



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN**

**“Trabajo de grado previo a la obtención del Título de Ingeniero en Sistemas  
y Computación”**

**TRABAJO DE GRADUACIÓN**

**Título del proyecto**

**ANÁLISIS DE FRAMEWORK PHP OPENSOURCE EN EL DESARROLLO  
DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE REPORTES  
ESTADÍSTICOS EN EL SISTEMA DE AGUA POTABLE DE LA PROVINCIA  
DE CHIMBORAZO.**

**Autor:**

Juan Carlos Moyota Chafra

**Directora:** Ing. Anita Congacha

**Riobamba – Ecuador**

**2015**

## PÁGINA DE REVISIÓN

Los miembros del Tribunal de Graduación del proyecto de investigación de título “Análisis de Framework PHP opensource en el desarrollo de una aplicación web para la gestión de reportes estadísticos en el sistema de agua potable de la provincia de Chimborazo” presentado por: Juan Carlos Moyota Chafra y dirigida por: Ing. Anita Congacha.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Fernando Molina



Presidente del Tribunal

Firma

Ing. Ana Congacha



Miembro del Tribunal

Firma

Ing. Danny Velasco



Miembro del Tribunal

Firma

## AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Juan Carlos Moyota Chafra y de Ing. Anita Congacha quien es la Directora del Proyecto; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”

Autor:



Juan Carlos Moyota

Directora del proyecto:



Ing. Anita Congacha

## **AGRADECIMIENTO**

Un agradecimiento muy especial a mis padres: Carmen y Rodrigo que a pesar de todos los inconvenientes han estado conmigo apoyándome en todas las etapas de mi formación profesional, a mis hermanos: Albita, Holger y Jairo a quienes trato dar ejemplo y que con perseverancia y esfuerzo todo se puede en la vida.

A la ingeniera Anita Congacha quien ha sido mi guía durante toda la realización de este trabajo de investigación, de manera muy especial a la ingeniera Lida Barba quien fué la persona promotora durante un tiempo pero que lamentablemente no pudo continuar guiándome, a la ingeniera Lady Espinoza en fin a todas y cada una de las personas que han aportado un granito de arena para con mi carrera.

*Juan Carlos Moyeta*

## **DEDICATORIA**

El presente trabajo de investigación se lo dedico a mi familia Moyota Chafra quienes han sido un pilar fundamental en la vida, una razón para continuar con mi formación profesional, personal y espiritual.

A mi padre por haberme ayudado a levantarme cuando caí, a mi madre por haberme dado tantos consejos de los que puedo recordar y a mis hermanos por darme razones para continuar luchando en este largo camino que es la vida.

También a mis abuelitos Félix y María por darme consejos como si fuesen mis propios padres, y a toda mi familia quienes de una u otra manera me brindaron sus palabras de aliento y ganas de luchar.

*Juan Carlos Moyota*

## INDICE GENERAL

Indice de tablas.....	vi
Índice de gráficos e ilustraciones .....	viii
Resumen.....	xii
Summary .....	xiii
Introducción .....	1
Capítulo I.....	3
Marco referencial .....	3
1.1.  Identificación y descripción del problema .....	3
1.2.  Formulación del problema.....	5
1.3.  Objetivos .....	5
1.3.1.  Objetivo general .....	5
1.3.2.  Objetivos específicos .....	5
1.4.  Justificación.....	5
1.5.  Antecedentes del tema.....	6
1.6.  Hipótesis.....	6
Capítulo II.....	7
Fundamentación teórica .....	7
2.1.  Arquitectura cliente-servidor.....	7
2.1.1.  Separación de funciones.....	8
2.2.  Aplicaciones informáticas .....	9
2.3.  Aplicaciones web .....	9
2.3.1.  Atributos de una aplicación web .....	10
2.3.2.  Características de una aplicación web.....	10
2.3.3.  Arquitectura de las aplicaciones web.....	11
2.3.4.  Tecnologías para la creación de aplicaciones web.....	12

2.3.5.	Tipos de aplicaciones web .....	14
2.4.	PHP (PHP Hypertext Preprocessor) .....	15
2.4.1.	Breve historia de PHP .....	15
2.4.2.	Características de PHP .....	16
2.4.3.	Ventajas y desventajas .....	17
2.4.4.	Razones para usar php en proyectos .....	17
2.4.5.	Popularidad de PHP .....	18
2.5.	Framework.....	20
2.5.1.	Características de un framework.....	21
2.5.2.	Clasificación de los framework.....	21
2.