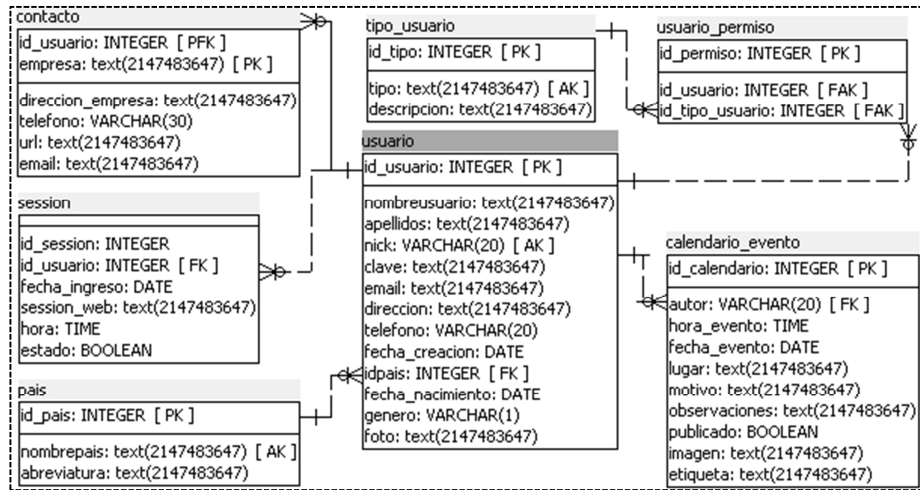
| PILA DE SPRINT | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|-----------------------------------|--|--------------|-----------|--|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| ID | Tareas | Tipo | Estado | Esfuerzo estimado (horas). | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Diseño de tablas para módulo de gestión de usuarios | | | | | | | | | | | | | | | | | | | |
| 1 | Análisis de tablas | Análisis | Terminada | 1 | | | | | | | | | | | | | | | | | | | |
| 2 | Creación de tablas. | Codificación | Terminada | 1 | | | | | | | | | | | | | | | | | | | |
| 3 | Datos de prueba. | Prueba | Terminada | | 1 | 1 | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Desarrollo del módulo de gestión de usuarios. | | | | | | | | | | | | | | | | | | | |
| 4 | Análisis de requerimientos | Análisis | Terminada | | 2 | 2 | | | | | | | | | | | | | | | | | |
| 5 | Estudio de herramientas | Análisis | Terminada | | 3 | 3 | 3 | | | | | | | | | | | | | | | | |
| 6 | Creación de módulo de gestión de usuarios. | Codificación | Terminada | | | 1 | | | | | | | | | | | | | | | | | |
| 7 | Creación de plantillas | Codificación | Terminada | | | 2 | 2 | 2 | 2 | | | | | | | | | | | | | | |
| 8 | Funcionalidad al módulo. | Codificación | Terminada | | | 1 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 1 |
| 9 | Pruebas funcionalidad | Prueba | Terminada | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| 10 | Pruebas de aceptación | Prueba | Terminada | | | | | | | | | | | | | | | | | | | | 3 |
| 11 | Integración | Codificación | Terminada | | | | | | | | | | | | | | | | | | | | 1 |

Tabla 36: Pila de sprint N° 2.

Fuente: Propia.

Desarrollo.

Tarea 1, 2, 3: Análisis de tablas. Creación de tablas. Datos de prueba. Para el módulo de gestión de usuarios se analizó y diseñó las correspondientes tablas, tomando en cuenta la dependencia de las mismas en los siguientes módulos a desarrollar.



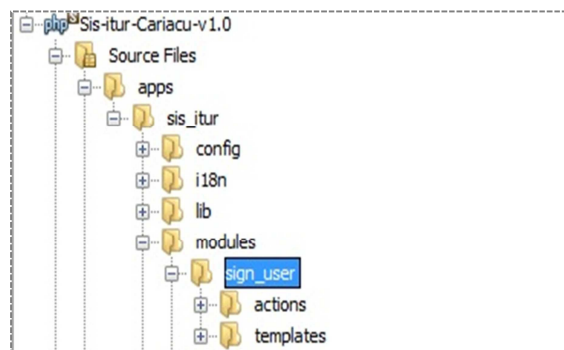
Gráfica 41: Diagrama entidad - relación de tablas del módulo de gestión de usuarios.

Fuente: Propia.

Tarea 4: Análisis de requerimientos. En esta parte se analiza los requerimientos necesarios para la gestión de las cuentas de los usuarios para el sistema. Se define que un usuario puede registrar su información y con ello obtendrá una cuenta con acceso privilegiado a funcionalidades del sistema.

Tarea 5: Estudio de herramientas. En esta tarea se continúa con el estudio del framework Symfony, así como también de lenguajes como Ajax, php, css.

Tarea 6: Creación de módulo de gestión de usuarios y seguridad. Para crear este módulo: clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony generate:module sis_itur sign_user* → *run*.



Gráfica 42: Creación de módulo sign_user.

Fuente: Propia.

Tarea 7: Creación de plantillas. Las plantillas creadas para este módulo son:

Para autenticación de usuarios se emplea la siguiente ventana:



Gráfica 43: Ventana usuario – para autenticarse.

Fuente: Propia.



Gráfica 44: Ventana de creación de cuentas de usuarios.

Fuente: Propia.



Gráfica 45: Ventana de perfil de usuario.

Fuente: Propia

Para actualizar la información del usuario este dispone de la siguiente ventana:

| Editar Perfil usuario ↔ luchin | |
|--|---|
| Nombre usuario | <input type="text" value="Luis Alberto"/> |
| Apellidos | <input type="text" value="Caluguillín Yascual"/> |
| Nick | <input type="text" value="luchin"/> |
| Email | <input type="text" value="atencionalcliente@ca"/> |
| Dirección | <input type="text" value="Cayambe"/> |
| Teléfono | <input type="text" value="0997695311"/> |
| Id país | <input type="text" value="Peru"/> |
| Fecha nacimiento | <input type="text" value="10"/> <input type="text" value="01"/> <input type="text" value="2000"/> |
| Genero [M = Masculino, F = Femenino] | <input type="text" value="M"/> |
| <input type="button" value="Regresar a Perfil Delete"/> <input type="button" value="Guardar"/> | |

Gráfica 46: Ventana editar perfil usuario.
Fuente: Propia.

Tarea 8: Funcionalidad al módulo. Se utilizaron varias clases .php generadas por Symfony, a continuación se mencionan las que agregan funcionalidad al módulo:

| TARJETA CRC | | | |
|------------------------|--|--------------------|--------------------------------------|
| Nombre Clase: | BaseUsuario | | |
| Descripción: | Es la clase de abstracción de datos de la tabla usuario. Acceso a Base de datos. | | |
| | Responsabilidades | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper Clase: | sfDoctrineRecord (Clase de Doctrine) |
| setUp() | Hace referencia a las claves foráneas | | |

Tabla 37: Tarjeta CRC de la clase BaseUsuario.
Fuente: Propia.

| TARJETA CRC | | | |
|-----------------------------------|---|--------------------|---|
| Nombre Clase: | UsuarioTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a la tabla usuario de la base de datos. Capa Modelo. | | |
| | Responsabilidades | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper Clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getUpdateFoto(\$id, \$foto) | Actualizar imagen de usuario | | |
| getChangePass(\$id, \$pass) | Cambiar contraseña | | |
| getValidarDatos(\$nick, \$pass) | Validar datos en login | | |
| getCheckUsuarioAndPass(\$id, \$p) | Valida contraseña para cambiar. | | |
| getFindByNick(\$nick) | Buscar por Nick. | | |
| getFindByIdUsuario(\$id) | Buscar por Id. | | |
| getFindAllUsuario() | Todos los usuarios | | |
| getFindByNombres(\$nom) | Buscar por nombre | | |
| getFindByGenero(\$genero) | Buscar por género. | | |
| getUsuarios() | Seleccionar Nick de usuarios. | | |
| getParametroUsuarios(\$par) | Usuarios por parámetro. | | |

Tabla 38: Tarjeta CRC de la clase UsuarioTable.
Fuente: Propia.

| TARJETA CRC | | | |
|------------------------|---|--------------------|--|
| Nombre Clase: | BaseContacto | | |
| Descripción: | Es la clase de abstracción de datos de la tabla contacto . Acceso a base de datos. | | |
| | Responsabilidades | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper Clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Hace referencia a las claves foráneas | | |

Tabla 39: Tarjeta CRC de la clase BaseContacto.
Fuente: Propia.

| TARJETA CRC | | | |
|-----------------------|--|---------------|---|
| Nombre Clase: | ContactoTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a la tabla contacto de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper Clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getContactoById(\$id) | Obtener contacto por id de usuario | | |

Tabla 40: Tarjeta CRC de la clase ContactoTable.

Fuente: Propia.

| TARJETA CRC | | | |
|---------------------------|---|---------------|---|
| Nombre Clase: | SessionTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a la tabla session de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper Clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| insertSession(\$id_u,\$s) | Insertar sesión de usuario | | |
| closeSession(\$id_u,\$f) | Actualizar sesión de usuario | | |

Tabla 41: Tarjeta CRC de la clase SessionTable.

Fuente: Propia.

| TARJETA CRC | | | |
|-----------------------|--|---------------|--|
| Nombre Clase: | sign_userActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar los procesos de autenticación, registro y actualización de los usuarios. Capa Controlador. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Ejecuta pantalla de inicio de módulo. | Súper Clase: | SfActions (Controlador genérico, clase de Symfony) |
| executeUpload() | Subir foto de usuario. | | |
| executeEdit() | Editar usuario | | |
| executeUpdate() | Actualizar usuario. | | |
| executeDelete() | Eliminar usuario. | | |
| processForm() | Procesar formulario de usuario. | | |
| executeSignIn() | Para crear usuario. | | |
| executeLogin() | Autenticación de usuario. | | |
| executeCerrarSesion() | Cerrar sesión de usuario. | | |
| executePerfil() | Ver pantalla de perfil de usuario. | | |
| executeAjaxSign() | Verificar nick de nuevo usuario. | | |
| executeSignUp() | Ver pantalla de registro de usuario. | | |
| executeChangePass() | Cambiar contraseña de usuario | | |

Tabla 42: Tarjeta CRC de la clase sign_userActions.

Fuente: Propia.

Tarea 9: Pruebas funcionalidad. Clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony test:functional sis_itur sign_userActions* → *Run*.

```

Output - Sis-itur-Cariacu-v1.0 (test:functional sis_itur sign_userActions)
# get /sign_user/signUp
ok 1 - request parameter module is sign_user
ok 2 - request parameter action is signUp
ok 3 - status code is 200
ok 4 - response selector body does not match regex /This

```

Gráfica 47: Prueba de funcionalidad de módulo de gestión de usuarios.

Fuente: Propia.

Crear un nuevo usuario. Ventana “Creación de Cuentas de Usuarios”. El formulario valida los datos ingresados:

| Datos Personales | | Empresa | |
|------------------|---|------------|---|
| Nombre Usuario: | <input type="text"/> * | Empresa | <input type="text"/> * |
| Apellidos: | <input type="text"/> * | Dirección | <input type="text"/> * |
| Nick: | <input type="text"/> x Nick No Disponible | Teléfono | <input type="text"/> * ej: 999.999.999 |
| Contraseña: | <input type="text"/> máximo 10 caracteres * | Página web | <input type="text"/> (opcional) ej: http://www.google.com |

Gráfica 48: Campos inválidos en ventana creación de cuentas de usuarios.

Fuente: Propia.

Creación correcta de usuarios: Una vez validados los campos en la ventana anterior se crea una nueva cuenta y re-direcciona a la ventana de perfil del usuario.

Gráfica 49: Creación correcta de nuevo usuario en ventana Perfil.

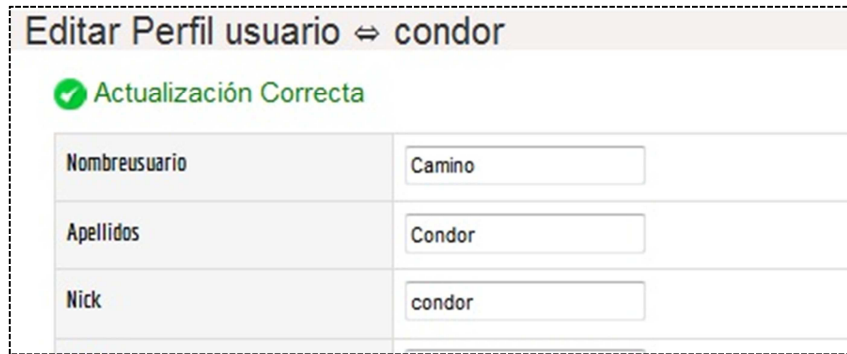
Fuente: Propia.

Actualización de datos: Para editar los datos del usuario seleccionar el link “**Editar**” de la ventana anterior. En la siguiente ventana se muestra los datos validados.

| Editar Perfil usuario ⇌ condor | |
|--------------------------------|---|
| ⊗ Datos Erróneos | |
| Nombre usuario | <input type="text" value="Camino"/> |
| Apellidos | <input type="text" value="Condor"/> |
| Nick | <ul style="list-style-type: none"> • Este valor es obligatorio. <input type="text"/> |

Gráfica 50: Actualización errónea en ventana editar perfil usuario.

Fuente: Propia.



Gráfica 51: Actualización correcta en ventana editar perfil usuario.

Fuente: Propia.

Autenticación e ingreso al sistema: Para autenticarse en el sistema se debe dirigir a la ventana "Usuario".



Gráfica 52: Autenticación errónea y correcta de usuarios.

Fuente: Propia.

Finalización.

Tarea 10: Pruebas de aceptación. A continuación las pruebas de aceptación:

| Prueba funcional | | |
|---|---|------------------------------------|
| Número Prueba: | 1 | Nro. Historia de Usuario: 3 |
| Nombre de la Prueba: | Crear tablas para módulo de gestión de usuarios. | |
| Descripción: | Análisis y diseño de las tablas adecuadas en la base de datos bb_turismo. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Creación de tablas para módulo. | Tablas normalizadas y con integridad referencial. | Satisfactoria |
| Ingresar, actualizar, selección y eliminar registros. | Operación satisfactoria. | Satisfactoria. |

Tabla 43: Prueba de aceptación de creación de tablas para módulo gestión usuarios.

Fuente: Propia.

La siguiente prueba de aceptación es la de creación de cuentas de usuarios.

| Prueba funcional | | |
|------------------------------|---|-----------------------------|
| Número Prueba: | 2 | Nro. Historia de Usuario: 4 |
| Nombre de la Prueba: | Crear y modificar cuentas de usuarios. | |
| Descripción: | El módulo debe permitir crear usuarios para el sistema y además debe permitir modificar los datos del mismo. Los datos son validados. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Crear cuentas de usuarios. | Crear usuarios para el sistema. | Satisfactoria |
| Modificar datos de usuarios. | Datos de usuarios modificados. | Satisfactoria. |

Tabla 44: Prueba de aceptación de creación y modificación de cuentas de usuarios.

Fuente: Propia.

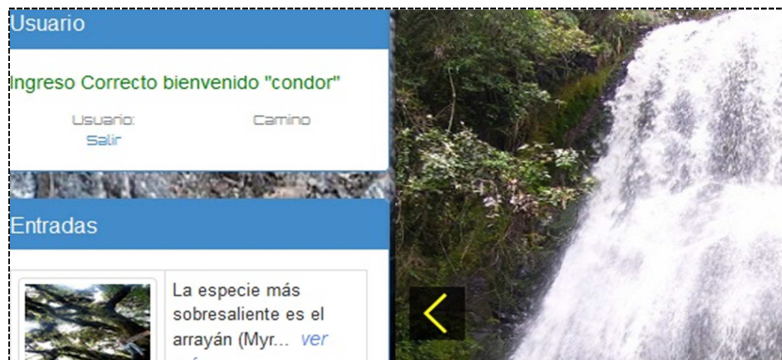
La siguiente prueba de aceptación es la de autenticación de usuarios.

| Prueba funcional | | |
|----------------------------|--|-----------------------------|
| Número Prueba: | 3 | Nro. Historia de Usuario: 4 |
| Nombre de la Prueba: | Autenticación correcta de usuarios. | |
| Descripción: | El módulo debe validar los datos del usuario para que pueda autenticarse y entrar con privilegios. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Autenticación de usuarios. | Validar datos de ingreso de usuario. | Satisfactoria |

Tabla 45: Prueba de aceptación de autenticación de usuarios.

Fuente: Propia.

Tarea 11: Integración. Se integra el módulo con el sistema al ubicar la pantalla de “Usuario” en la página de inicio del aplicativo.



Gráfica 53: Integración de módulo gestión de usuarios al sistema.

Fuente: Propia.

Se finaliza la iteración con la integración y con la reunión de fin de iteración.

4.6. SPRINT 2: Desarrollo de módulo foro

Continuando con la implementación del sistema turístico, se procede a analizar y planificar la funcionalidad del siguiente módulo de foros y comentarios, el cual tiene por finalidad la recolección de las opiniones de los visitantes. Este módulo depende del módulo anterior debido a que se necesitará una previa autenticación.

Planificación.

✓ Historias de usuario.

Fruto del acta de reunión de definición de requerimientos se analiza los requisitos planteados y se plasman en las siguientes historias de usuarios:

| HISTORIA DE USUARIO N° 5 | | | |
|--|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Diseño de tablas de la base de datos para el módulo foro. | | |
| ID de historia: | 5 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 3 | Dependencia: | 3 |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 11 – 08 – 2013 |
| Descripción | | | |
| Analizar y crear las tablas para la base de datos del módulo de foros. Normalizar las tablas. | | | |
| Pruebas de Aceptación | | | |
| Tablas normalizadas que permitan ingresar, modificar, eliminar y seleccionar datos con integridad referencial. | | | |
| Observaciones | | | |

Tabla 46: Historia de usuario N° 5.
Fuente: Propia.

| HISTORIA DE USUARIO N° 6 | | | |
|---|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Desarrollo del módulo de foros y comentarios. | | |
| ID de historia: | 6 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 3 | Dependencia: | 4 |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 11 – 08 – 2013 |
| Descripción | | | |
| Desarrollar el módulo de Foros y comentarios, donde se podrá crear temas de discusión y permitirá dejar comentarios en ellos. Para acceder al módulo se debe de autenticar ya que será solo para usuarios registrados. | | | |
| Pruebas de Aceptación | | | |
| El módulo debe permitir crear, editar, eliminar temas de discusión por los usuarios registrados y autenticados. El módulo debe permitir publicar comentarios en los temas de discusión. | | | |
| Observaciones | | | |
| Solo los usuarios registrados y autenticados deben tener acceso a este módulo, para un mejor control del contenido publicado en el módulo. | | | |

Tabla 47: Historia de usuario N° 6.
Fuente: Propia.

✓ Pila de producto.

| PILA DE PRODUCTO | | |
|------------------|---|-----------|
| ID | Nombre Historia de usuario | Prioridad |
| 5 | Diseño de tablas de la base de datos para el módulo foro. | ALTA |
| 6 | Desarrollo del módulo de foros y comentarios. | ALTA |

Tabla 48: Pila de producto del N° 3.
Fuente: Propia.

✓ Pila de Sprint.

A continuación la pila de sprint de la iteración:

| PILA DE SPRINT | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | |
|-----------------------------------|----------------------------------|--------------|-----------|--|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|---|
| ID | Tareas | Tipo | Estado | Esfuerzo estimado (horas). | | | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Diseño de tablas de la base de datos para el módulo foro. | | | | | | | | | | | | | | | | | | | | | |
| 1 | Análisis de tablas | Análisis | Terminada | 1 | | | | | | | | | | | | | | | | | | | | | |
| 2 | Creación de tablas. | Codificación | Terminada | 1 | | | | | | | | | | | | | | | | | | | | | |
| 3 | Datos de prueba. | Prueba | Terminada | | 1 | | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Desarrollo del módulo de foros y comentarios. | | | | | | | | | | | | | | | | | | | | | |
| 4 | Estudio de Herramientas | Análisis | Terminada | | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | |
| 5 | Creación de módulo de foro. | Codificación | Terminada | | | | 1 | | | | | | | | | | | | | | | | | | |
| 6 | Creación de plantillas | Codificación | Terminada | | | 3 | 2 | 1 | | | | | | | | | | | | | | | | | |
| 7 | Agregar funcionalidad al módulo. | Codificación | Terminada | | | | 4 | 4 | 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 4 | 4 | 4 | 4 | 2 | 1 | |
| 8 | Pruebas funcionalidad | Prueba | Terminada | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | Pruebas de aceptación | Prueba | Terminada | | | | | | | | | | | | | | | | | | | | | | 2 |
| 10 | Integración | Codificación | Terminada | | | | | | | | | | | | | | | | | | | | | | 1 |

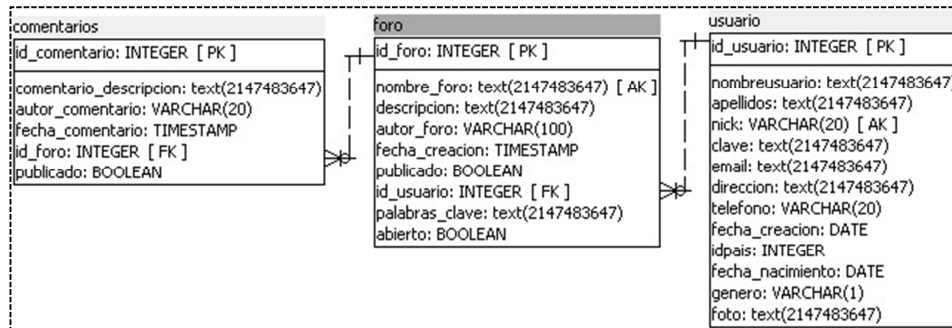
Tabla 49: Pila de Sprint N° 3.

Fuente: Propia.

Desarrollo

Una vez definidas las tareas se procede a implementarlas.

Tarea 1, 2, 3. Análisis de tablas. Creación de tablas. Datos de prueba. Se realiza el análisis y creación de las tablas necesarias para el módulo de foro.

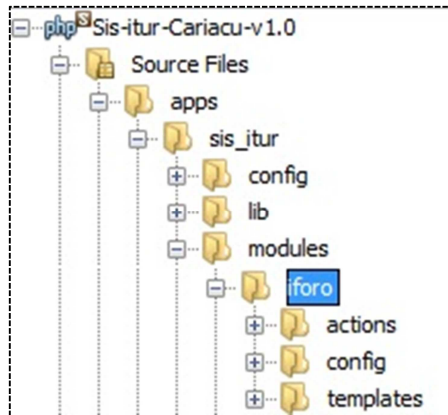


Gráfica 54: Diagrama entidad – relación de tablas para módulo foro.

Fuente: Propia.

Tarea 4: Estudio de herramientas. Se continúa con el estudio del framework y la herramienta de paginación, además se estudia el lenguaje de programación JavaScript, CSS para la maquetación y efectos.

Tarea 5: Creación de módulo de foro. Para la creación del módulo se sigue los siguientes comandos: Clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony generate:module sis_itur iforo* → *Run*.



Gráfica 55: Creación de módulo foro.
Fuente: Propia.

Tarea 6: Creación de plantillas. Para la interacción entre los usuarios del foro se crean varias plantillas.



Gráfica 56: Pantalla de bienvenida de foro
Fuente: Propia.

En la siguiente pantalla se muestran los temas publicados.



Gráfica 57: Pantalla de temas publicados.
Fuente: Propia.

Pantalla creación de nuevo temas de discusión y edición.

CREACIÓN DE NUEVO TEMA

Contenido moderado

| | |
|-------------|---|
| Título tema | <input type="text" value="Fiestas de Cariacu"/> |
| Descripción | <input type="text" value="Las fiestas anuales de la comunidad de Cariacu"/> |
| Etiquetas | <input type="text" value="fiesta,cariacu, 2014, San Pedro"/> |
| Publicado | <input checked="" type="checkbox"/> |
| Abierto | <input checked="" type="checkbox"/> |

[\(inicio\)](#) | [\(Mis temas\)](#) |

Gráfica 58: Pantalla de creación de nuevo tema.
Fuente: Propia.

EDICIÓN DE TEMA

✔ Ejecución Correcta

Contenido moderado

| | |
|-------------|---|
| Título tema | <input type="text" value="Fiestas de Cariacu 2014"/> |
| Descripción | <input type="text" value="Las fiestas anuales de la comunidad de Cariacu"/> |
| Etiquetas | <input type="text" value="fiesta,cariacu, 2014, San Pedro"/> |

Gráfica 59: Pantalla de edición de nuevo tema.
Fuente: Propia.

Pantalla de citar tema y mis temas.

Citar Tema

Fiestas de Cariacu 2014

Creado por: **condor** | Fecha: 2014-07-16 18:12:13.214854+00

" Las fiestas anuales de la comunidad de Cariacu "

» RE: condor | Fecha : 2014-07-16 18:15:08.19864+00

✓ como se llega

Gráfica 60: Pantalla de citar tema.
Fuente: Propia.

Mis Temas

| Acción | Tema | Contenido | Fecha |
|---------------------------------------|-------------------------|--|-------------------------------|
| <input type="button" value="editar"/> | Fiestas de Cariacu 2014 | Las fiestas anuales de la comunidad de Cariacu | 2014-07-16 18:12:13.214854+00 |

Gráfica 61: Pantalla mis temas.
Fuente: Propia.

Pantalla búsqueda de temas.

| BUSQUEDA DE TEMAS | | |
|---|--|---------------------------------|
| Busqueda: | | |
| Resultados de la busqueda 'Cariacu' = (2) encontrados | | |
| Tema | Descripción | Palabra Clave |
| Fiestas de Cariacu 2014 | Las fiestas anuales de la comunidad de Cariacu | fiesta.cariacu, 2014, San Pedro |

Gráfica 62: Pantalla de búsqueda de temas.
Fuente: Propia.

Tarea 7: Agregar funcionalidad al módulo. Para dar funcionalidad al módulo de foro se utiliza las clases .php generadas por el framework Symfony:

| TARJETA CRC | | | |
|----------------------|---|---------------|--|
| Nombre Clase: | BaseForo | | |
| Descripción: | Es la clase de abstracción de datos de la tabla foro . Acceso a Base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper Clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a las claves primarias y foráneas | | |

Tabla 50: Tarjeta CRC de la clase BaseForo.
Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|---------------|--|
| Nombre Clase: | BaseComentarios | | |
| Descripción: | Es la clase de abstracción de datos de la tabla comentarios . Acceso a base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper Clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a la clave foránea. | | |

Tabla 51: Tarjeta CRC de la clase BaseComentarios.
Fuente: Propia.

| TARJETA CRC | | | |
|-----------------------------|--|---------------|---|
| Nombre Clase: | ForoTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a la tabla foro de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getForoDisponible() | Retorna foros disponibles. | | |
| getFindByIdForo(\$id_foro) | Retorna foro por id. | | |
| getFindByForoByUsuario(\$i) | Retorna foro por id usuario | | |
| getBusqueda(\$frase) | Retorna foro por frase de búsqueda. | | |

Tabla 52: Tarjeta CRC de la clase ForoTable.
Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------------|---|---------------|---|
| Nombre Clase: | ComentariosTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a la tabla comentarios de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getFindComentarioByIdForo(\$i) | Retorna comentario por id de foro | | |
| getFindComentarioDisponible() | Retorna comentarios publicados | | |

Tabla 53: Tarjeta CRC de la clase ComentariosTable.
Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|---|---------------|--|
| Nombre Clase: | iforoActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar creación, actualización de temas de discusión y publicación de comentarios. Capa Controlador. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Procesa la pantalla de inicio del módulo. | Súper clase: | sfActions (Controlador genérico, clase de Symfony) |
| executeMytopics() | Procesa pantalla temas publicados. | | |
| executeTema() | Procesa pantalla citar tema. | | |
| executeBuscar() | Procesa pantalla búsqueda de temas. | | |
| executeNew() | Procesa pantalla creación nuevo tema. | | |
| executeCreate() | Procesa creación de tema. | | |
| executeEdit() | Procesa pantalla edición de tema. | | |
| executeUpdate() | Procesa actualización de datos. | | |
| executeDelete() | Procesa eliminación de temas. | | |
| processForm() | Procesa formulario de datos. | | |
| executeNewC() | Creación de comentario y enviar email. | | |

Tabla 54: Tarjeta CRC de la clase iforoActions.

Fuente: Propia.

Tarea 8: Pruebas funcionalidad.

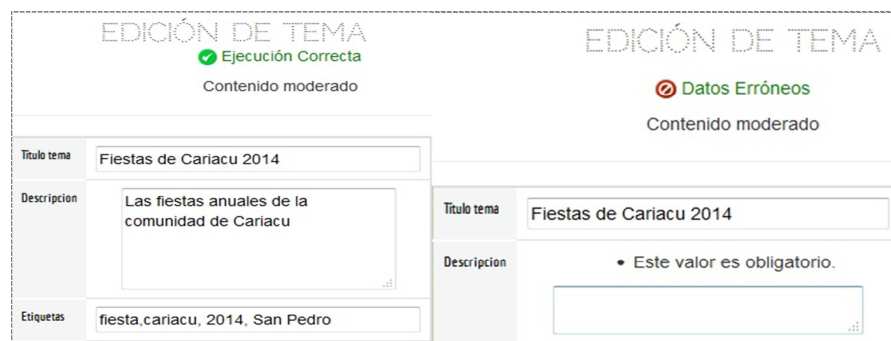
Ingreso al módulo. Pantalla de bienvenida, autenticarse o registrarse.



Gráfica 63: Prueba de ingreso al módulo foro.

Fuente: Propia.

Creación, edición de temas. Para crear un nuevo tema dar clic en botón “Nuevo Tema”, y para editar ingresar a pantalla “Mis Temas” y seleccionar “editar”.



Gráfica 64: Prueba de creación y actualización de tema de discusión.

Fuente: Propia.

Creación de Comentarios. Para dejar un comentario se debe seleccionar el tema dando clic en el botón “Citar” o en “Responder” según sea el caso.



Gráfica 65: Prueba de creación de comentario.
Fuente: Propia.

Finalización.

Tarea 9: Pruebas de aceptación. A continuación las pruebas de aceptación:

| Prueba funcional | | |
|--|---|-----------------------|
| Número Prueba: 1 | Nro. Historia de Usuario: 5 | |
| Nombre de la Prueba: | Crear tablas para módulo de foro. | |
| Descripción: | Análisis y diseño de las tablas adecuadas en la base de datos bb_turismo. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Creación de tablas para módulo. | Tablas normalizadas con integridad referencial. | Satisfactoria |
| Ingresar, actualizar, seleccionar, eliminar registros. | Operación satisfactoria. | Satisfactoria. |

Tabla 55: Prueba de aceptación de creación de tablas para módulo foro.
Fuente: Propia.

| Prueba funcional | | |
|----------------------------------|--|-----------------------|
| Número Prueba: 2 | Nro. Historia de Usuario: 6 | |
| Nombre de la Prueba: | Autenticación de usuarios. | |
| Descripción: | Para acceder al módulo es necesario que los usuarios se autentifiquen. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Autenticación de usuarios. | Datos validados. | Satisfactoria |
| Error en autenticación. | Mensaje de error. | Satisfactoria. |

Tabla 56: Prueba de aceptación de autenticación de usuarios.
Fuente: Propia.

| Prueba funcional | | |
|----------------------------------|---|-----------------------|
| Número Prueba: 3 | Nro. Historia de Usuario: 6 | |
| Nombre de la Prueba: | Crear, editar, eliminar, seleccionar temas de discusión. | |
| Descripción: | El módulo debe permitir realizar las 4 operaciones indicadas con éxito. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Crear tema. | Tema creado. | Satisfactoria. |
| Editar Tema. | Tema editado y actualizado. | Satisfactoria. |
| Eliminar Tema. | Tema eliminado. | Satisfactoria. |
| Seleccionar Temas | Mostrar listado de temas. | Satisfactoria. |

Tabla 57: Prueba de aceptación de los temas de discusión.
Fuente: Propia.

| Prueba funcional | | |
|----------------------------------|---|-----------------------|
| Número Prueba: 4 | Nro. Historia de Usuario: 6 | |
| Nombre de la Prueba: | Creación de comentarios. | |
| Descripción: | El módulo debe permitir crear comentarios en los temas. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Crear comentario. | Comentario creado. | Satisfactoria. |

Tabla 58: Prueba de aceptación de creación de comentarios.
Fuente: Propia.

Tarea 10: Integración. Integrar el módulo al sistema desde el menú principal.



Gráfica 66: Integración de módulo foro al sistema.

Fuente: Propia.

4.7. SPRINT 3: Desarrollo de módulo de contrataciones.

Siguiendo la planificación del proyecto el siguiente módulo a desarrollar es el módulo de contrataciones, el cual consta de un catálogo de productos y servicios que se ofertan en línea, a través de un carrito de compras, esto previa autenticación.

Planificación.

✓ Historias de usuario.

| HISTORIA DE USUARIO N° 7 | | | |
|--|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Diseño de tablas de la base de datos para el módulo de contrataciones y reservaciones. | | |
| ID de historia: | 7 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 4 | Dependencia: | 3 |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 02 – 09 – 2013 |
| Descripción | | | |
| Analizar y crear las tablas para la base de datos del módulo de contrataciones y reservaciones. | | | |
| Normalizar las tablas. | | | |
| Pruebas de Aceptación | | | |
| Las tablas deben de estar normalizadas, deben permitan ingresar, modificar, eliminar y seleccionar datos con integridad referencial. | | | |
| Observaciones | | | |
| Las tablas deben ser del tipo maestro – detalle para guardar la información de las transacciones. | | | |

Tabla 59: Historia de usuario N° 7

Fuente: Propia.

| HISTORIA DE USUARIO N° 8 | | | |
|--|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Análisis y desarrollo del módulo de contrataciones. | | |
| ID de historia: | 8 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 4 | Dependencia: | 4 |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 02 – 09 – 2013 |
| Descripción | | | |
| Analizar y desarrollar el módulo de contrataciones, el cual debe contar con un catálogo en línea de los productos y servicios disponibles en la localidad. La información de cada producto y servicio debe presentarse bien detallada. | | | |
| El módulo debe permitir mostrar todos los productos del catálogo, para que los usuarios autenticados puedan agregarlos a un carrito de compras. | | | |
| Además debe enviar correos electrónicos a los involucrados de las compras, para informar el detalle de la transacción. | | | |
| Se generarán documentos PDF con el detalle de la compra. El pago será de acuerdo a las políticas de la empresa. | | | |
| Pruebas de Aceptación | | | |
| El módulo debe mostrar el catalogo en línea, para que los usuarios puedan navegar a través de él, y si lo desean ellos puedan comprarlos, esto se realizará previa autenticación. | | | |
| El módulo debe enviar correos electrónicos al momento de realizar una transacción, para su posterior seguimiento, tanto al comprador como a la empresa, además debe generar un documento en formato PDF con el detalle de la compra. | | | |
| Observaciones | | | |
| Todo usuario puede acceder al módulo y navegar en su contenido, pero solo los usuarios registrados y autenticados pueden realizar las compras en línea. | | | |

Tabla 60: Historia de usuario N° 8.

Fuente: Propia.

✓ **Pila de producto.**

| PILA DE PRODUCTO | | |
|------------------|--|-----------|
| ID | Nombre Historia de usuario | Prioridad |
| 7 | Diseño de tablas de la base de datos para el módulo de contrataciones y reservaciones. | ALTA |
| 8 | Análisis y desarrollo del módulo de contrataciones. | ALTA |

Tabla 61: Pila de producto N° 4.
Fuente: Propia.

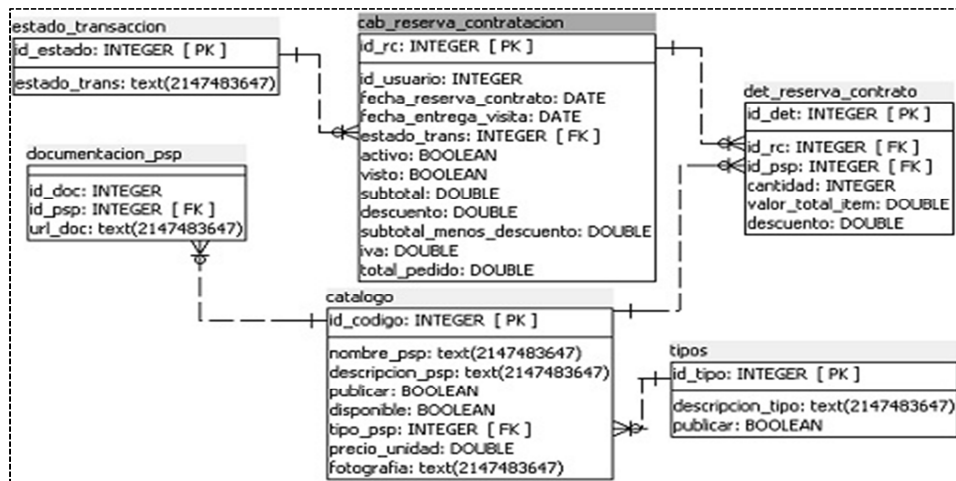
✓ **Pila de Sprint.**

| PILA DE SPRINT | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
|----------------------------|---------------------------------------|------------|-----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | Tareas | Tipo | Estado | Esfuerzo estimado (horas). | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Diseño de tablas de la base de datos para el módulo de contrataciones y reservaciones | | | | | | | | | | | | | | | | | | | |
| 1 | Análisis de tablas | Análisis | Terminada | 2 | | | | | | | | | | | | | | | | | | | |
| 2 | Creación de tablas. | Codificar | Terminada | 2 | | | | | | | | | | | | | | | | | | | |
| 3 | Datos de prueba. | Prueba | Terminada | | 2 | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Análisis y desarrollo del módulo de contrataciones. | | | | | | | | | | | | | | | | | | | |
| 4 | Estudio de herramientas | Análisis | Terminada | | 3 | 2 | 2 | 3 | | | | | | | | | | | | | | | |
| 5 | Creación de módulo de contrataciones. | Codificar. | Terminada | | | 1 | | | | | | | | | | | | | | | | | |
| 6 | Creación de plantillas | Codificar | Terminada | | | 4 | 2 | 2 | | | | | | | | | | | | | | | |
| 7 | Agregar funcionalidad al módulo. | Codificar | Terminada | | | | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 4 |
| 8 | Pruebas funcionalidad | Prueba | Terminada | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | Pruebas de aceptación | Prueba | Terminada | | | | | | | | | | | | | | | | | | | | 3 |
| 10 | Integración | Codificar | Terminada | | | | | | | | | | | | | | | | | | | | 2 |

Tabla 62: Pila de Sprint N° 4.
Fuente: Propia.

Desarrollo.

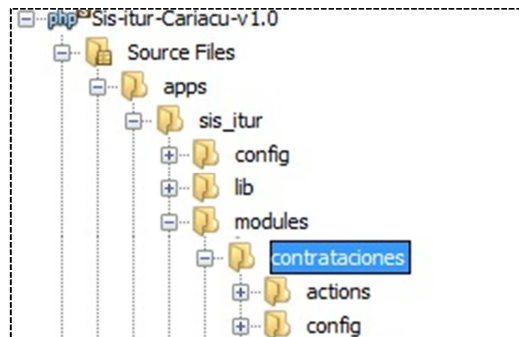
Tarea 1, 2, 3: Análisis de tablas. Creación de tablas. Datos de prueba. Se realiza el análisis y creación de las tablas necesarias para el módulo de contrataciones.



Gráfica 67: Diagrama entidad - relación de tablas para módulo contrataciones.
Fuente: Propia.

Tarea 4: Estudio de herramientas. En esta tarea se realiza el estudio de: comercio electrónico (PayPal), generación de reportes en documentos .pdf, y el uso del framework en generación de plantillas maestro-detalle.

Tarea 5: Creación de módulo de contrataciones. Para generar el módulo: clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony generate:module sis_itur contrataciones* → *Run*.



Gráfica 68: Creación de módulo de contrataciones.
Fuente: Propia.

Tarea 6: Creación de plantillas. Se crearon las plantillas necesarias para la implementación del módulo de contrataciones. A continuación la ventana de inicio.



Gráfica 69: Pantalla de inicio de módulo contrataciones
Fuente: Propia.

Pantalla de Catálogo: Muestra los productos y servicios disponibles.



Gráfica 70: Pantalla de catálogo de productos – servicios.
Fuente: Propia.

Pantalla de detalle de pedido: Muestra el detalle de una contratación.

| Datos de Contratación | | | | | | |
|-------------------------|----------------------|--------|-------------|--------------|-----------|--------|
| Usuario | Camino Condor | | Fecha | 2014/07/16 | | |
| Fecha Visita | ej.(2012/01/30) | | Transacción | Contrato | | |
| | | | Total | 28 | | |
| Productos Seleccionados | | | | | | |
| Codigo | Nombre | Precio | Cantidad | Valor Item | Descuento | Accion |
| 9 | llaveros decorativos | 5 | 5 | 25 | 0 | Quitar |
| | | | | Sub total: | 25 | |
| | | | | Desc: | 0 | |
| | | | | Total: | 25 | |
| | | | | Iva 12% : | 3 | |
| | | | | Total Pagar: | 28 | |

Gráfica 71: Pantalla de detalle de pedido.
Fuente: Propia.

Pantalla de forma de pago: Forma de pago de la contratación.

Gráfica 72: Pantalla de forma de pago.
Fuente: Propia.

Pantalla de pedidos de usuario: Muestra pedidos del usuario.

| Mis Pedidos - Camino Condor | | | | | | | | | | | |
|---------------------------------------|----------------|---------------|--------|-------|-------|---------------------------------------|----------------|---------------|--------|-------|------|
| Contratado - Reservado | | | | | | Cancelado - Finalizado | | | | | |
| Estado: 1 = Contratado, 2 = Reservado | | | | | | Estado: 3 = Cancelado, 4 = Finalizado | | | | | |
| Id Pedido | Fecha Contrato | Fecha Entrega | Estado | Valor | Ver. | Id Pedido | Fecha Contrato | Fecha Entrega | Estado | Valor | Ver. |
| 20 | 2014-07-16 | 2014-07-31 | 1 | 28 | más.. | | | | | | |

Gráfica 73: Pantalla mis pedidos.
Fuente: Propia.

Tarea 7: Agregar funcionalidad al módulo. Se emplean las clases .php generadas por Symfony.

| TARJETA CRC | | | |
|----------------------|---|--------------|--|
| Nombre Clase: | BaseCatalogo | | |
| Descripción: | Es la clase de abstracción de datos de la tabla catalogo . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas. | | |

Tabla 63: Tarjeta CRC de la clase BaseCatalogo.
Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|--|
| Nombre Clase: | BaseCabReservaContratacion | | |
| Descripción: | Clase de abstracción de la tabla cab_reserva_contratacion . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas. | | |

Tabla 64: Tarjeta CRC de la clase BaseCabReservaContratacion.
Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|--|
| Nombre Clase: | BaseDetReservaContrato | | |
| Descripción: | Clase de abstracción de la tabla det_reserva_contrato . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla. | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas. | | |

Tabla 65: Tarjeta CRC de la clase BaseDetReservaContrato.
Fuente: Propia.

Las siguientes tablas muestran las clases modelos del módulo.

| TARJETA CRC | | | |
|------------------------------|---|--------------|---|
| Nombre Clase: | CatalogoTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla catalogo de la base de datos. Capa Modelo. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene instancia de la tabla. | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getFindByCodigoCatalogo(\$i) | Buscar producto por id | | |
| getBusqueda(\$frase) | Busca producto por frase. | | |
| getAllPublicado() | Productos publicados. | | |
| findAllCatalogos() | Todos los productos. | | |

Tabla 66: Tarjeta CRC de la clase CatalogoTable.
Fuente: Propia.

| TARJETA CRC | | | |
|------------------------|---|--------------|---|
| Nombre Clase: | DetReservaContratoTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla det_reserva_contrato de la base de datos. Capa Modelo. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getDetallePedido(\$id) | Obtiene detalle contrato por id | | |

Tabla 67: Tarjeta CRC de la clase DetReservaContratoTable.
Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------------|---|---------------|---|
| Nombre Clase: | CabReservaContratacionTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla cab_reserva_contratacion de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getFindByCodigoPedido(\$id) | Buscar contrato por id | | |
| getContratoCRByUsuario(\$id) | Buscar contrato por usuario | | |
| getFindAllContratos() | Obtener todos los contratos. | | |
| getFindByFecha(\$fecha) | Buscar por fecha. | | |
| getFindByMesAnio(\$año, \$mes) | Buscar por mes y año. | | |
| getConfirm(\$id_rc) | Confirmar contrato por email. | | |
| setEstadoTrans(\$id_rc) | Actualizar estado de contrato. | | |
| getFindNoVistos() | Contratos no verificados. | | |
| getUpdateVisto(\$id_rc) | Actualizar contrato. | | |

Tabla 68: Tarjeta CRC de la clase CabReservaContratacionTable.

Fuente: Propia.

La siguiente tabla muestra la clase controladora del módulo.

| TARJETA CRC | | | |
|-------------------------------|--|---------------|--|
| Nombre Clase: | contratacionesActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar creación, visualización, actualización de productos, servicios del catálogo, además controla creación, actualización, visualización de contratos y reservas de elementos del catálogo. Capa Controlador. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Procesa la pantalla de inicio del módulo. | Súper clase: | sfActions (controlador genérico, clase de Symfony) |
| executeCatalogo() | Procesa pantalla de catálogo. | | |
| executeShowElement() | Visualiza elemento de catálogo. | | |
| executeAddCar() | Agrega elemento a carrito de compras. | | |
| executeDeleteCatalogo() | Eliminar elemento de carrito. | | |
| executeSaveContrato() | Guardar datos de pedido. | | |
| executeContrato() | Procesa pantalla de Detalle de pedido. | | |
| executeCancelarpedido() | Cancelar contratación. | | |
| executeCancelarPedidoPaypal() | Cancelar contratación desde paypal. | | |
| executePayment() | Procesa pantalla de forma de pago. | | |
| executeEndPayment() | Finaliza proceso de contratación. | | |
| executePayEfecto() | Procesa pago en efectivo. | | |
| executeReserva() | Procesa reservación de elementos. | | |
| executeConfirm() | Procesa confirmación de contratación. | | |
| executeMyOrders() | Procesa pantalla mis pedidos. | | |
| executeViewPedido() | Visualiza pedido. | | |
| executeGeneratePdf() | Genera PDF de contratación. | | |

Tabla 69: Tarjeta CRC de la clase contratacionesActions.

Fuente: Propia.

Tarea 8: Pruebas de funcionalidad.

Ingreso al módulo: Para ingresar al módulo no es necesario estar autenticado.

Visualización de productos: Se visualiza la información de los elementos del catálogo al hacer clic en la opción “ver más.”



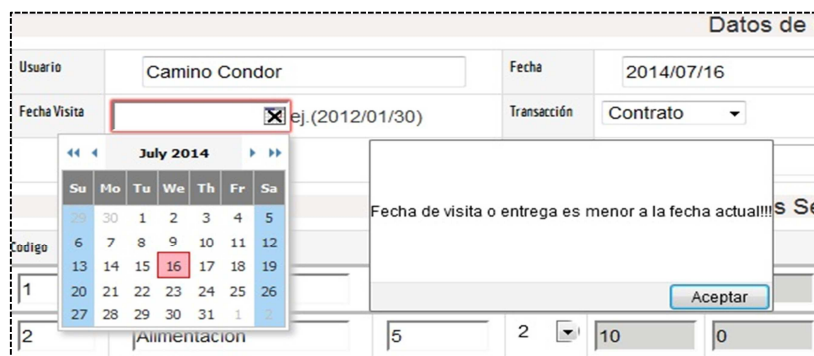
Gráfica 74: Ver información de elemento de catálogo.
Fuente: Propia.

Agregar elemento al carrito de compras: Para agregar elementos al carrito de compras es necesario autenticarse. Añadir elementos, clic en la imagen del carrito.



Gráfica 75: Prueba agregar elemento a carrito compras.
Fuente: Propia.

Datos de contratación: Para visualizar los datos de la contratación dar clic en “ver pedido”, y seguidamente modificar los datos necesarios.



Gráfica 76: Prueba de validación de datos en pedido.
Fuente: Propia.

| Productos Seleccionados | | | | | | |
|-------------------------|------------------|--------|----------|------------|-----------|--------|
| Codigo | Nombre | Precio | Cantidad | Valor Item | Descuento | Accion |
| 1 | colchas tejidas. | 10 | 1 | 10 | 0 | Quitar |
| 2 | Alimentacion | 5 | 2 | 10 | 0 | Quitar |

| Productos Seleccionados | | | | | | |
|-------------------------|------------------|--------|----------|------------|-----------|--------|
| Codigo | Nombre | Precio | Cantidad | Valor Item | Descuento | Accion |
| 1 | colchas tejidas. | 10 | | 0.0 | 0.0 | Quitar |

Gráfica 77: Prueba "quitar" elemento de pedido.
Fuente: Propia.

| Codigo | Nombre | Precio | Cantidad | Valor Item | Descuento | Accion |
|--------|------------------|--------|----------|------------|-----------|--------|
| 1 | colchas tejidas. | 10 | 6 | 60 | 0 | Quitar |



| | |
|--------------|-------------|
| Sub total: | 60 |
| Desc: | 0 |
| Total: | 60 |
| Iva 12% : | 7.2 |
| Total | 67.2 |

Gráfica 78: Prueba de cálculo de costos.
Fuente: Propia.

Catalogo de Productos / Servicios / Paquetes

✓ Su pedido fue Cancelado..

← 1 2 3 →

| | |
|--|--|
|  <p>Alimentacion \$ 5</p> |  <p>colchas tejidas. \$ 10</p> |
|--|--|

Gráfica 79: Prueba "cancelar pedido".
Fuente: Propia.

FORMA DE PAGO - GRACIAS POR SU COMPRA

Seleccione la **forma de pago**, para finalizar la transacción

| | |
|---|---|
| <p>Pago Normal - Efectivo</p> <p>Pagar en efectivo, esto se realizará en nuestras oficinas, contáctese con el representante del grupo de Ecoturismo</p> <p></p> | <p>Pago por PayPal</p> <p>Inactivo</p> |
|---|---|

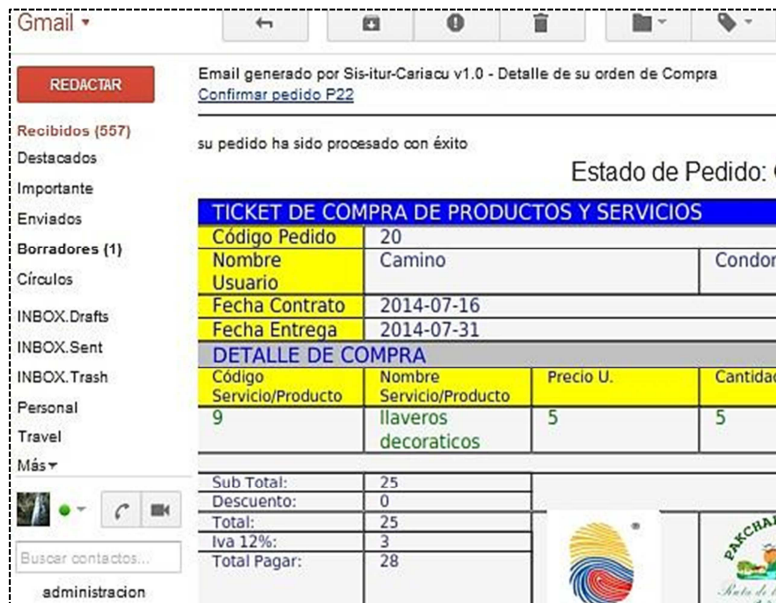
| TICKET DE COMPRA DE PRODUCTOS Y SERVICIOS | | | | |
|--|--------------------------|-----------|----------|-------------|
| Código Pedido | 20 | | Condor | |
| Nombre | Camino | | | |
| Usuario | | | | |
| Fecha Contrato | 2014-07-16 | | | |
| Fecha Entrega | 2014-07-31 | | | |
| DETALLE DE COMPRA | | | | |
| Código Servicio/Producto | Nombre Servicio/Producto | Precio U. | Cantidad | Valor total |
| 9 | llaveros decorativos | 5 | 5 | 25 |
| Sub Total: | | 25 | | |
| Descuento: | | 0 | | |
| Total: | | 25 | | |
| Iva 12%: | | 3 | | |
| Total Pagar: | | 28 | | |



Gráfica 80: Prueba forma de pago, "elegir y finalizar".
Fuente: Propia.



Gráfica 81: Prueba compra correcta.
Fuente: Propia.



Gráfica 82: Prueba envío de informe por email.
Fuente: Propia.

Finalización.

Tarea 9: Pruebas de aceptación. En las siguientes tablas se muestran las pruebas de aceptación del módulo.

| Prueba funcional | | |
|--|---|-----------------------------|
| Número Prueba: | 1 | Nro. Historia de Usuario: 7 |
| Nombre de la Prueba: | Crear tablas para módulo de contrataciones. | |
| Descripción: | Análisis y diseño de las tablas adecuadas en la base de datos bb_turismo. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Creación de tablas para módulo. | Tablas normalizadas y con integridad referencial. | Satisfactoria |
| Ingresar, actualizar, selección, eliminar registros. | Operación satisfactoria. | Satisfactoria. |

Tabla 70: Prueba de aceptación de creación de tablas para módulo contrataciones.
Fuente: Propia.

| Prueba funcional | | |
|--|---|------------------------------------|
| Número Prueba: | 2 | Nro. Historia de Usuario: 8 |
| Nombre de la Prueba: | Navegación por catálogo en línea y opción de compra. | |
| Descripción: | Para acceder al módulo no es necesario autenticarse pero para contratar sí. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Disponer de catálogo en línea. | Catálogo con productos y servicios. | Satisfactoria |
| Visualización de información de elementos de catálogo. | Información detallada de cada elemento de catálogo. | Satisfactoria. |
| Autenticación para comprar. | Usuario registrado y autenticado | Satisfactoria. |
| Comprar elementos del catálogo. | Productos y servicios contratados | Satisfactoria. |

Tabla 71: Prueba de aceptación de disponibilidad de catálogo en línea y compra.

Fuente: Propia.

| Prueba funcional | | |
|-------------------------------------|---|------------------------------------|
| Número Prueba: | 3 | Nro. Historia de Usuario: 8 |
| Nombre de la Prueba: | Generación de informes .PDF y envío de email | |
| Descripción: | En caso de comprar generar informe y enviar por email los datos de la compra. | |
| Condiciones de ejecución: | Ninguna. | |
| Entradas | | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Generar informe PDF de compra. | Documento .pdf de compra. | Satisfactoria |
| Envío de datos de compra por email. | Envío de email a usuario y administrador | Satisfactoria. |

Tabla 72: Prueba de generación de informe PDF y envío por email.

Fuente: Propia.

Tarea 10: Integración. El paso final es integrar el módulo con el sistema, para ello se incorpora un vínculo en el menú principal, para desde ahí acceder al módulo.



Gráfica 83: Integración del módulo de contrataciones.

Fuente: Propia.

4.8. SPRINT 4: Desarrollo de módulo de gestión de la información.

Continuando con el proyecto se desarrolla el módulo de gestión de la información, el cual tiene el funcionamiento de mostrar el contenido web en la parte frontal y la administración y gestión de los mismos en la parte backend.

Planificación.

✓ Historias de usuario.

| HISTORIA DE USUARIO N° 9 | | | |
|--|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Diseño de tablas de la base de datos para el módulo de gestión de la información. | | |
| ID de historia: | 9 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 5 | Dependencia: | 3, 5, 7 |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 22 – 10 – 2013 |
| Descripción | | | |
| Analizar diseñar y crear las tablas en la base de datos para el módulo de gestión de la información. | | | |
| Normalizar las tablas, integridad referencial. | | | |
| Pruebas de Aceptación | | | |
| Las tablas deben de estar normalizadas, permitan ingresar, modificar, eliminar y seleccionar datos con integridad referencial. | | | |
| Observaciones | | | |
| Las tablas se crearán de acuerdo a las necesidades del sitio y su tipo de contenido. | | | |

Tabla 73: Historia de usuario N° 9.

Fuente: Propia.

| HISTORIA DE USUARIO N° 10 | | | |
|---|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Análisis y desarrollo del módulo de gestión de la información. | | |
| ID de historia: | 10 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 5 | Dependencia: | 2, 4, 6, 8 |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 22 – 10 – 2013 |
| Descripción | | | |
| Analizar, diseñar y desarrollar el módulo de gestión de la información, el cual cuenta de 2 partes; la primera es la parte frontal que mostrará el contenido web del sitio y la segunda es la parte administrativa que servirá para gestionar el contenido. | | | |
| La parte frontal del módulo debe mostrar toda la información que disponen todos los módulos del aplicativo. | | | |
| La parte administrativa permitirá crear, actualizar, eliminar, seleccionar los elementos de todos los módulos. | | | |
| Para acceder a la parte administrativa debe hacerse previa autenticación. | | | |
| La gestión de la información debe ser dinámica. | | | |
| Pruebas de Aceptación | | | |
| El módulo debe integrar todos los módulos del sistema. | | | |
| Se debe autenticar como administrador para acceder a la parte administrativa. | | | |
| La parte administrativa debe permitir gestionar toda la información de los módulos existentes. | | | |

Tabla 74: Historia de usuario N° 10.

Fuente: Propia.

✓ Pila de producto.

| PILA DE PRODUCTO | | |
|------------------|---|-----------|
| ID | Nombre Historia de Usuario | Prioridad |
| 9 | Diseño de tablas de la base de datos para el módulo de gestión de la información. | ALTA |
| 10 | Análisis y desarrollo del módulo de gestión de la información. | ALTA |

Tabla 75: Pila de producto del módulo de gestión de la información.

Fuente: Propia.

✓ **Pila de Sprint**

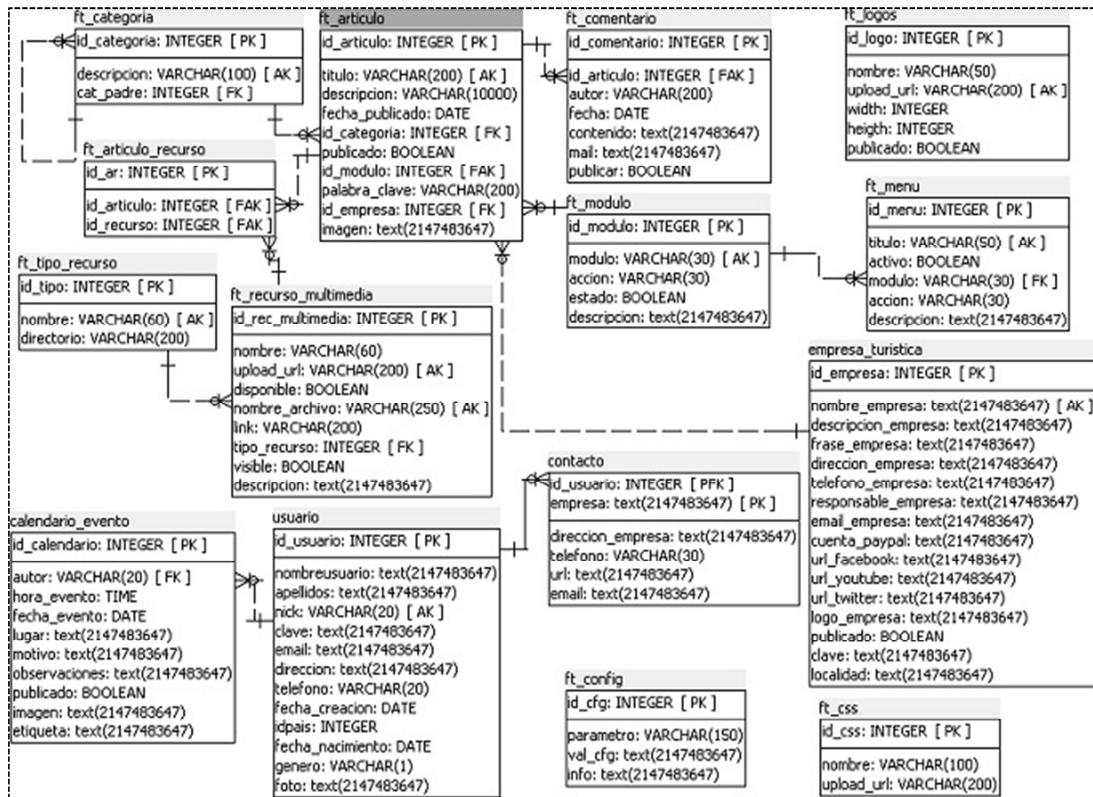
| PILA DE SPRINT | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | | | | |
|-----------------------------------|---------------------------------------|--------------|-----------|--|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|--|
| ID | Tareas | Tipo | Estado | Esfuerzo estimado (horas). | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Diseño de tablas de la base de datos para el módulo de gestión de la información. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | Análisis de tablas. | Análisis | Terminada | 3 | 3 | 3 | 3 | 3 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | Creación de tablas. | Codificación | Terminada | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 | Datos de prueba. | Prueba | Terminada | 1 | 1 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nombre Historia de Usuario | | | | Análisis y desarrollo del módulo de gestión de la información. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | Estudio de herramientas | Análisis | Terminada | 2 | 2 | 3 | 3 | 3 | 3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5 | Creación de aplicación administrativa | Codificación | Terminada | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6 | Creación de módulos administrativos | Codificación | Terminada | | | | | 1 | 2 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7 | Creación de módulos frontales. | Codificación | Terminada | | | | | | | 2 | 2 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8 | Desarrollo de parte frontal. | Codificación | Terminada | | | | | | | | | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | |
| 9 | Desarrollo de parte administrativa | Codificación | Terminada | | | | | | | | | | 1 | 1 | 1 | 2 | 2 | 3 | 2 | 2 | 2 | 2 | 5 | 5 | 5 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | |
| 10 | Pruebas de funcionalidad | Prueba | Terminada | | | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| 11 | Pruebas de aceptación | Prueba | Terminada | | | | | | | | | | | | | | | | | | | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | |
| 12 | Integración | Codificación | Terminada | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | 2 | 4 | | |

Tabla 76: Pila de Sprint del módulo de gestión de la información.
Fuente: Propia.

Desarrollo.

Una vez definidas las actividades se procede a ejecutarlas cada una de ellas.

Tarea 1, 2, 3: Análisis de tablas. Creación de tablas. Datos de prueba. Se realiza el análisis y creación de las tablas necesarias para el módulo.

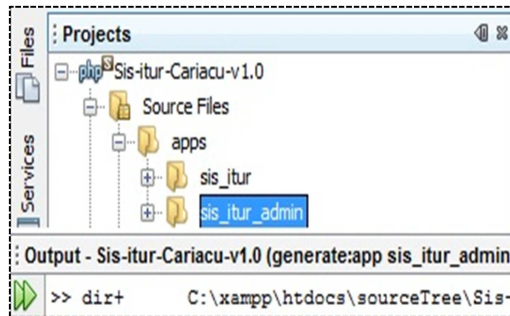


Gráfica 84: Diagrama entidad - relación de tablas para módulo de gestión de la información.

Fuente: Propia.

Tarea 4: Estudio de herramientas. En esta tarea se continua con el estudio de las herramientas de programación, tales como, el framework Symfony que es utilizado para generar la parte frontal como también en la generación de la parte backend o administrativa de la aplicación, además se estudia la utilización de plugin para la autenticación, la vista y reportes. También se analiza la forma de gestionar toda la información que se muestra al turista.

Tarea 5: Creación de aplicación administrativa. Para la creación de la aplicación de administración de proyecto se procede a ejecutar los siguientes comandos del framework: clic derecho sobre *proyecto* → *Symfony* → *Run Command* → *symfony generate:app sis_itur_admin* → *Run*.



Gráfica 85: Creación de aplicación administrativa.
Fuente: Propia.

Tarea 6: Creación de módulos administrativos. En esta tarea se generan todos los módulos creados en las anteriores iteraciones, con la diferencia que son utilizados para gestionar toda la información que en cada uno de ellos se dispone, es decir, crear, actualizar, seleccionar y eliminar todo lo que se muestra al turista. Para ello se ejecuta el siguiente comando: clic derecho sobre *proyecto* → *Symfony* → *Run command* → *symfony doctrine:generate-admin aplicación_backend Clase --module=Módulo*.

| Módulo Frontal | Comando/Módulo administrativo |
|------------------------|---|
| Visita virtual | <code>symfony doctrine:generate-admin sis_itur_admin SitiosVv --module=visitavirtual</code> |
| Gestión usuarios | <code>symfony doctrine:generate-admin sis_itur_admin Usuario --module=usuario</code> |
| Foro | <code>symfony doctrine:generate-admin sis_itur_admin Foro --module=foro</code> <code>symfony doctrine:generate-admin sis_itur_admin Comentarios --module=comentarios</code> |
| Contrataciones | <code>symfony doctrine:generate-admin sis_itur_admin Catalogo --module=catalogo</code> <code>symfony doctrine:generate-admin sis_itur_admin CabReservaContratacion --module=contratocab</code> <code>symfony doctrine:generate-admin sis_itur_admin DocumentacionPsp --module=docatalog</code> |
| Gestión de Información | <code>symfony doctrine:generate-admin sis_itur_admin FtArticuloRecurso --module=addressresourcearticle</code> <code>symfony doctrine:generate-admin sis_itur_admin FtCss --module=admcss</code> <code>symfony doctrine:generate-admin sis_itur_admin CalendarioEvento --module=adminevento</code> <code>symfony doctrine:generate-admin sis_itur_admin FtArticulo --module=articulo</code> <code>symfony doctrine:generate-admin sis_itur_admin FtComentario --module=comentarioarticulo</code> <code>symfony doctrine:generate-admin sis_itur_admin EmpresaTuristica --module=empresas</code> <code>symfony doctrine:generate-admin sis_itur_admin EnlacesExternos --module=enlaces</code> <code>symfony doctrine:generate-admin sis_itur_admin FtRecursoMultimedia --module=recmultimedia</code> |

Tabla 77: Creación de módulos de administración.
Fuente: Propia.

Tarea 7: Creación de módulos frontales. Se crean varios módulos en la parte frontal del aplicativo, con el objetivo de mostrar de forma dinámica toda la información sobre los sitios turísticos de interés para los turistas. Los módulos creados agregan funcionalidades extras al sistema ya que ellos colaboran en la gestión de los recursos.

| Aplicación / Módulo | Comando / Módulo |
|---|--|
| sis_itur (módulo gestión de la información) | <code>symfony generate:module sis_itur inicio</code> |
| | <code>symfony generate:module sis_itur turismoAventura</code> |
| | <code>symfony generate:module sis_itur turismoCultural</code> |
| | <code>symfony generate:module sis_itur turismoComunitario</code> |
| | <code>symfony generate:module sis_itur gastronomia</code> |
| | <code>symfony generate:module sis_itur evento</code> |
| | <code>symfony generate:module sis_itur galeria</code> |
| | <code>symfony generate:module sis_itur busqueda</code> |
| | <code>symfony generate:module sis_itur contactenos</code> |
| | <code>symfony generate:module sis_itur mapaSitio</code> |
| | <code>symfony generate:module sis_itur noticias</code> |
| | <code>symfony generate:module sis_itur upload</code> |
| | <code>symfony generate:module sis_itur error</code> |

Tabla 78: Creación de módulos frontales.
Fuente: Propia.

Tarea 8: Desarrollo de parte frontal. Una vez creados los módulos frontales se procede a desarrollar las funcionalidades en cada uno de ellos.

- **Creación de vistas.** A continuación se muestran las vistas creadas:



Gráfica 86: Pantalla frontal de inicio.
Fuente: Propia.



Gráfica 87: Pantalla turismo de aventura.
Fuente: Propia.



Gráfica 88: Pantalla de galería de imágenes.
Fuente: Propia.

Mediano «2014-03-17» **Gastronomía** [Regresar](#)

etiqueta: mediano



Mote cocinado, huevos cocinados, queso y papas cocinadas con salsa de pepa de zampu zapallo, y sobre esto van doce gallinas asadas, cuyes asados y carne de borrego cocinada, hojas de lechuga.

Ver Recursos

Ver Comentarios(0)

Nombre: *

Gráfica 89: Pantalla ver artículo.
Fuente: Propia.

Contáctenos

INFORMACIÓN CONTACTO

| Empresa | Dirección | Teléfono | Responsable | Email | Comunidad |
|-----------------|----------------------------|-----------|------------------|-----------------------------|----------------------------|
| Pakchapi Pakari | Comunidad de Cariacu | 022129255 | Germanico Guacan | ecoturismocariacu@gmail.com | Comunidad de Cariacu |
| Empresa | Dirección | Teléfono | Responsable | Email | Comunidad |
| Paquiestancia | Comunidad de Paquiestancia | (02) | Sr..... | admin@webmaster.com | Comunidad de Paquiestancia |



Gráfica 90: Pantalla contáctenos.
Fuente: Propia.

Buscador Interno

Resultados de la búsqueda 'Cariacu' = (15) encontrados

| Nombre | Descripción | etiqueta |
|-----------------------------|--|----------------------------------|
| Fiestas de Cariacu 2014 | Las fiestas anuales de la comunidad de Cariacu | fiesta, cariacu, 2014, San Pedro |
| Paisajes en las comunidades | en lo personal me parece muy interesante los paisajes que se pueden observar en las comunidades de Cariacu y Paquiestancia | paisaje, cariacu, Paisaje |

Gráfica 91: Pantalla de búsqueda.
Fuente: Propia.

Calendario de Eventos

08 EVENTOS

Próximo Evento a realizarse



CAMINO DEL CONDOR DESEA

FELIZ DÍA DE SAN VALENTÍN
A TODOS

| CONTACTO / EVENTO | |
|---------------------|---|
| Empresa: | gaby |
| Dirección: | los andes carchi |
| Teléfono: | 0997224215 |
| Correo Electrónico: | gabybenalcazar06@outlook.com |
| Sitio web: | http://wwwwgaby.com |

EVENTO

Autor: Joleida Guerrero

Gráfica 92: Pantalla de eventos.
Fuente: Propia.

- **Codificación del modelo:** Para que las vistas creadas tengan la funcionalidad requerida se implementa el código en las clases .php generadas por el framework Symfony. Las cuales se muestran en las siguientes tablas CRC.

| TARJETA CRC | | | |
|----------------------|---|--------------|--|
| Nombre Clase: | BaseContacto | | |
| Descripción: | Es la clase de abstracción de datos de la tabla contacto . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 79: Tarjeta CRC de la clase BaseContacto.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|--|
| Nombre Clase: | BaseEmpresaTuristica | | |
| Descripción: | Es la clase de abstracción de datos de la tabla empresa_turistica . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 80: Tarjeta CRC de la clase BaseCatalogo.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|--|
| Nombre Clase: | BaseFtRecursoMultimedia | | |
| Descripción: | Clase de abstracción de datos de la tabla ft_recurso_multimedia . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 81: Tarjeta CRC de la clase BaseFtRecursoMultimedia.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|--|
| Nombre Clase: | BaseFtMenu | | |
| Descripción: | Es la clase de abstracción de datos de la tabla ft_menu . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 82: Tarjeta CRC de la clase BaseFtMenu

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|--|
| Nombre Clase: | BaseFtModulo | | |
| Descripción: | Es la clase de abstracción de datos de la tabla ft_modulo . Acceso a base de datos. | | |
| | Responsabilidades | | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 83: Tarjeta CRC de la clase BaseFtModulo.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|----------------------|--|
| Nombre Clase: | BaseFtArticulo | | |
| Descripción: | Es la clase de abstracción de datos de la tabla ft_articulo . Acceso a base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 84: Tarjeta CRC de la clase BaseFtArticulo.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|----------------------|--|
| Nombre Clase: | BaseFtConfig | | |
| Descripción: | Es la clase de abstracción de datos de la tabla ft_config . Acceso a base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 85: Tarjeta CRC de la clase BaseFtConfig.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|----------------------|--|
| Nombre Clase: | BaseFtComentario | | |
| Descripción: | Es la clase de abstracción de datos de la tabla ft_comentario . Acceso a base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 86: Tarjeta CRC de la clase BaseFtComentario.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|----------------------|--|
| Nombre Clase: | BaseCalendarioEvento | | |
| Descripción: | Es la clase de abstracción de datos de la tabla calendario_evento . Acceso a Base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 87: Tarjeta CRC de la clase BaseCalendarioEvento.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|---|----------------------|---|
| Nombre Clase: | ContactoTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla contacto de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getContactoByld(\$id) | Obtener contacto por id de usuario | | |

Tabla 88: Tarjeta CRC de la clase ContactoTable.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|----------------------|---|
| Nombre Clase: | EmpresaTuristicaTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla empresa_turistica de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getFindByldEmpresa(\$id) | Retorna empresa por id. | | |
| getEmpresasDisponibles() | Retorna empresas disponibles | | |

Tabla 89: Tarjeta CRC de la clase EmpresaTuristicaTable.

Fuente: Propia.

| TARJETA CRC | | | |
|-------------------------------|--|---------------|---|
| Nombre Clase: | FtRecursoMultimediaTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla ft_recurso_multimedia de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getRecursoDisponible() | Retorna recursos disponibles | | |
| getRecursoPorId(\$id) | Retorna recursos por id | | |
| getRecursoByIdDisponible(\$i) | Obtiene recursos disponibles por id | | |
| getDeletePorId(\$id) | Elimina recurso por id. | | |

Tabla 90: Tarjeta CRC de la clase FtRecursoMultimediaTable.

Fuente: Propia.

| TARJETA CRC | | | |
|-------------------------|--|---------------|---|
| Nombre Clase: | FtMenuTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla ft_menu de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getMenusActivos() | Retorna ítem activos del menú | | |
| getAllMenus() | Retorna todos los ítems del menú | | |
| getInfoModulo(\$modulo) | Retorna la descripción del ítem. | | |

Tabla 91: Tarjeta CRC de la clase FtMenuTable.

Fuente: Propia.

| TARJETA CRC | | | |
|------------------------|--|---------------|---|
| Nombre Clase: | FtModuloTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla ft_modulo de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getModuloByNombre(\$m) | Retorna módulo por nombre. | | |

Tabla 92: Tarjeta CRC de la clase FtModuloTable.

Fuente: Propia.

| TARJETA CRC | | | |
|-----------------------------|--|---------------|---|
| Nombre Clase: | FtArticuloTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla ft_articulo de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getContenido() | Retorna artículos disponibles | | |
| getContenidoThisModulo(\$i) | Retorna artículos por id de módulo. | | |
| getBusqueda(\$frase) | Retorna artículo por frase búsqueda | | |
| getFindByIdArticulo(\$id) | Retorna artículo por id. | | |
| getFindByTitulo(\$titulo) | Retorna artículo por título. | | |
| getBuscarByIdEmpresa(\$id) | Retorna artículo por empresa. | | |

Tabla 93: Tarjeta CRC de la clase FtArticuloTable.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|---------------|---|
| Nombre Clase: | FtConfigTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla ft_config de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getEstiloActivo() | Obtiene estilo css activo | | |
| getParametroCfg(\$p) | Retorna registro por parámetro. | | |
| getUpdateDatos(\$p, \$v) | Actualiza valores de ítem por parámetros. | | |

Tabla 94: Tarjeta CRC de la clase FtConfigTable.

Fuente: Propia.

| TARJETA CRC | | | |
|------------------------------------|--|---------------|---|
| Nombre Clase: | FtComentarioTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla ft_comentario de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene una instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getFindByIdArticuloDisponible(\$i) | Retorna comentario de artículo. | | |
| getConfirm(\$mail, \$autor) | Actualiza estado de comentario. | | |

Tabla 95: Tarjeta CRC de la clase FtComentarioTable.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------------------|--|---------------|---|
| Nombre Clase: | CalendarioEventoTable | | |
| Descripción: | Clase generada por Symfony, para la abstracción de datos. Permite acceder a los datos de la tabla calendario_evento de la base de datos. Capa Modelo. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| getInstance() | Obtiene instancia de la tabla | Súper clase: | Doctrine_Table (Clase doctrine, representa una tabla) |
| getEventoDisponible() | Retorna eventos disponibles. | | |
| getEventosCalendarioByAutor(\$a) | Retorna eventos por autor | | |
| getByIdCalendario(\$id_cal) | Retorna evento por id. | | |
| getBusqueda(\$frase_b) | Retorna eventos por frase. | | |

Tabla 96: Tarjeta CRC de la clase CalendarioEventoTable.

Fuente: Propia.

- **Codificación de los controladores:** Para controlar el flujo de la información se implementan las clases .php generadas por el framework.

| TARJETA CRC | | | |
|----------------------|---|---------------|--|
| Nombre Clase: | busquedaActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la visualización de los resultados de las búsquedas. Capa Controlador. Módulo búsqueda. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Ejecuta pantalla de búsqueda. | Súper clase: | sfActions (Controlador genérico, clase de Symfony) |
| executeFindResult() | Ejecuta visualización de ítem de búsqueda | | |

Tabla 97: Tarjeta CRC de la clase busquedaActions.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|---|---------------|---|
| Nombre Clase: | contactenosActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la visualización de los datos de los contactos. Capa Controlador. Módulo contactenos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Ejecuta pantalla contactenos. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |

Tabla 98: Tarjeta CRC de la clase contactenosActions.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|---------------|---|
| Nombre Clase: | errorActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la visualización de la pantalla errores. Capa Controlador. Módulo error. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeError404() | Ejecuta pantalla de error. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeError503() | Ejecuta pantalla de error. | | |

Tabla 99: Tarjeta CRC de la clase errorActions.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|---------------|---|
| Nombre Clase: | eventoActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la creación, edición, eliminación y visualización de los eventos. Capa Controlador. Módulo evento. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Visualiza eventos de usuario. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeView() | Ejecuta visualización evento. | | |
| executeNew() | Para nuevo evento. | | |
| executeCreate() | Permite crear el nuevo evento. | | |
| executeEdit() | Permite editar evento. | | |
| executeUpdate() | Permite actualizar evento. | | |
| executeDelete() | Permite eliminar evento. | | |
| processForm() | Procesa crear, editar evento. | | |

Tabla 100: Tarjeta CRC de la clase eventoActions.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|---|---------------|---|
| Nombre Clase: | galeriaActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la visualización de la galería fotográfica. Capa Controlador. Módulo galería. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Ver pantalla de galería. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |

Tabla 101: Tarjeta CRC de la clase galeriaActions.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|---|---------------|--|
| Nombre Clase: | inicioActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la visualización de la pantalla de inicio del sistema. Capa Controlador. Módulo inicio. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Ver pantalla de inicio. | Súper clase: | sfActions (Controlador genérico, clase de Symfony) |

Tabla 102: Tarjeta CRC de la clase inicioActions.

Fuente: Propia.

| TARJETA CRC | | | |
|------------------------|---|---------------|---|
| Nombre Clase: | gastronomiaActions, turismoComunitarioActions, turismoCulturalActions, turismoAventuraActions. | | |
| Descripción: | Clases controladoras generadas por Symfony, permite controlar la visualización de los artículos, recursos y comentarios de cada uno de ellos. Capa Controlador. Módulos gastronomía, turismoComunitario, turismoCultural, turismoAventura. Estos módulos tienen las mismas funcionalidades. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Visualiza pantalla con artículos. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeArticulo() | Visualiza artículo elegido. | | |
| executeNewCommentArt() | Crea nuevo comentario. | | |
| executeConfirm() | Confirma comentario por email. | | |

Tabla 103: Tarjeta CRC de las clases para gestión de artículos.

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|---|---------------|---|
| Nombre Clase: | mapaSitoActions, noticiasActions | | |
| Descripción: | Clases controladoras generadas por Symfony, permite controlar la visualización del mapa del sitio y las noticias del Ecuador respectivamente. Capa Controlador. Módulos mapaSito, noticias. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Permite visualizar el mapa del sitio y las noticias nacionales respectivamente. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |

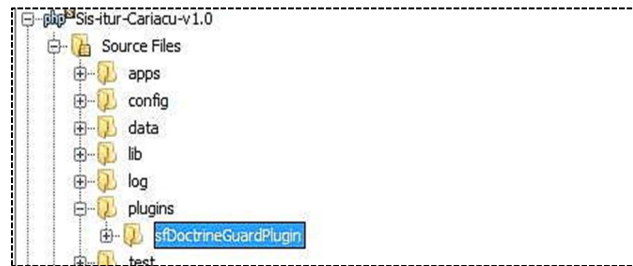
Tabla 104: Tarjeta CRC de las clases mapaSitoActions y noticiasActions.

Fuente: Propia.

Tarea 9: Desarrollo de parte administrativa. A continuación se describen las funcionalidades agregadas en la aplicación administrativa.

- **Creación de ventana de autenticación.** Para esto se instaló el plugin sfDoctrineGuardPlugin en el proyecto.

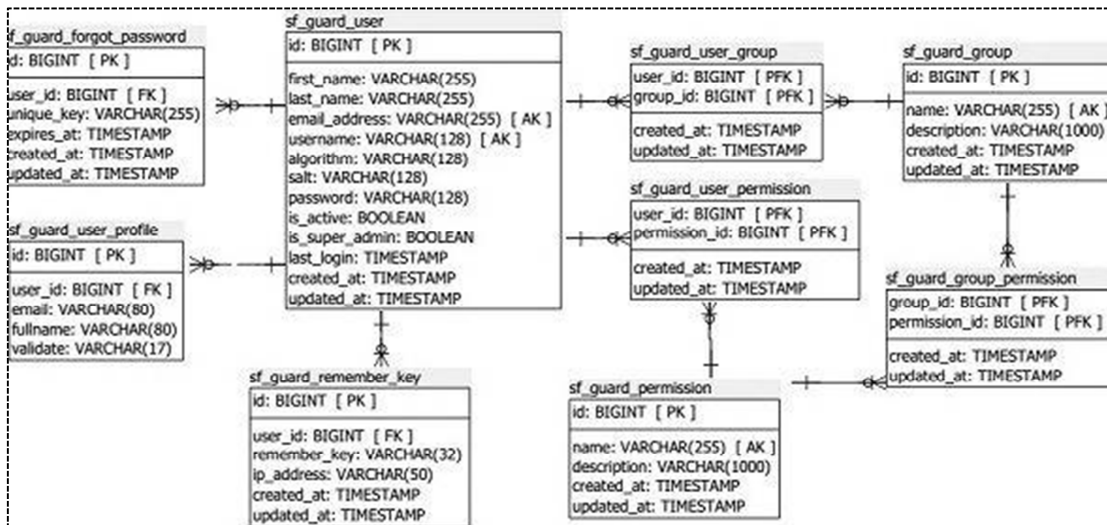
```
Clic derecho sobre proyecto → Symfony → Run Command → symfony
plugin:install sfDoctrineGuardPlugin → Run
```



Gráfica 93: Instalación de sfDoctrineGuardPlugin

Fuente: Propia.

Seguidamente se configura este plugin siguiendo el manual de <http://www.symfony-project.org/plugins/sfDoctrineGuardPlugin>, se generan las tablas en la base de datos.



Gráfica 94: Diagrama entidad-relación para el plugin sfDoctrineGuardPlugin.

Fuente: Propia.



Gráfica 95: Pantalla de autenticación de aplicación administrativa.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|---|----------------------|--|
| Nombre Clase: | BasesfGuardAuthActions | | |
| Descripción: | Clase controlador del plugin sfDoctrineGuardPlugin, controla el acceso a la aplicación sis_itur_admin que es de administración o backend. Capa Controlador. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeSignin() | Controla ingreso a sistema. | Súper clase: | sfActions (Controlador genérico, clase de Symfony) |
| executeSignout() | Salir de sistema y cerrar sesión. | | |
| executeSecure() | Controla seguridad de acceso. Valida la autenticación de usuarios. | | |

Tabla 105: Tarjeta CRC de la clase BasesfGuardAuthActions.

Fuente: Propia.

| TARJETA CRC | | | |
|--------------------------|--|----------------------|--|
| Nombre Clase: | BasesfGuardUser | | |
| Descripción: | Es la clase de abstracción de datos de la tabla sf_guard_user. Acceso a base de datos. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| setTableDefinition() | Definición de los datos de la tabla | Súper clase: | sfDoctrineRecord (Clase de ORM Doctrine) |
| setUp() | Referencia a claves primarias y foráneas | | |

Tabla 106: Tarjeta CRC de la clase BasesfGuardUser.

Fuente: Propia.

- **Creación de las vistas de la aplicación sis_itur_admin.** A continuación se muestran las pantallas creadas para la administración.



Gráfica 96: Pantalla de inicio de aplicación administrativa.

Fuente: Propia.

| Módulo Usuarios | | | | | | | | |
|--------------------------|----|-----------|--------------|------|-------------------------|-------------------|------------|-----|
| Usuario List | | | | | | | | |
| | Id | Nombre | Apellidos | Nick | Email | Dirección | Telefono | Gen |
| <input type="checkbox"/> | 15 | Nathaly A | Benalcazar G | naty | nathalyguerro@gmail.com | El olivo - Ibarra | 0959803658 | F |

Gráfica 97: Pantalla de módulo de gestión de usuarios.

Fuente: Propia.

| Módulo Foro | | | | | | | |
|-------------|-----------------|---|---------|-------------------------------|-----------|-------------------------------|---------|
| Foro List | | | | | | | |
| Id | Nombre foro | Descripción | Autor | Fecha creacion | Publicado | Palabras clave | Abierto |
| 30 | Comidas Tipicas | La gastronomía de la zona es muy variada, y | krlos81 | 17 de diciembre de 2013 15:34 | ✓ | gastronomía, comidas, bebidas | ✓ |

Gráfica 98: Pantalla de gestión de módulo foro.

Fuente: Propia.

| Módulo Contrataciones | | | | | | | |
|--------------------------------|-------------------|----------------------------|----------|------------|------|--------|------------------------------|
| Lista de Productos / Servicios | | | | | | | |
| Id | Nombre | Descripción | Publicar | Disponible | Tipo | Precio | Fotografía |
| 20 | Montura a caballo | Monturas para los caballos | ✓ | | 1 | 500 | uploads/catalogo/montura.jpg |

Gráfica 99: Pantalla de gestión de productos y servicios de módulo contrataciones.

Fuente: Propia.

| Módulo Contrataciones | | | | | | | |
|---|----|--------------------------|---------------------------|--------|--------|-------|--------------|
| Lista de Contrataciones(Estado:1 contratado, 2 reservado, 3 cancelado, 4 finalizado, 5 ninguno) | | | | | | | |
| Usuario | Id | Fecha Contrato | Fecha entrega visita | Estado | Activo | Visto | Total pedido |
| 1 | 7 | 6 de enero de 2014 00:00 | 15 de enero de 2014 00:00 | 4 | | ✓ | 753.76 |
| 1 | 6 | 6 de enero de 2014 00:00 | 15 de enero de 2014 00:00 | 3 | | ✓ | 190.4 |

Gráfica 100: Pantalla de gestión de contratos de módulo contrataciones.

Fuente: Propia.

| Módulo Visita virtual | | | | | | | |
|---------------------------------|------------------------------|---|--|-----------|------------|------------------------|---|
| Inventario de Visitas virtuales | | | | | | | |
| Id | Nombre | Descripción | Url | Publicado | Id empresa | Lugar | Miniatura |
| 24 | Centro Poblado Paquiestancia | Vista del centro poblado de la comunidad de Paquiestancia, donde se aprecia la escuela y la iglesia | /images/panoramicas/panorama paquiestancia.jpg | ✓ | 2 | Centro - Paquiestancia | /images/panoramicas/miniaturas/panorama paquiestancia.jpg |

Gráfica 101: Pantalla de gestión de módulo visita virtual.

Fuente: Propia.

Módulo Visita virtual

Agregar nueva Visita Virtual

Nuevo Tour Virtual

Nombre:(*)

Descripción:(*)

Ubicación:(*)

Imagen Panorámica:(*) No se ha seleccionado ningún archivo.

Miniatura:(*) No se ha seleccionado ningún archivo.

Publicado:(*) si no

Comunidad:(*)

Nota Importante

Este es el lugar donde puedes crear un nuevo Tour Virtual. Suba una imagen panorámica y podrá verla instantaneamente transformada en Tour Virtual Interactivo. El sitio admite todo tipo de imagenes de 360 grados, además de disparos de ojo de pez y panoramas parciales.

(*): son campos requeridos

Por favor solo suba imágenes de formato JPG, GIF o PNG, con otros formatos el tour no funcionará

Tamaño máximo de la imagen: 2Mb

El ancho máximo admitido no puede exceder los 8000 pixeles por alto 1500 pixeles

Gráfica 102: Pantalla de creación de visitas de módulo visita virtual.
Fuente: Propia.

Módulo Frontal

Artículo List

Filters

| ID | Título | Descripcion | Fecha publicado | Publicado | Modulo | Palabra clave | Empresa |
|----|------------------------|---|-----------------------------|-----------|--------|-----------------------------------|---------|
| 10 | Feria en Paquiestancia | Los fines de semana se realizan las ferias en la comunidad de paquiestancia, donde se promocionan los productos que se cosechan en la zona. | 22 de enero de 2014 00:00 | ✓ | 6 | feria, paquiestancia, gastronomia | 2 |
| 12 | Huertos Orgánicos | Los pobladores de la comunidad hacen uso de sus terrenos para la siembra de todo tipo de hortaliza, hortalizas. | 10 de febrero de 2014 00:00 | ✓ | 3 | huertos, familiares | 1 |

Gráfica 103: Pantalla gestión de artículos del módulo gestión de información.
Fuente: Propia.

Módulo Frontal

Lista de Recursos multimedia

Filters

| ID | Nombre | Upload url | Activo | Nombre archivo | Link | Visible | Descripcion |
|----|------------------------|--|--------|------------------------|-----------|---------|-------------------------------------|
| 1 | Cascada Cariacu | images/imagenes/Cascada de Cariacu.jpg | ✓ | Cascada de Cariacu.jpg | | ✓ | Cascadas de la comunidad de Cariacu |
| 10 | Entrada a las cascadas | images/imagenes/IMG_8806.JPG | ✓ | IMG_8806.JPG | identi.li | ✓ | Entrada a las cascadas de cariacu |

Gráfica 104: Pantalla gestión de recursos del módulo gestión de información.
Fuente: Propia.

Módulo Frontal

Subir Recursos Multimedia

Nombre Recurso

Descripción

Tipo Recurso

Seleccione Recurso No se ha seleccionado ningún archivo.

Disponible

Visible en portada:

Enlace(opcional)

Gráfica 105: Pantalla creación de recursos del módulo gestión de información.
Fuente: Propia.

Extras

Empresas List

Filters Rest

| Id empresa | Nombre empresa | Direccion empresa | Telefono empresa | Acciones |
|------------|-----------------|----------------------------|------------------|---|
| 1 | Pakchapi Pakari | Comunidad de Cariacu | 022129255 | <input type="button" value="Show"/> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> |
| 2 | Paquiestancia | Comunidad de Paquiestancia | (02) | <input type="button" value="Show"/> <input type="button" value="Editar"/> <input type="button" value="Borrar"/> |

Gráfica 106: Pantalla gestión de empresas turísticas.
Fuente: Propia.

Calendario de Eventos

Crear Nuevo Evento

Autor

Hora evento

Fecha evento

Lugar

Título

Detalle

Imagen No se ha seleccionado ningún archivo (tamaño : 500px ancho por 300px alto)

Publicado

Etiqueta

[Regresar a Lista](#)

Gráfica 107: Pantalla creación de eventos.
Fuente: Propia.



Gráfica 108: Pantalla gestión de reportes.
Fuente: Propia.



Gráfica 109: Pantalla de resultado de búsqueda.
Fuente: Propia.



Gráfica 110: Pantalla configuraciones extras.
Fuente: Propia.

- **Clases modelos y controladoras.** Estos módulos utilizan los archivos de la capa modelo descritos en las iteraciones anteriores, por otra parte los archivos de la capa controladora es creada automáticamente por el generador de administración del framework, los cuales se almacenan en el directorio *cache/sis_itur_admin/prod/modules/autoModule* del proyecto. Sin embargo el aplicativo necesita de funcionalidades especiales para lo cual se crean módulos adicionales y se implementan las clases controladoras de los mismos.

| Aplicación | Comando – Módulo |
|-----------------|---|
| sis_itur_admin. | symfony generate:module <i>sis_itur_admin addtocatalogo</i> symfony generate:module <i>sis_itur_admin addvisit</i> symfony generate:module <i>sis_itur_admin busqueda</i> symfony generate:module <i>sis_itur_admin contrataciones</i> symfony generate:module <i>sis_itur_admin evento</i> symfony generate:module <i>sis_itur_admin registro</i> symfony generate:module <i>sis_itur_admin report</i> symfony generate:module <i>sis_itur_admin upload</i> |

Tabla 107: Creación de módulos frontales

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|---|--------------|---|
| Nombre Clase: | addtocatalogoActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar creación, de productos, servicios del catálogo. Capa Controlador. Módulo addtocatalogo. | | |
| Responsabilidades | | Colaborador | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Procesa la pantalla de creación de producto y servicio. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeUpload() | Para agregar nuevo producto/servicio | | |

Tabla 108: Tarjeta CRC de la clase addcatalogoActions

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|---|
| Nombre Clase: | busquedaActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la generación de resultados de las búsquedas. Capa Controlador. Módulo búsqueda. | | |
| Responsabilidades | | Colaborador | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Procesa la pantalla de visualización de resultado de búsqueda. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeFindResult() | Visualiza resultado a detalle | | |

Tabla 109: Tarjeta CRC de la clase busquedaActions

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|---|
| Nombre Clase: | contratacionesActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar visualización de las nuevas contrataciones. Capa Controlador. Módulo contrataciones. | | |
| Responsabilidades | | Colaborador | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Permite visualizar todas las nuevas contrataciones. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeViewPedido() | Visualiza contratación seleccionada. | | |
| executePrint() | Permite generar documento .pdf de contrato. | | |

Tabla 110: Tarjeta CRC de la clase contratacionesActions

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|--------------|---|
| Nombre Clase: | eventoActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar creación, edición, actualización, eliminación de eventos. Capa Controlador. Módulo evento. | | |
| Responsabilidades | | Colaborador | Colaboradores |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Visualizar eventos no verificados. | Súper clase: | sfActions (Controlador genérico, clase Symfony) |
| executeView() | Visualizar evento seleccionado. | | |
| executeNew() | Para visualizar creación de nuevo evento. | | |
| executeCreate() | Ejecuta creación nuevo evento. | | |
| executeEdit() | Editar eventos. | | |
| executeUpdate() | Actualizar eventos. | | |
| executeDelete() | Eliminar eventos. | | |
| processForm() | Procesar creación y actualización de eventos. | | |

Tabla 111: Tarjeta CRC de la clase eventoActions

Fuente: Propia.

| TARJETA CRC | | | |
|----------------------|--|---------------|---|
| Nombre Clase: | registroActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar creación, actualización, eliminación de usuarios. Capa Controlador. Módulo registro. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Visualizar todos los usuarios. | Súper clase: | sfActions (Controlador genérico de Symfony) |
| executeNew() | Para visualizar creación de nuevo usuario. | | |
| executeCreate() | Ejecuta creación nuevo usuario. | | |
| executeEdit() | Editar usuarios. | | |
| executeUpdate() | Actualizar usuarios. | | |
| executeDelete() | Eliminar usuarios. | | |
| processForm() | Procesar creación y actualización de usuarios. | | |

Tabla 112: Tarjeta CRC de la clase registroActions

Fuente: Propia.

| TARJETA CRC | | | |
|---------------------------|---|---------------|---|
| Nombre Clase: | reportActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar búsqueda y generación de reportes. Capa Controlador. Módulo report. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Visualizar pantalla de inicio de reportes. | Súper clase: | sfActions (Controlador genérico de Symfony) |
| executeIndexuser() | Buscar usuarios. | | |
| executeIndexforo() | Buscar temas Foro. | | |
| executeIndexvisita() | Buscar Visita virtual. | | |
| executeIndexcontratos() | Buscar Contratos. | | |
| executeIndexarticulo() | Buscar Artículos. | | |
| executeIndexcatalogo() | Buscar producto de catálogo. | | |
| executeObtenerUser() | Resultado de búsqueda de usuarios. | | |
| executeObtenerForo() | Resultado búsqueda de temas de foro. | | |
| executeObtenerVisita() | Resultado de búsqueda de visita. | | |
| executeObtenerArticulo() | Resultado de búsqueda de artículo. | | |
| executeObtenerCatalogo() | Resultado de búsqueda de productos. | | |
| executeObtenerContratos() | Resultado de búsqueda de contratos. | | |
| executeGeneratePdf() | Generar documento .pdf de resultado. | | |

Tabla 113: Tarjeta CRC de la clase reportActions

Fuente: Propia.

| TARJETA CRC | | | |
|------------------------|---|---------------|---|
| Nombre Clase: | uploadActions | | |
| Descripción: | Clase controlador generada por Symfony, permite controlar la subida y eliminación de recursos multimedia al sitio. Capa Controlador. Módulo upload. | | |
| Responsabilidades | | Colaboradores | |
| Responsabilidad | Descripción | Colaborador | Descripción |
| executeIndex() | Visualizar pantalla de subida de recursos. | Súper clase: | sfActions (Controlador genérico de Symfony) |
| executeSubirRecursoM() | Controla subida de recursos. | | |
| executeDelete() | Elimina recursos del sitio | | |

Tabla 114: Tarjeta CRC de la clase uploadActions

Fuente: Propia.

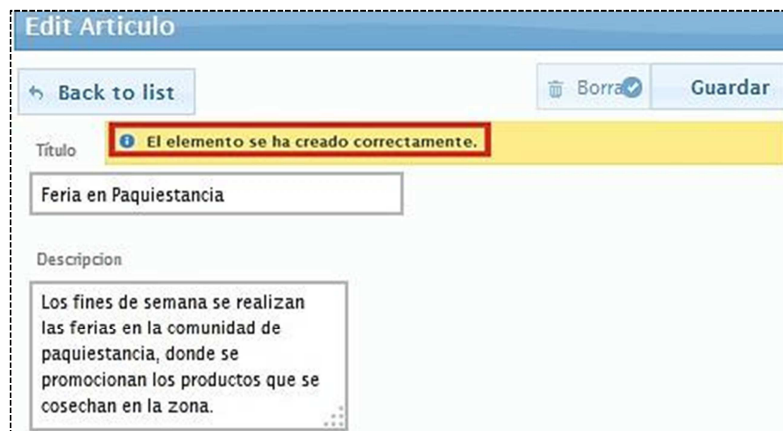
Tarea 10: Pruebas de funcionalidad.

Ingreso a la aplicación. Autenticarse para acceder a la parte de administración.

Gráfica 111: Prueba ingreso a aplicación administrativa.

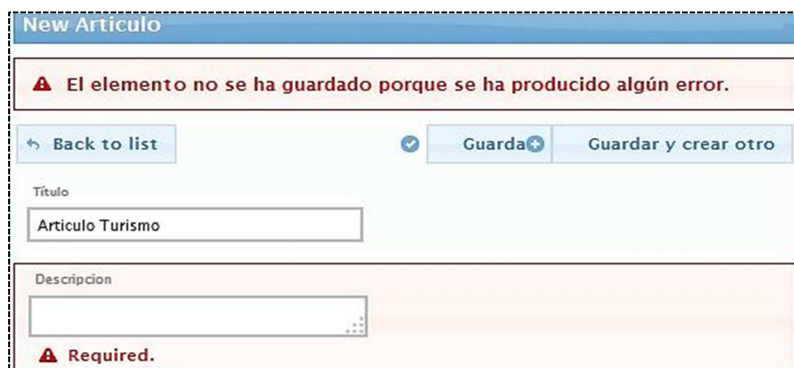
Fuente: Propia.

Gestionar información. El proceso de creación, edición y eliminación de la información es el mismo para todos los módulos administrativos.



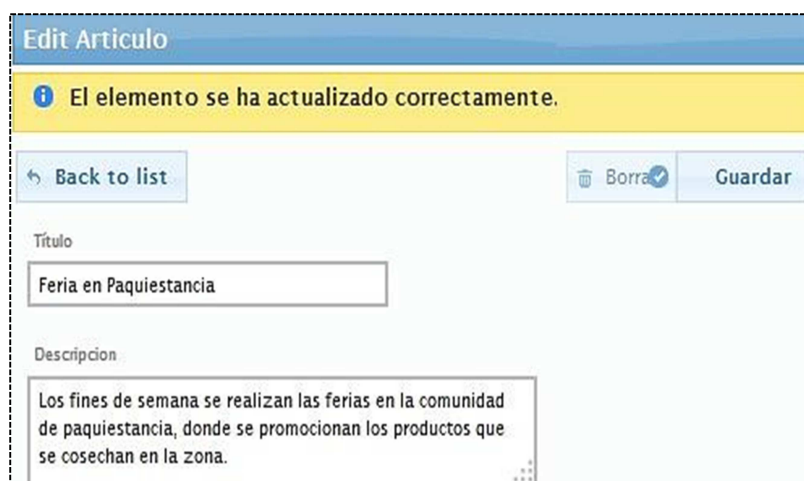
The screenshot shows the 'Edit Artículo' form. At the top, there is a blue header with the title 'Edit Artículo'. Below the header, there are three buttons: 'Back to list' (with a left arrow icon), 'Borra' (with a trash can icon and a checkmark), and 'Guardar'. A yellow banner with an information icon (i) contains the message 'El elemento se ha creado correctamente.'. Below this, there is a 'Título' field with the text 'Feria en Paquistancia'. Underneath is a 'Descripción' field with the text 'Los fines de semana se realizan las ferias en la comunidad de paquistancia, donde se promocionan los productos que se cosechan en la zona.'.

Gráfica 112: Prueba de creación de información.
Fuente: Propia.



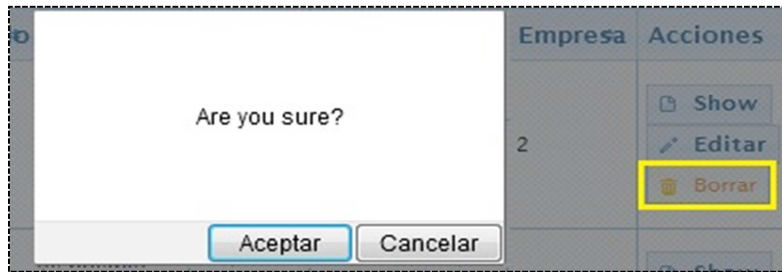
The screenshot shows the 'New Artículo' form. At the top, there is a blue header with the title 'New Artículo'. Below the header, there is a red banner with a warning icon (triangle) and the message 'El elemento no se ha guardado porque se ha producido algún error.'. Below this, there are three buttons: 'Back to list' (with a left arrow icon), 'Guarda' (with a checkmark and a plus sign), and 'Guardar y crear otro'. Below the buttons, there is a 'Título' field with the text 'Articulo Turismo'. Underneath is a 'Descripción' field which is empty. At the bottom of the description field, there is a red banner with a warning icon (triangle) and the message 'Required.'.

Gráfica 113: Prueba de edición errónea de información.
Fuente: Propia.



The screenshot shows the 'Edit Artículo' form. At the top, there is a blue header with the title 'Edit Artículo'. Below the header, there is a yellow banner with an information icon (i) and the message 'El elemento se ha actualizado correctamente.'. Below this, there are three buttons: 'Back to list' (with a left arrow icon), 'Borra' (with a trash can icon and a checkmark), and 'Guardar'. Below the buttons, there is a 'Título' field with the text 'Feria en Paquistancia'. Underneath is a 'Descripción' field with the text 'Los fines de semana se realizan las ferias en la comunidad de paquistancia, donde se promocionan los productos que se cosechan en la zona.'.

Gráfica 114: Prueba de edición correcta de información.
Fuente: Propia.



Gráfica 115: Prueba de eliminación de información.
Fuente: Propia.

Agregar nueva Visita Virtual

La transacción se realizó con éxito.

Nuevo Tour Virtual

Nombre: (*)

Descripción: (*)

Ubicación: (*)

Imagen Panorámica: (*) No se ha seleccionado ningún archivo.

Miniatura: (*) No se ha seleccionado ningún archivo.

Publicado: (*)

Comunidad: (*)

Gráfica 116: Prueba de creación de visitas virtuales.
Fuente: Propia.



Gráfica 117: Prueba de visualización correcta de artículos.
Fuente: Propia.



Gráfica 118: Prueba de creación de comentarios.
Fuente: Propia.

Finalización.

Tarea 11: Pruebas de aceptación. Pruebas de aceptación descritas en las historias.

| Prueba funcional | | | |
|--|---|---------------------------|-----------------------|
| Número Prueba: | 1 | Nro. Historia de Usuario: | 9 |
| Nombre de la Prueba: | Crear tablas para módulo de gestión de información.(frontend y backend) | | |
| Descripción: | Análisis y diseño de las tablas adecuadas en la base de datos bb_turismo. | | |
| Condiciones de ejecución: | Ninguna. | | |
| Entradas | | | |
| Caso | Resultado Esperado. | | Evaluación de Prueba. |
| Creación de tablas para módulo. | Tablas normalizadas, integridad referencial. | | Satisfactoria |
| Ingresar, actualizar, selección, eliminar registros. | Operación satisfactoria. | | Satisfactoria. |

Tabla 115: Prueba de aceptación de creación de tablas para módulo gestión de información.

Fuente: Propia.

| Prueba funcional | | | |
|--------------------------------------|---|---------------------------|-----------------------|
| Número Prueba: | 2 | Nro. Historia de Usuario: | 10 |
| Nombre de la Prueba: | Integración de módulos(frontend y backend) e ingreso autenticado al aplicativo "sis_itur_admin" | | |
| Descripción: | Integrar los módulos existentes y creados, gestionar información desde backend. Para acceder a la aplicación de administración es necesario que el usuario se autentifique. | | |
| Condiciones de ejecución: | Ninguna. | | |
| Entradas | | | |
| Caso | Resultado Esperado. | | Evaluación de Prueba. |
| Integrar Módulos | Módulos totalmente integrados. | | Satisfactoria. |
| Gestión dinámica de información. | Sitio dinámico. | | Satisfactoria. |
| Ingreso autenticado a la aplicación. | Validación de datos e ingreso satisfactorio. | | Satisfactoria. |
| Cerrar sesión de usuario. | Salir de aplicación y cerrar sesión. | | Satisfactoria. |

Tabla 116: Prueba de aceptación de integración de módulos e ingreso a backend.

Fuente: Propia.

Tarea 12: Integración. La tarea final es integrar los módulos creados.



Gráfica 119: Integración de módulos en aplicación frontal.

Fuente: Propia.



Gráfica 120: Integración de módulos en aplicación backend.
Fuente: Propia.

4.9. SPRINT 5: Desarrollo de módulo GIS

Finalmente se implementa el último módulo, el cual tiene la función de mostrar los sitios turísticos disponibles en las localidades a través de un GIS.

Planificación.

✓ Historias de usuario.

| HISTORIA DE USUARIO N° 11 | | | |
|---|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Estudio de herramientas de desarrollo adecuadas para la construcción del módulo. | | |
| ID de historia: | 11 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | MEDIA | Estimación: | |
| Iteración Asignada: | 6 | Dependencia: | |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 23 – 11 – 2013 |
| Descripción | | | |
| Realizar el estudio de las herramientas de desarrollo para que se pueda construir el módulo gis, estas deben ser de software libre. | | | |
| Pruebas de Aceptación | | | |
| Las herramientas deben ser Open Source, y deben permitir desarrollar la aplicación GIS, con el uso de archivos .shp. | | | |
| Observaciones | | | |
| Las herramientas deben permitir la integración entre ellas. | | | |
| Los archivos .shp se recopilarán desde las fuentes relacionadas al sitio. | | | |

Tabla 117: Historia de usuario N° 11.
Fuente: Propia

| HISTORIA DE USUARIO N° 12 | | | |
|---|---|---------------------|--------------------------------------|
| Proyecto: | Sistema web de gestión y promoción de los atractivos turísticos de la comunidad de Cariacu. | | |
| Nombre de la historia: | Análisis y desarrollo del Módulo Gis con información de los sitios turísticos de la zona. | | |
| ID de historia: | 12 | Usuario: | Sr. Germánico Guacán – Grupo turismo |
| Prioridad: | ALTA | Estimación: | |
| Iteración Asignada: | 6 | Dependencia: | |
| Programador Responsable: | Sr. Carlos Caluguillín | Fecha: | 23 – 11 – 2013 |
| Descripción | | | |
| El módulo debe presentar toda la información geo-referenciada disponible de los sitios turísticos en un mapa a través de un navegador. | | | |
| El mapa debe mostrar los puntos de interés de los turistas. | | | |
| El módulo debe dar la alternativa de consumir servicios web de mapas. | | | |
| Pruebas de Aceptación | | | |
| La aplicación debe estar disponible a través de la web y mostrar la información de los sitios turísticos en un mapa con las descripciones apropiadas. | | | |
| El mapa debe permitir acerca y alejar, además debe permitir la búsqueda de sitios turísticos. | | | |
| Observaciones | | | |
| En caso de disponer los equipos apropiados el aplicativo debe consumir datos desde una base de datos para habilitar la opción de edición y creación de puntos turísticos. | | | |

Tabla 118: Historia de usuario N° 12.

Fuente: Propia.

✓ **Pila de producto.**

A partir de las historias de usuarios se genera la pila de producto.

| PILA DE PRODUCTO | | |
|-------------------------|---|------------------|
| ID | Nombre Historia de Usuario | Prioridad |
| 11 | Estudio de herramientas de desarrollo adecuadas para la construcción del módulo. | MEDIA |
| 12 | Análisis y desarrollo del módulo gis con información de los sitios turísticos de la zona. | ALTA |

Tabla 119: Pila de producto 6

Fuente: Propia.

✓ **Pila de sprint.**

Las actividades a desarrollar en la construcción del módulo GIS se dividen en tareas pequeñas para poder implementarlas con mayor facilidad y para hacer un mejor seguimiento en su proceso.

Desarrollo.

Tarea 1: Estudio de herramientas. En esta tarea se hace un estudio de las herramientas seleccionadas para el desarrollo del módulo. Se estudia la estructura, composición y funcionamiento tanto del cliente pesado QGIS, el cliente liviano web PMAPPER, el servidor de mapas MapServer y la base de datos PostGIS.

Tarea 2: Instalación y configuración de herramientas (PostGIS, QGIS, MapServer, Pmapper).



Quantum GIS: Es el cliente pesado GIS que sirve para crear mapas a partir de conexiones a bases de datos, servicios web gis, archivos shapefile y otros, y exportar a MapServer. Este está disponible para plataformas Linux y Windows.

- Para entornos windows descargar la versión desde la página oficial <http://qgis.org/downloads/QGIS-OSGeo4W-2.0.1-3-Setup-x86.exe>, e instalar seleccionando las opciones adecuadas según las necesidades.
- Para entornos linux existen diferentes versiones de acuerdo a las distribuciones, para seleccionar la versión adecuada visitar el sitio oficial <http://www.qgis.org/es/site/forusers/alldownloads.html#linux> y seleccionar la adecuada. En este caso se instala en debian para lo cual seguir los siguientes pasos:

Agregar los repositorios en /etc/apt/source.list.

```
deb http://qgis.org/debian wheezy main
deb-src http://qgis.org/debian wheezy main
```

Actualizar el sistema.

```
#sudo aptitude update
```

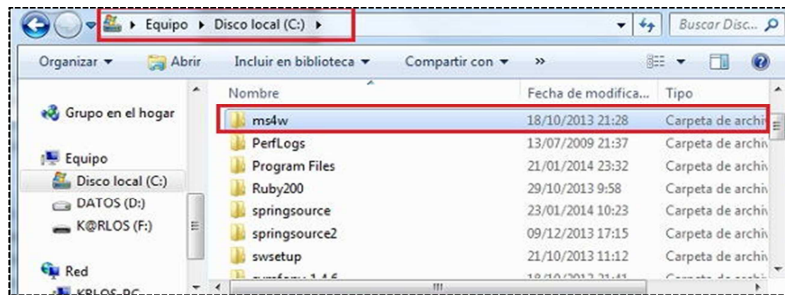
Ejecutar comando de instalación.

```
#aptitude install qgis python-qgis qgis-plugin-grass qgis-mapserver
```

Es importante recalcar que en la anterior línea comando se instale el soporte para MapServer ya que es un componente necesario para la construcción de los mapas.

MapServer: Servidor de mapas utilizado que permitirá la visualización del mapa construido. Disponible para plataformas linux, windows.

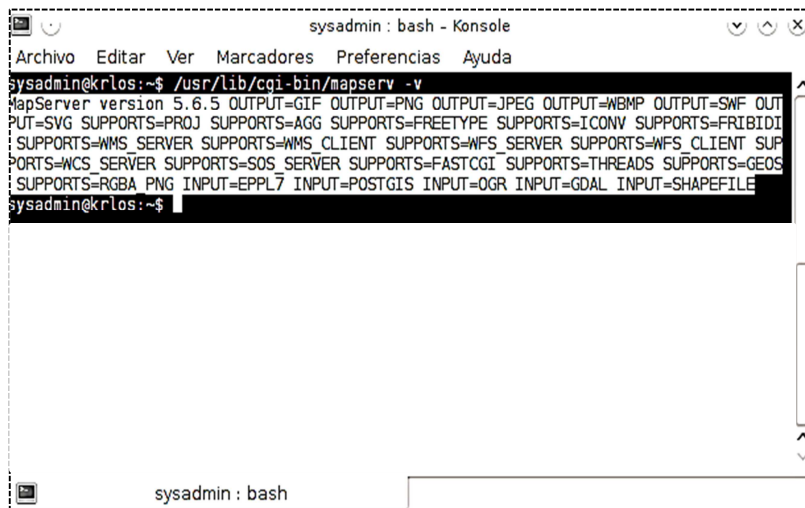
- Para instalar en windows descargar la versión .zip desde http://maptools.org/dl/ms4w/ms4w_3.0.6.zip, y descomprimir en el directorio raíz C: quedando como en siguiente gráfico.



Gráfica 121: MapServer en windows.
Fuente: Propia.

- Para linux existen varias alternativas: seleccionar la versión adecuada desde <http://mapserver.org/download.html#linux> e instalar siguiendo las recomendaciones expuestas en dicha página, en este proyecto se utilizaron los paquetes para el sistema operativo debian, para instalar se ejecutan los siguientes comandos:

```
$ sudo aptitude install cgi-mapserver mapserver-bin mapserver-doc php5-mapsript
```



Gráfica 122: MapServer en linux
Fuente: Propia.

Pmapper: Cliente liviano GIS que sirve para crear el visor del mapa. Compatible con el servidor de mapas MapServer.

- Para instalar en windows es necesario descargar el paquete disponible **ms4w** en la página oficial del proyecto pmapper: <http://www.pmapper.net/download.shtml>, y copiar su contenido en las carpetas correspondientes en C:/ms4w

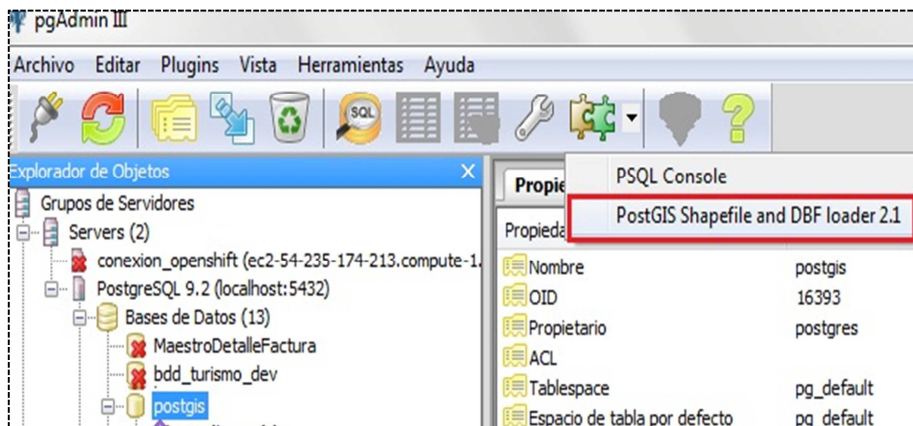
- Para instalar en otras plataformas descargar la versión **dev** desde el sitio <http://www.pmapper.net/dl/pmapper-dev.zip>, y descomprimir en el servidor web.

PostGIS: Base de datos con soporte a datos espaciales.

- Para instalar en windows descargar la versión apropiada desde el sitio <http://download.osgeo.org/postgis/windows>, y ejecutar el instalador.
- En linux existen varias formas de hacerlo, todo depende de la distribución. Ver en <http://postgis.net/install/>. En debian ejecutar como usuario root el comando: `#apt-get install postgis postgresql-x.x-postgis postgresql-doc`.

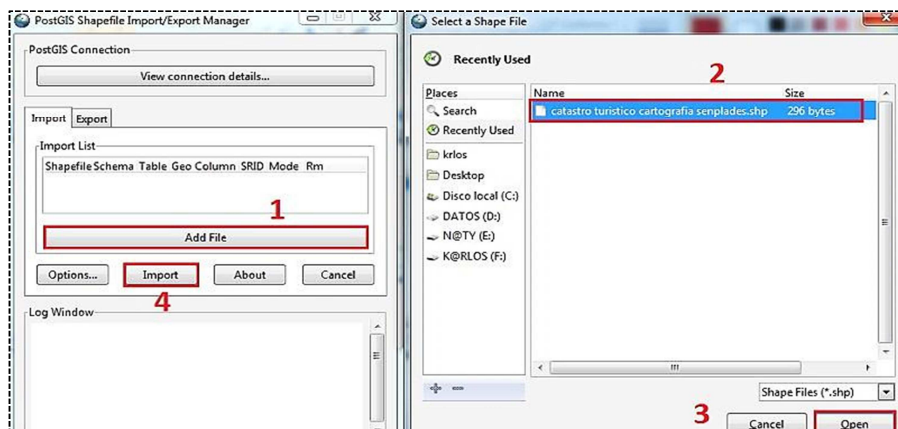
Tarea 3: Recolección, análisis y carga de datos geográficos. Los datos geográficos con información de los sitios turísticos son obtenidos en formato shapefile (.shp).

- Carga de datos en la base de datos espacial PostGIS desde archivo .shp con el uso del plugin de carga de archivos de PgAdminIII :



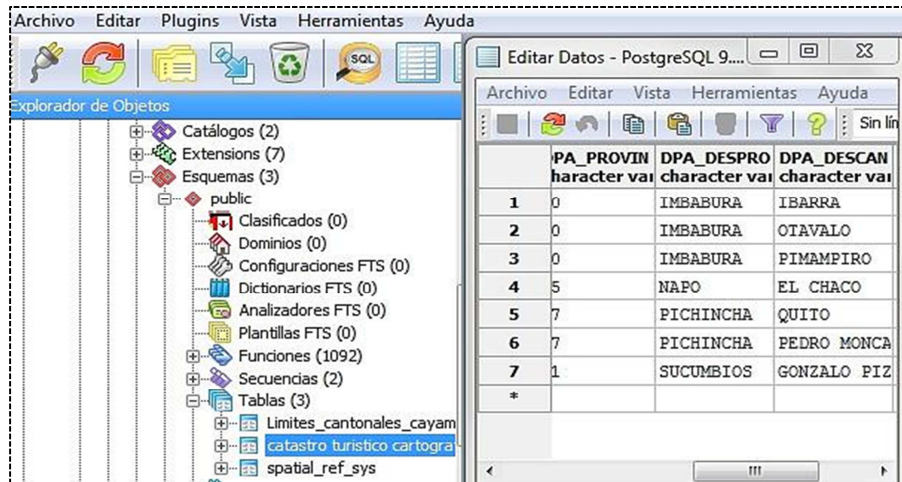
Gráfica 123: Carga de archivos .shp en PostGIS.

Fuente: propia.



Gráfica 124: Importar archivo .shp a base de datos PostGIS.

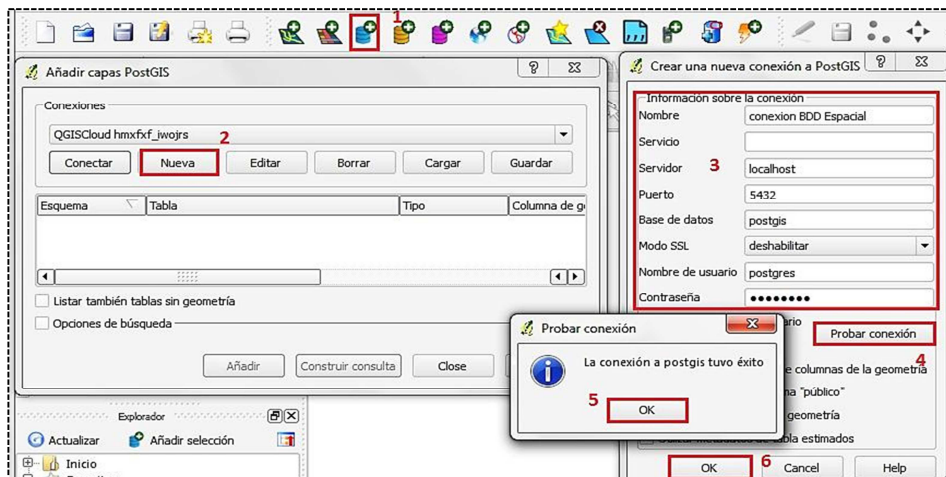
Fuente: propia.



Gráfica 125: Importación correcta de datos.
Fuente: propia.

Tarea 4: Creación de mapa en QGIS. Es indispensable diseñar y crear el mapa que se va a mostrar al turista, para ello se hace uso de la herramienta Quantum GIS.

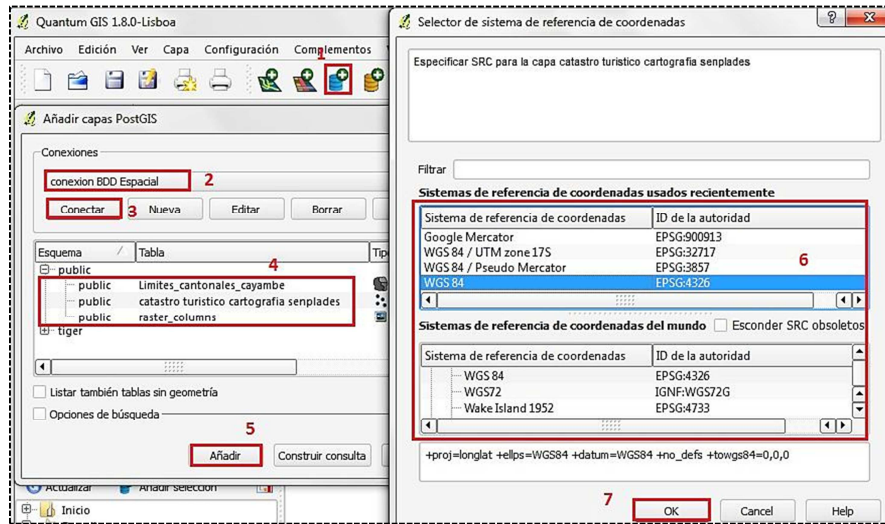
- Creación de conexión a base de datos PostGIS en QGIS.
 - Ejecutar Quantum GIS.
 - Seleccionar la opción “Añadir capas PostGIS”.
 - Clic opción “Nueva”.
 - Llenar los datos de la conexión.
 - Comprobar la conexión y clic en “OK”.
 - Presionar botón “Ok”.



Gráfica 126: Crear conexión a base de datos PostGIS.
Fuente: Propia.

- Añadir capas PostGIS a Quantum GIS.

- Seleccionar opción “Añadir capa PostGIS”.
- Seleccionar conexión.
- Clic en “Conectar”.
- Seleccionar tablas de la conexión.
- Clic en botón “Añadir”.
- Establecer el sistema de referencia de coordenadas.
- Finalizar presionando en botón “OK”.



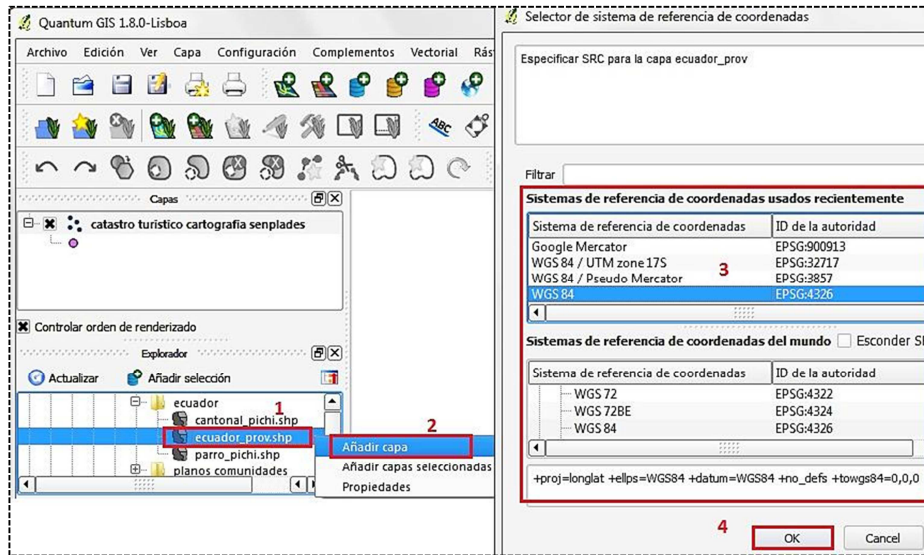
Gráfica 127: Agregar capa PostGIS a Quantum GIS
Fuente: Propia.

Visualización de capa desde base de datos PostGIS.



Gráfica 128: Visualizar capa PostGIS.
Fuente: Propia.

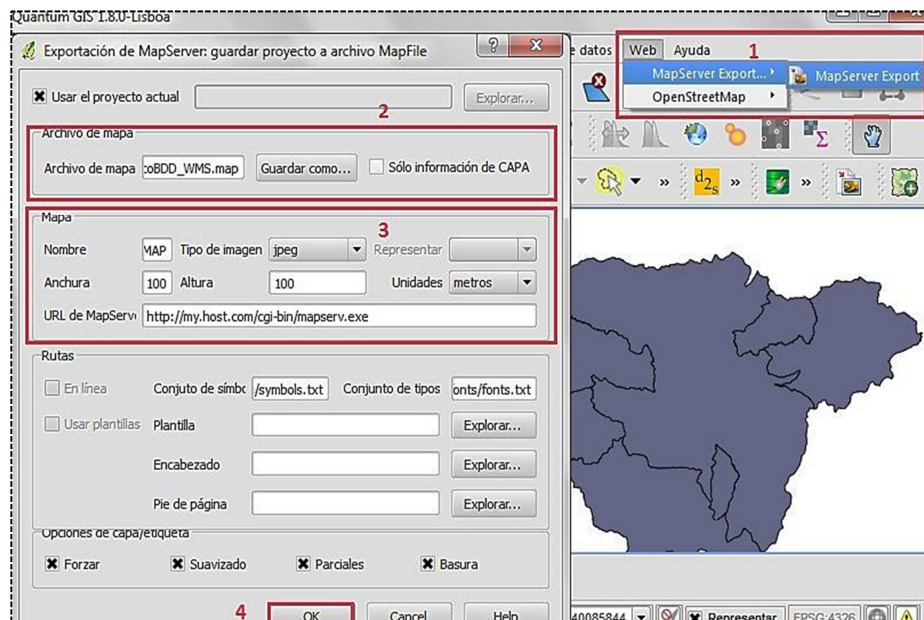
- Añadir capas desde archivos .shp al mapa.
 - Seleccionar archivo .shp en directorio.
 - Clic derecho sobre archivo y seleccionar “Añadir capa”.
 - Seleccionar el sistema de referencia de coordenadas.
 - Finalizar con clic en botón “OK”.



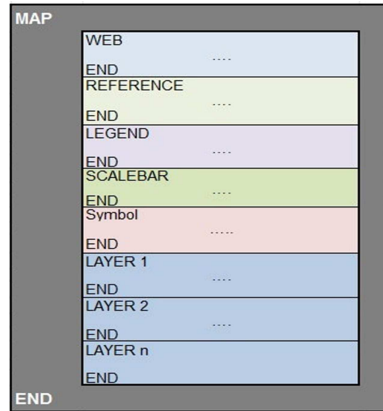
Gráfica 129: Añadir capa desde archivo .shp
Fuente: Propia.

Tarea 5: Exportación de mapa desde QGIS. Una vez creado totalmente el mapa se procede a exportarlo a un archivo .map, para lo cual se hace uso de la herramienta “MapServer Export” de QGIS.

- Clic menú “Web” → MapServer Export → MapServer Export.
- Seleccionar directorio donde se va a crear el archivo de mapa.
- Configurar parámetros de salida del archivo.
- Finalizar dando clic en botón “OK”.

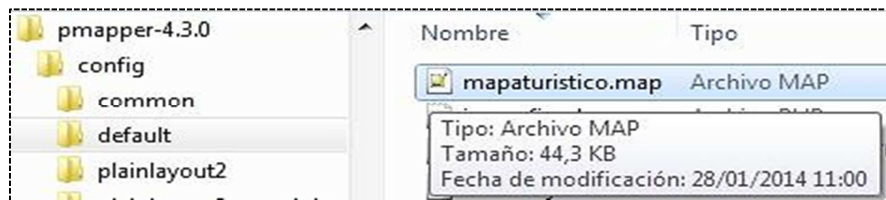


Gráfica 130: Exportar mapa a archivo .map
Fuente: Propia.

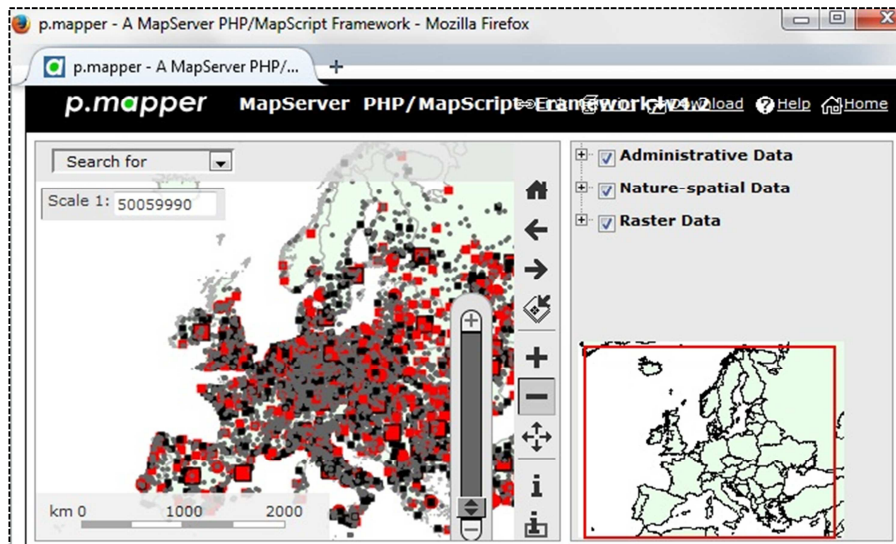


Gráfica 131: Estructura de archivo .map
Fuente: Propia.

Tarea 6: Implementación de mapa en el cliente liviano P.mapper. El mapa generado desde QGIS es el que se utiliza dentro del P.mapper, para ello se traslada dentro del directorio *pmapper-4.3.2/config/nombre_configuración/mapaturistico.map*.



Gráfica 132: Integración de nuevo mapa a Pmapper
Fuente: Propia.



Gráfica 133: Visor de mapas sin configurar.
Fuente: Propia.

Configuración del mapa y del visor.

- **Cabecera del mapa.** Definen las fuentes de datos, letras, símbolos, proyección.

```
MAP #Inicio del mapa
SIZE 1000 1000 #Tamaño del mapa
UNITS meters #Unidades de medida (metros)
#Extensión espacial del mapa
EXTENT 247931.913810 9427331.655388 1390416.920743 10181064.114063
SHAPEPATH "../../../datos_shapefiles" #directorio de archivos .shp
SYMBOLSET "../common/symbols/symbols-pmapper.sym" #Símbolos.
FONTSET "../common/fonts/msfontset.txt" #Tipo de Fuente.
RESOLUTION 96 #Resolución del mapa.
IMAGETYPE png #Tipo de Imagen
PROJECTION #Inicio de Proyección
    'proj=longlat'
    'datum=WGS84'
    'no_defs'
END
END #Fin de Mapa
```

- **Capa WEB.** Esta define el comportamiento de la interfaz web.

```
WEB #Inicio de la definición de la interfaz WEB.
TEMPLATE "map.html" #Plantilla de presentación de resultados.
IMAGEPATH "/home/www/tmp/" #Ruta de archivos temporales.
IMAGEURL "/tmp/" #Directorio base de IMAGEPATH.
END #Fin Web.
```

- **Capa REFERENCE.** Determina como se crearán las capas de referencia. Configura la visualización de la miniatura del mapa expuesto.

```
REFERENCE #Inicio de capa Reference.
#Extensión espacial de la imagen de referencia
EXTENT 247931.913810 9427331.655388 1390416.920743 10181064.114063
IMAGE "../../../images/ecuador.png" #Imagen miniatura de referencia.
SIZE 199 110 #Tamaño de la imagen en pixeles.
COLOR -1 -1 -1 #Color del cuadro de referencia de la imagen.
OUTLINECOLOR 255 0 0 #Color de líneas del cuadro de referencia.
END #Fin de capa Referencia.
```

- **Capa LEGEND.** Especifica cómo se construye la leyenda del mapa.

```
LEGEND #Inicio capa Legend
STATUS OFF #Crear imagen de leyenda de mapa
IMAGECOLOR 255 255 255 #Color inicial de la leyenda
POSITION ll #Posición de la leyenda en el mapa
KEYSIZE 18 12 #Tamaño de símbolos en pixeles.
KEYSPACING 10 5 #Espacio en símbolos y etiquetas.
TEMPLATE "void" #Plantilla para mostrar resultados.
LABEL #Inicio Label, para especificar título de etiqueta
    TYPE TRUETYPE #Tipo de fuente
    FONT "FreeSans" #Nombre Tipo letra.
    COLOR 0 0 89 #Color de la etiqueta
END #fin Label
END #Fin Legend
```

- **Capa SCALEBAR.** Determina como se crea la barra de escala del mapa.

```
SCALEBAR #inicio de Barra de la escala
STATUS off #Estado de la Barra de escala que se creó
TRANSPARENT off #Transparencia de la barra
INTERVALS 4 #Número de intervalos de la barra.
SIZE 200 3 #Tamaño de la barra.
UNITS kilometers #Unidad de medida.
LABEL #Inicio objeto label.
    COLOR 0 0 90 #Color de etiqueta.
    SIZE small #Tamaño de etiqueta.
END # Fin Label
END # Fin de Barra
```

- **Capa Symbol.** Determina el tipo de símbolos para las etiquetas.

```
.....
Symbol #Inicio objeto Symbol.
Name 'circle' #Nombre del símbolo.
Type ELLIPSE #Tipo de símbolo.
Filled TRUE #Rellena con color el símbolo.
Points #Secuencia de puntos que forma el símbolo.
  1 1
END #Fin secuencia puntos
END #Fin Symbol
.....
```

- **Capa LAYER.** Capa más usada ya que describe las capas del visor de mapas.

```
....
LAYER #Inicio de capa shp
NAME "division_provincial_ecuador" #Nombre de capa
TYPE polygon #Tipo de la capa.
DATA "division_provincial_ecuador.shp" #Nombre archivo .shp
PROJECTION #Proyección de la capa.
  "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs no_defs"
END #Fin de Proyección.
METADATA #Inicio Metadata. Almacenar datos para consultas.
  "DESCRIPTION" "Provincias_Ecuador"
  "RESULT_FIELDS" "DPA_PROVIN, DPA_DESPRO"
  "RESULT_HEADERS" "Codigo, Provincia"
  "ows_title" "division_provincial_ecuador"
END #Fin Metadata
LABELITEM "DPA_DESPRO" #Nombre de campo para la etiqueta
LABELMAXSCALE 12000000 #Tamaño de la etiqueta.
CLASS #Define clase para etiquetado
  Name 'division_provincial_ecuador' #nombre de la clase.
  COLOR 170 170 255 #Color de la etiqueta.
  LABEL #Inicio Label
    TYPE truetype #tipo de letra
    ENCODING "UTF-8" #codificación de letra.
    FONT FreeSans #nombre letra
    SIZE 2 #Tamaño letra.
  END #Fin Label
END #Fin Clase
END #Fin capa Layer
....
```

- **Capa con acceso a base de datos.** Establecer la conexión.

```
.....
LAYER #Inicio capa
NAME 'Limites_cantoniales' #Nombre capa
TYPE POLYGON #Tipo capa
EXTENT 247931.913810 9427331.655388 1390416.920743 10181064.114063 #Extensión Espacial
CONNECTIONTYPE POSTGIS #Tipo de conexión, conexión a base y consultas.
CONNECTION "host=localhost dbname=bdgis user=carlos password=abc port=5432"
DATA 'the_geom FROM "Limites_cantoniales" USING UNIQUE gid USING srid=4326'
.....
END #Fin Capa
.....
```

- **Configuraciones de visor de mapas.** Las principales configuraciones del visor de mapas de pmapper se realizan en el archivo `pmapper/pmapper-4.3.2/config/config_default.xml`.

- Cambiar título de Página: `<pmTitle>GIS | Cariacu</pmTitle>`.
- Configuración global: `<pm_config_location>default</pm_config_location>`
- Archivo mapa en uso: `<mapFile>mapaturistico.map</mapFile>`.
- Configuración de visualización de las capas:

```

<category name="cat_nacional">
  <group>division_provincial_ecuador</group>
</category>
<allGroups>
  <group>division_provincial_ecuador</group>
</allGroups>

```

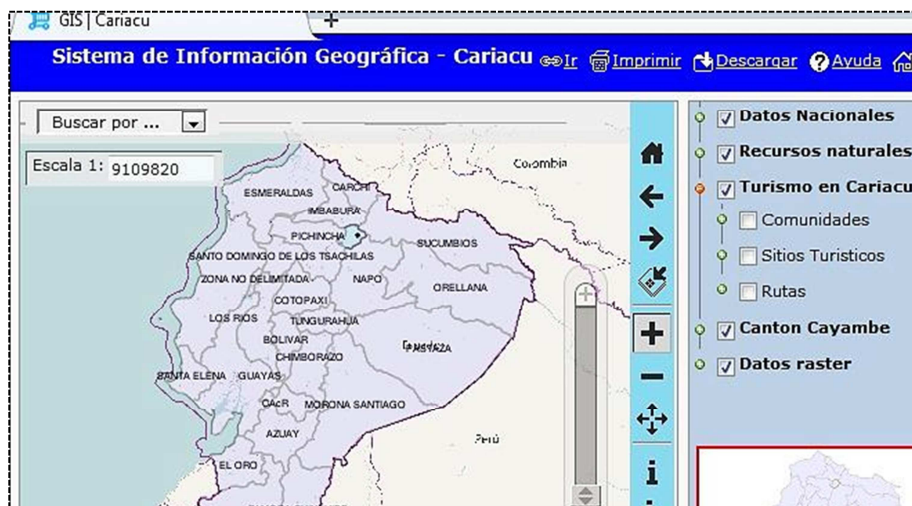
- Configurar lenguaje: `<defaultLanguage>es</defaultLanguage>`
- Configuración de consultas:

```

<searchlist version="1.0"> #Lista de Consulta
  <dataroot>${</dataroot> #Consulta Capa .shp
  <searchitem name="division_provincial_ecuador" description="Estate">
    <layer type="shape" name="division_provincial_ecuador"> #Consulta de provincias
      <field type="n" name="DPA_PROVIN" description="Provincia(1-24)" wildcard="0" />
    </layer>
  </searchitem>
</searchlist>

```

- Configurar plantilla de visor: La página principal del visor es totalmente configurable en el archivo `pmapper/pmapper-4.3.2/incphp/uitablement.php`



Gráfica 134: Visor de mapas completo.
Fuente: Propia.

Tarea 7: Pruebas de funcionalidad. A continuación las pruebas funcionales



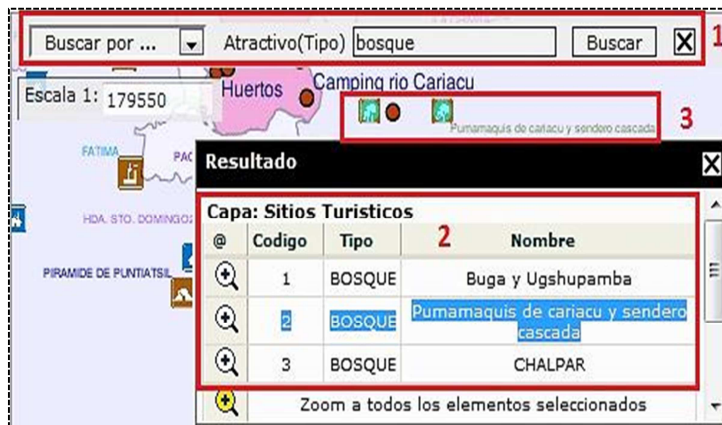
Gráfica 135: Prueba de funcionalidad de visualización de capas.
Fuente: Propia.

Prueba de visualización de información de objetos seleccionados del mapa.



Gráfica 136: Prueba de funcionalidad de selección de objetos del mapa.
Fuente: Propia.

Prueba de funcionalidad de búsqueda de datos en el mapa.



Gráfica 137: Prueba de funcionalidad de búsqueda de sitios en el mapa.
Fuente: Propia.

Prueba de funcionalidad de impresión de mapa.



Gráfica 138: Prueba de funcionalidad de impresión de mapas
Fuente: Propia.

Finalización.

Tarea 8: Pruebas de aceptación.

| Prueba funcional | | |
|--|--|------------------------------------|
| Número Prueba: | 1 | Nro. Historia de Usuario: 11 |
| Nombre de la Prueba: | Estudio de Herramientas de desarrollo GIS. | |
| Descripción: | Las Herramientas a utilizar deben ser de software libre y deben permitir la construcción de un GIS con archivos shapefile. | |
| Condiciones de ejecución: | Software libre. | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Uso de herramientas de desarrollo GIS de software libre. | Herramientas de desarrollo gratuitas. | Satisfactoria (MapServer, Pmapper) |
| Integración con archivos shapefile. | Disponibilidad y uso de archivos shapefile. | Satisfactoria. |

Tabla 121: Prueba de aceptación de estudio de herramientas GIS de software libre.

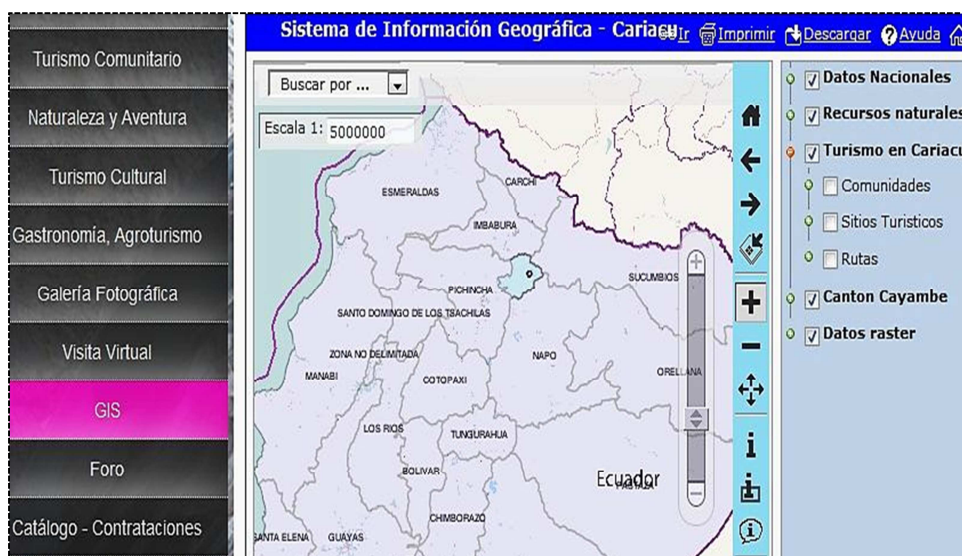
Fuente: Propia.

| Prueba funcional | | |
|--|---|------------------------------|
| Número Prueba: | 2 | Nro. Historia de Usuario: 12 |
| Nombre de la Prueba: | Visualización de sitios turísticos en GIS web | |
| Descripción: | La aplicación debe presentar los sitios turísticos de las comunidades a través de la web en un GIS. | |
| Condiciones de ejecución: | Aplicación totalmente configurada. | |
| Caso | Resultado Esperado. | Evaluación de Prueba. |
| Visualizar mapa de los sitios turísticos en un visor GIS web | Visor web con datos geográficos de los sitios turísticos. | Satisfactoria |
| Funcionalidad completa en el mapa (Zoom/Pan, elegir, buscar, imprimir) | Operación satisfactoria. | Satisfactoria. |

Tabla 122: Prueba de aceptación de visualización de sitios turísticos en GIS web.

Fuente: Propia.

Tarea 9: Integración de visor de mapa en el proyecto. El módulo GIS se integra al proyecto al incluir un vínculo en el menú principal.



Gráfica 139: Integración de módulo GIS en el sistema.

Fuente: Propia.

4.10. ENTREGABLES

✓ **Documentación de seguimiento del proyecto.**

Se entregan los siguientes documentos (disponible en CD):

- Proyecto Informático.
- Actas de reuniones.
- Actas de entrega- recepción de módulos.
- Acta de entrega – recepción del proyecto.
- Historias de usuarios.
- Pilas de producto.
- Pilas de Sprint.
- Pruebas de aceptación

✓ **Aplicativo.**

Los entregables del aplicativo son (disponibles en el CD):

- Sistema web: *Sis-itur-Cariacu-v1.0.zip*
- Módulo GIS: *pmapper.zip*
- Base de datos: *bdd_turismo_dev.backup*

✓ **Manuales.**

Se entrega los siguientes manuales sobre el sistema (disponibles en el CD):

- Manual técnico del sistema, y
- Manual de usuario.

CAPÍTULO V: CONCLUSIONES, RECOMENDACIONES, E

IMPACTO

✓ ANÁLISIS COSTO – BENEFICIO

Este análisis es realizado para establecer el costo en el cual se incurrió en el desarrollo del sistema y los beneficios que se obtendrán de este.

Por la parte de las herramientas de desarrollo el costo es cero, debido a que se estableció el uso de herramientas de software libre con lo cual se facilitó la implementación del sistema, esto implica de manera muy significativa que los beneficios frente a los costos sean altos.

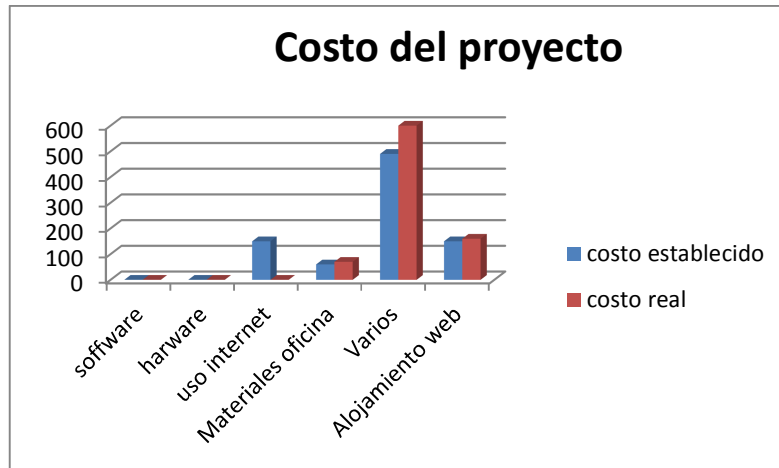
En relación a las herramientas de hardware usadas en el desarrollo del aplicativo, no se entraron en gastos ya que la entidad auspiciante cuenta con los equipos necesarios, esto también colaboró en la ejecución del proyecto.

A continuación se describen los aspectos del análisis de costo – beneficio del desarrollo del sistema.

| COSTO | |
|--|---------------|
| Software de desarrollo (Software libre) | 0 \$ |
| Costo de hardware (1 PC donada) | 0 \$ |
| Uso internet | 0 \$ |
| Materiales de oficina(copias, CDs y otros) | 70 \$ |
| Movilización, capacitación, anillados, empastados, imprevistos | 600 \$ |
| Alquiler de alojamiento web. (Dirección de turismo de Cayambe) | 160 \$ |
| Costo total de desarrollo: | 830 \$ |

Es importante mencionar que el costo de implementación del sistema tiene un pequeño incremento en comparación a lo establecido en el planteamiento del proyecto, y esto es debido a que los valores fueron establecidos con costos anteriores y que a comparación de los actuales existe un incremento considerable.

Un factor beneficioso en cuanto al costo, es el apoyo en el pago de varios de los aspectos descritos anteriormente por parte de la Dirección de Turismo del cantón Cayambe, ya que esto favoreció a la entidad auspiciante al no incurrir en ese gasto.

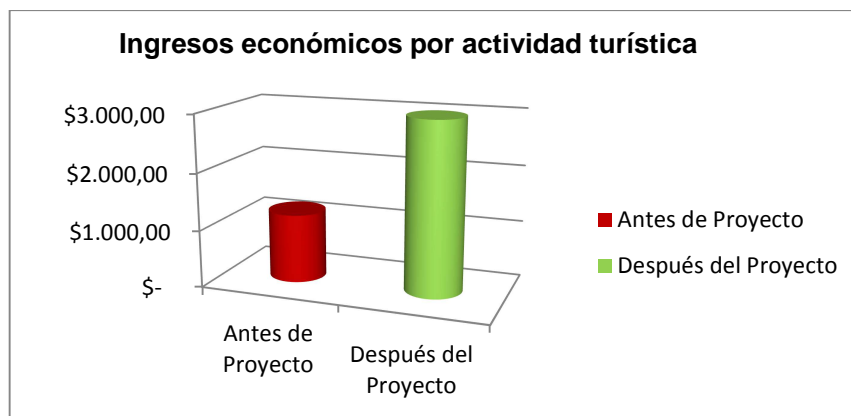


Gráfica 140: Costos de desarrollo del proyecto.
Fuente: Propia.

Para la recuperación de lo invertido el grupo de ecoturismo proyecta realizarlo en un año ya que las llegadas de los turistas son inminentes.

A continuación se describen los beneficios del proyecto.

| Beneficio |
|---|
| <ul style="list-style-type: none"> • En cuanto al uso de software libre ya que facilita el desarrollo por que no son necesarias las compras de las licencias de los programas. • Los turistas cuentan con información detallada de los sitios turísticos. Lo cual facilitará al momento de elegir un lugar para visitar. • La entidad auspiciante cuenta con un medio de difusión mundial lo que le facilitará la promoción adecuada de sus recursos turísticos con mayor efectividad ya que dejará de depender de las volantes en papel. Los ingresos económicos se ven reflejados con la llegada de los turistas que tienen más conocimiento del lugar. • Será más acogida la actividad turística en toda la zona de afluencia, ya que se dará a conocer todos los atractivos turísticos disponibles. |



Gráfica 141: Ingresos económicos por actividad turística.
Fuente: Propia.

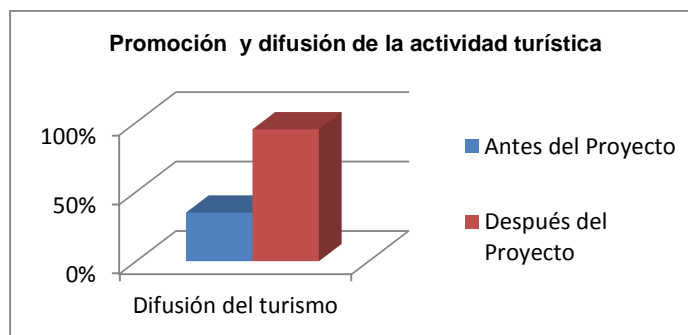
En la gráfica anterior se ilustra el incremento de los ingresos económicos debido a la eficiente promoción del turismo y por todo lo que contempla el proyecto turístico que impulsa el cantón Cayambe.

✓ IMPACTO

Los impactos del desarrollo del sistema web se detallan en los siguientes aspectos:

Aspecto social.

El desarrollo del sistema tiene un gran impacto social ya que se consigue la difusión efectiva de la actividad de la comunidad de Cariacu, esto incrementa los ingresos económicos de todas las familias de la zona, ya que de una u otra manera todas estas se benefician de la llegada de los turistas.



Gráfica 142: Promoción y difusión de la actividad turística.
Fuente: Propia.

Más acogida de los turistas ya que al disponer de la información de la zona, su preferencia hacia estos sitios aumenta paulatinamente.

La localidad da a conocer su tradición, costumbres, gastronomía y otros, a nivel nacional e internacional, además se abren nuevas opciones para la inversión externa.

Aspecto ambiental.

En este aspecto el desarrollo del sistema favorece en la conservación del medio ambiente, ya que anteriormente toda publicidad sobre las actividades turísticas se realizaban en papel, y en la actualidad esto ya no es necesario porque el sistema se encargará de publicitar esas actividades

Aspecto personal.

La implementación de este sistema percibe la satisfacción personal de haber ayudado a una comunidad que tanto necesitaba de un medio de difusión acorde a las tecnologías del momento. Además sirve para alcanzar una meta profesional.

✓ **CONCLUSIONES**

- Las comunidades rurales están experimentando un cambio en su actividad económica, ya que han visto alternativas turísticas muy eficientes que pueden suplir a su actividad económica tradicional.
- La promoción de la actividad turística es más eficiente con el uso de las nuevas tecnologías.
- Una aplicación web es una excelente opción al momento de seleccionar un medio de promoción de cualquier actividad económica, como el turismo.
- La información sobre los sitios turísticos está al alcance de toda persona con preferencia a ellos.
- Existe una gran variedad de herramientas de software libre para el desarrollo de aplicaciones con las características del presente proyecto.
- Las metodologías ágiles de desarrollo de software (SCRUM) son una gran ayuda al momento de entrar en el proceso de implementación ya que se adaptan constantemente a los requerimientos.
- Para una mejor organización y optimización del código fuente es imprescindible el uso de un framework en este caso Symfony, que establecen una arquitectura apropiada.
- Para la construcción de una aplicación GIS se disponen de herramientas innovadoras que facilitan su implementación ya que estas vienen prácticamente configuradas para el uso inmediato.
- La realidad virtual es un campo en pleno auge y más aún si se la utiliza para dar a conocer sitios desconocidos.
- Para la venta en línea se cuentan con catálogos de todos los tipos, los cuales son muy solicitados por los navegantes de la red.
- En la actualidad la web ha sufrido grandes cambios ya que esta busca la interacción dinámica con el usuario, por tal motivo, hoy en día las aplicaciones web se construyen con las mejores prácticas de desarrollo.
- Lenguajes de programación como HTML y PHP son muy útiles para crear este tipo de aplicaciones.

- Netbeans es un excelente IDE de desarrollo ya que tiene incorporado soporte para los lenguajes de programación web, así como también el soporte para framework como Symfony.
- El framework Symfony es una herramienta fantástica para realizar aplicaciones de forma rápida, ya que genera la mayoría de código y la organiza adecuadamente obedeciendo a patrones de diseño (MVC).
- MapServer y Pmapper son totalmente compatibles y configurables para el desarrollo de aplicaciones GIS.
- Seguir una metodología de desarrollo de software es muy importante ya que esta garantiza un software de calidad. Scrum dispone de un marco de trabajo muy flexible.
- Los módulos se desarrollaron por iteraciones, en las cuales se construyeron todos los requerimientos establecidos por el dueño del software.

✓ **RECOMENDACIONES.**

- Se recomienda a la comunidad de Cariacu seguir con la explotación de la actividad turística, ya que su localización geográfica les favorece porque son dueños de espectaculares paisajes naturales.
- Se recomienda seguir buscando nuevas alternativas turísticas que sean vistosas para los turistas. Y difundirla de la mejor manera.
- Recopilar más información sobre los sitios turísticos y promocionarlos.
- Se recomienda a los profesores y estudiantes hacer uso de metodologías ágiles de desarrollo de software, ya que con estos se producen software de calidad y de gusto del cliente.
- Se recomienda a la Institución educativa dar a conocer las herramientas más actuales en tanto a desarrollo de software.
- A los docentes de la carrera de sistemas hacer uso de herramientas de software libre para el aprendizaje.
- Se recomienda al grupo de ecoturismo de la comunidad de Cariacu brindar más ayuda a todos los estudiantes que acuden en su ayuda.
- Al administrador del sistema web, alimentar continuamente de información para que la promoción de la actividad sea más eficiente.
- Prestar mucha atención a las solicitudes de los usuarios en tanto a la petición de información, de productos, servicios y otros.

GLOSARIO DE TÉRMINOS

A

ASD: Metodología de desarrollo ágil de software, que incorpora el principio de adaptación continua.

Android: Sistema operativo basado en Linux para dispositivos móviles con pantallas táctiles.

ACID: (Atomicidad, Consistencia, Durabilidad, Aislamiento) Características de las bases de datos.

B

Brendan Eich: Programador de computadoras estadounidense que invento el lenguaje de programación JavaScript.

BSD: (Distribución de software Berkeley), es una licencia de software.

C

C: Lenguaje de programación desarrollado en 1972.

CGI: (Common Gateway Interface), es una tecnología usada en los servicios web.

CPU: (Unidad Central de Procesamiento), componente principal de un computador, que interpreta las instrucciones de un programa.

D

DSDM: (Dynamic Systems Development Method), metodología de desarrollo de sistemas dinámicos que proporciona un framework de desarrollo ágil.

DOM: (Document Object Model), es un modelo de objetos de documento que provee de un conjunto de estándares de objetos para representar documentos.

E

ESRI: (Environmental Systems Research Institute), empresa que desarrolla y vende software para Sistemas de Gestión Geográfica.

F

FDD: (Feature Driven Development), proceso de desarrollo de software iterativo e incremental.

FreeBSD: Sistema operativo libre para arquitecturas Intel, AMD.

G

GIS: (Geographic Information Systems), Sistema de Información Geográfica, que está compuesto de hardware, software, métodos, personas, datos geográficos, diseñado para la manipulación de los datos.

GNU/Linux: Combinación del sistema GNU con el núcleo de Linux. El código fuente es libre para su modificación y redistribución.

GNU/GPL: Licencia pública de GNU, su propósito es proteger al software libre de los intentos de apropiación.

GML: (Geography Markup Language), lenguaje de marcado geográfico, usado en el modelado, almacenamiento de los datos geográficos.

GDAL/OGR: (Geospatial Data Abstraction Library) para el procesamiento de varios formatos de datos geospaciales ráster y vectoriales.

GUI: (Graphical User Interface), programa que provee de una interfaz gráfica de usuario, a través de imágenes y objetos gráficos para representar la información en una interfaz.

H

HTTP: (HiperText Transfer Protocol), este es un protocolo empleado en los procesos de la web, usa el esquema solicitud-respuesta entre el cliente y el servidor.

HTML: (HiperText Markup Language), es un estándar que define la estructura básica y código de una página web.

J

Java: Es un lenguaje de programación derivado del lenguaje C y C++, orientado a clases y objetos.

JDK: (Java Development Kit), programa que proporciona herramientas de desarrollo para la creación de aplicaciones en java.

L

LTS: (Long Term Support), políticas de gestión del ciclo de vida en producto de software, especialmente en software de código abierto, que aplica los principios de la ingeniería de confiabilidad al proceso de desarrollo de software.

M

MVC: (Modelo Vista Controlador), es un patrón de diseño de software, el cual separa los datos de la aplicación, la lógica de negocio y de la interfaz del usuario en 3 diferentes capas.

MapScript: Es una interfaz de programación de aplicaciones para acceder a las funciones de MapServer a través del uso de los lenguajes de programación como java, php, perl, ruby, python, c#.

MySQL: Es un gestor de base de datos relacional, multiplataforma que se distribuye bajo la licencia pública de GNU. Desarrollado en lenguaje de programación C, C++

MVCC: (Multiversion Concurrency Control), control de concurrencia mediante versiones múltiples, es un método de control de acceso usado por los gestores de bases de datos.

O

ORM: (Object Relational Model), es una técnica de programación para trabajar con los datos, en la cual las tablas de una base de datos son tratadas como clases.

Open Source: término empleado para reconocer a un software como libre.

OSGeo: (Open Source Geospatial Foundation), organización no gubernamental que brinda soporte y promueve el desarrollo colaborativo de aplicaciones geospaciales y datos abiertos.

OGC: (Open Geospatial Consortium), es una organización internacional que establece estándares abiertos e interoperables entre los sistemas de información geográfica y la web.

P

PHP: es un lenguaje de programación del lado del servidor para la construcción de aplicaciones web dinámicas.

Python: Lenguaje de programación de interpretado que se basa en la premisa de código de sintaxis limpia y legible.

PERL: es un lenguaje de programación de propósito general estructurado en bloques como el lenguaje C.

PDO: (PHP Data Objects), extensión que implementa una capa de abstracción de acceso a datos para diferentes gestores de bases de datos.

Q

QSOs: (Qualification and Selection of Open Souce Software), es una metodología para la selección objetiva de software libre, desarrollado por la empresa francesa Atos Origin.

R

RAM: (Random Access Memory), memoria de acceso randómico para el trabajo de los sistemas operativos y software, en este dispositivo de cargan las instrucciones que son ejecutadas por el procesador.

S

SCRUM: Es una metodología de desarrollo de software ágil de procesos iterativos e incrementales.

Sprint: Es un periodo de tiempo repetitivo en el cual se producen los incrementos de un software.

SGBD: (Sistema Gestor de Base de datos), conjunto de programas que permiten la manipulación de los datos de una base de datos.

SQL: (Structured Query Language), es un lenguaje de consulta y acceso a las bases de datos relacionales.

SSL: (Secure Sockets Layer), protocolo de criptografía para las comunicaciones seguras dentro de una red.

T

TIN: (Triangulated Irregular Netwoks), estructura de datos digitales empleados en los sistemas de información geográficas para representar las superficies.

TCP/IP: Es el modelo base del internet, utilizado para enlazar computadoras que emplean diferentes sistemas operativos.

U

UNIX: Sistema operativo portable y multi hilos desarrollado en 1969 por la empresa AT & T.

Unicode: Estándar de codificación de caracteres. Diseñado para la manipulación de textos de múltiples lenguajes.

UTF-8: (8-bit Unicode Transformation Format), formato de codificación de caracteres, para representar cualquier tipo Unicode.

UTF-16: (16-bit Unicode Transformation Format), formato de codificación de caracteres Unicode, que utiliza símbolos de longitud variable.

URL: (Uniform Resource Locator), es una secuencia de caracteres que sigue un estándar para representar los recursos de internet, su localización e identificación.

W

W3C: (Word Wide Web Consortium), comunidad internacional que proporciona estándares para el crecimiento a largo plazo de la web.

WYSIWYG: (What You See Is What You Get), término que describe que lo que se ve es lo que se tiene, empleado en los procesadores de textos de desarrollo de aplicaciones web

X

XP: (eXtreme Programming), Metodología ágil de desarrollo de software que se basa en la adaptabilidad.

XML: (Extensible Markup Language), lenguaje de programación de marcas desarrollado por la W3C.

XSS: (Cross Site Scripting), agujero de inseguridad de las aplicaciones web, que permiten insertar código javaScript u otros lenguajes de script en las páginas web.

XLL: (eXtensible Linking Language), lenguaje de enlaces extendido, es un conjunto de parámetros utilizados en los enlaces entre documentos XML.

XHTML: (eXtensible Hipertext Markup Language), es un lenguaje de programación HTML que se lo expresa como XML válido.

XMLHttpRequest: (eXtensible Markup Language/Hypertext Transfer Protocol), es una interfaz utilizada para las peticiones http y https a servidores web.

BIBLIOGRAFÍA

LIBROS

- Álvarez García, A., De las Heras del Dedo, R., & Lasa Gómez, C. (2012). *Métodos Ágiles y Scrum*. Madrid, España: Ed. Anaya Multimedia.
- Bosque, J., Moreno, A., & Fuenzalida, M. (2012). *Sistemas de información geográfica y localización óptima de instalaciones y equipamientos* (Segunda ed.). Madrid: Alfaomega.
- Cibelli, C. (2012). *PHP programación Web avanzada para profesionales*. Madrid: Alfaomega.
- Crespi, M., & Planells, M. (2011). *Recurso turísticos*. Madrid: Síntesis.
- Pantaleo, G. (2011). *Calidad en el desarrollo de software*. Mexico, Alfaomega.
- Díaz, J. R. (Octubre de 2010). Las metodologías ágiles como garantía de calidad del software. (L. Fernández Sanz, Ed.) *Revista española de innovación, Calidad e Ingeniería de Software*, 5(3), 43.
- Kniberg, H., & Skarin, M. (2010). *Kanban and Scrum - Making the Most of Both (Paperback)*. (S. Agile, Trad.) United Kingdom: lulu.com.
- Martínez, J. (2012). *PostGis2: análisis espacial avanzado*. Madrid: Corporación para Investigaciones Biológicas.
- Matt, D. (2010). *PHP Práctico*. Madrid: ANAYA Multimedia.
- Matthew, D. (2011). *HTML 5*. Madrid: ANAYA Multimedia.
- Moreno, A. (2010). *Sistemas y análisis de la información geográfica : Manual de autoaprendizaje con ArcGIS* (Segunda ed.). Madrid: Alfaomega.
- Potencier, F., & Zaninotto, F. (2011). *The Definitive Guide to Symfony* (Tercera ed.). (J. Gilmore, Ed., & J. Eguiluz, Trad.) New York, Estados Unidos: Apress.
- Ramakrishnan, R., & Genrke, J. (s.f.). *Sistemas de gestión de bases de datos* (Tercera ed.). Madrid: McGraw-Hill.
- Roldán, D., & Valderas, P. (2010). *Aplicaciones web: Un enfoque práctico*. Madrid: Alfaomega.
- Sabana, M. (2010). *PHP con PostgreSQL 8*. Madrid: Megabyte.
- Schafer, S. (2010). *HTML, XHTML Y CSS*. Madrid: ANAYA Multimedia.
- Sommerville, I. (2011). *Ingeniería de software*. Madrid: PEARSON.
- Sutherland, J. (2 de Abril de 2012). *SCRUM LOG JEFF SUTHERLAND*. (Scruminc, Ed.) Obtenido de SCRUM LOG JEFF SUTHERLAND: jeffsutherland.com
- Van der Vlist, E. (2010). *Programación Web 2.0* (Primera ed.). (N. Bermúdez, Trad.) Madrid: ANAYA MULTIMEDIA.

Zaninotto, F., & Potencier, F. (2010). *A Gentle Introduction to Symfony 1.4*. (J. Eguiluz, Trad.) Guersey, United Kingdom: Sensio SA.

PUBLICACIONES EN LÍNEA.

Albaladejo, X. (s.f.). *Que es Scrum*. Recuperado el 12 de 06 de 2013, de Proyectos Ágiles: <http://www.proyectosagiles.org/que-es-scrum>

Anca Cisneros, W. (2010). *Proceso de Desarrollo: Metodología SCRUM*. Recuperado el 12 de 09 de 2013, de [www.wikispaces.com: http://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM](http://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM)

Blogspot. (s.f.). *Sistemas Gestores de Base de Datos*, Recuperado de <http://desarrollobasesdatos.blogspot.com/2011/02/sistema-gestor-de-bases-de-datos.html>, 2013.

Boundlessgeo. (s.f.). *Introduction to PostGIS*, Recuperado de: <http://workshops.boundlessgeo.com/postgis-intro/introduction.html>, 2013

Boundlessgeo. (s.f.). *Introducción PostGIS*, Recuperado de <http://workshops.boundlessgeo.com/postgis-intro/introduction.html>, 2013

Carto Virtual. (s.f.). *Modelo Ráster*, Recuperado de http://www.cartovirtual.es/aprendizaje/cursoTIG/modelo_raster.html#, 2013.

Clever Age. (s.f.). *Libro blanco sobre frameworks php para empresas*, Recuperado de: <http://fr.clever-age.com/veille/publications/livres-blancs/livre-blanc-%EF%BB%BF%EF%BB%BFframeworks-php.html>, 2013.

Emprender alia. (s.f.). *¿Deberías usar un framework para tu proyecto web?*, recuperado de <http://www.emprenderalia.com/deberias-usar-un-framework-para-tu-proyecto-web/>, 2013.

Ecuador Travel. (s.f.). *Documental versión española*, Recuperada de <http://discover.ecuador.travel/z1/potenciaturistica>, 2013.

EcuRed. (s.f.). *MapFile*, Recuperado de <http://www.ecured.cu/index.php/Mapfile>, 2013.

Eguiluz Perez, J. (7 de Junio de 2008). *Introducción a Ajax*. Recuperado el 17 de Agosto de 2013, de Libros Web: http://librosweb.es/ajax/capitulo_1.html

Doknos. (s.f.). *Symfony 1.4 en PHP Mejores Prácticas*. Recuperado de <http://www.doknos.com/node/27>, 2013

Geoespacial Perú. (s.f.). *Esquema de arquitectura de Servidor de Mapas*, Recuperado de http://www.geoespacialperu.com/index.php?option=com_content&view=article&id=112:cinc-o-servidores-de-mapas&catid=42:web-mapping, 2013

GeoTux. (s.f.). *Comparación clientes ligeros web*, Recuperado de: <http://geotux.tuxfamily.org/index.php/es/component/k2/item/291-comparacion-clientes-web-v6>, 2013.

GeoTux. (s.f.). *Algunas configuraciones básicas de pmapper*. Recuperado de <http://geotux.tuxfamily.org/index.php/es/component/k2/item/252-algunas-configuraciones-b%C3%A1sicas-en-pmapper-faq>, 2013.

Html en español. (s.f.). *La sintaxis básica de Css*, Recuperado de <http://es.html.net/tutorials/css/lesson2.php>, 2013.

- Isla Virtual. (s.f.). *Diferencia entre SCRUM y XP*, Recuperado de http://www.islavisual.com/articulos/desarrollo_web/diferencias-entre-scrum-y-xp.php, 2013.
- Libros web. (2013), *Capítulo 1. Introducción a AJAX*. Recuperado de http://librosweb.es/ajax/capitulo_1.html, 2013
- Libros web. (s.f.). *El patrón MVC*, Recuperado de: http://librosweb.es/symfony_1_4/capitulo_2/el_patron_mvc.html, 2013
- Libros web. (s.f.). *Soporte de CSS en los diferentes navegadores*, Recuperado de, http://librosweb.es/css/capitulo_1/soporte_de_css_en_los_navegadores.html, 2013.
- Mappinggis. (s.f.). *Por qué utilizar PostGIS*, Recuperado de <http://mappinggis.com/2012/09/por-que-utilizar-postgis/>, 2013.
- Mapserver. (s.f.). *Acerca de MapServer*, Recuperado de <http://mapserver.org/es/about.html>, 2013.
- Maptools. (s.f.). *How to install FGS*. Recuperado de <http://www.maptools.org/fgs/index.phtml?page=install.html>, 2013.
- Ministerio de Turismo. (08 de Mayo de 2008). *Ecuador entre los mejores destinos para Turismo de Lujo en 2013*. Recuperado el 21 de Agosto de 2013, de Amazonía Vive: http://www.amazoniavive.com/digital/index.php?option=com_content&view=article&id=3593
- Ministerio de turismo. (s.f.). *La experiencia turística en el Ecuador*. Recuperado de http://servicios.turismo.gob.ec/index.php?option=com_content&view=article&id=30, 2013.
- Netbeans Org. (s.f.). *Netbeans 7.0.1 download*. Recuperado de: <https://netbeans.org/downloads/7.0.1/index.html>, 2013
- Optur. (s.f.). *Estadísticas del turismo en el Ecuador*, Recuperado de <http://www.optur.org/estadisticas-turismo.html>, 2013.
- Osgeo. (s.f.). *Estándares del Open Geospatial Consortium*, Recuperado de <http://live.osgeo.org/es/standards/standards.html>, 2013.
- Php Frameworks. (s.f.). *Frameworks PHP*, Recuperado de <http://www.phpframeworks.com/>, 2013.
- Pmapper. (s.f.). *A MapServer PHP/MapScript Framework*, Recuperado de <http://www.pmapper.net/>, 2013.
- Potencier, F., & Zaninotto, F. (2011). *Symfony 1.4, la guía definitiva*. Recuperado el 14 de Mayo de 2013, de Libros Web: http://librosweb.es/symfony_1_4/
- Postgresql Org. (s.f.). *Componentes Principales de SGBD PostgreSQL*, Recuperado de http://www.postgresql.org/es/sobre_postgresql, 2013.
- PostGis. (s.f.). *About PostGIS*. Recuperado de <http://postgis.net/>, 2013.
- QGis. (s.f.). *Acerca de QGIS*. Recuperado de <http://www.qgis.org/es/site/about/index.html>, 2013.
- Redes de datos. (s.f.). *Servidor Web*, Recuperado de <http://redesdedatos-cesarito1023.blogspot.com/p/fundamentos-de-los-servidores-y-el.html>, 2013
- Scrummanager. (s.f.). *Marco Estándar SCRUM*, Recuperado de http://www.scrummanager.net/bok/index.php?title=Marco_est%C3%A1ndar_de_Scrum, 2013.
- Symfony en español. (s.f.). *NetBeans ya incluye soporte para Symfony*, Recuperado de: <http://symfony.es/noticias/2009/10/05/netbeans-ya-incluye-soporte-para-symfony/>, 2013

TuxPuc. (s.f.). *Comparativa de frameworks en PHP: CakePHP, Symfony y Zend Framework*, Recuperado de: <http://tuxpuc.pucp.edu.pe/articulo/comparativa-de-frameworks-en-php-cakephp-symfony-y-zend-framework>, 2013.

Wikipedia. (s.f.). *Turismo Comunitario*, Recuperado de http://es.wikipedia.org/wiki/Turismo_comunitario, 2013

wikipedia. (9 de Enero de 2013). *Turismo Comunitario*. Recuperado el 05 de Mayo de 2013, de Wikipedia: http://es.wikipedia.org/wiki/Turismo_comunitario

Wikispaces. (s.f.). *Arquitectura de Sistema Gestor de Base de datos*, Recuperado de <http://bdiesgc.wikispaces.com/Arquitectura+de+un+Sistema+Gestor+de+Bases+de+Datos>, 2013.

Wikispaces. (s.f.). *Metodología Scrum*, Recuperado de <http://procesosdesoftware.wikispaces.com/METODOLOGIA+SCRUM>, 2013.

360facil. (s.f.). *Tours Virtuales de 360 Grados con su propia cámara*. Recuperado de <http://www.360facil.com/esp/home.php>.

ANEXOS

Contenido (DISPONIBLE EN CD)

- Actas de reuniones.
- Actas de entrega – recepción.
- Historias de usuarios.
- Pilas de productos.
- Pilas de Sprint
- Manuales
- Varios

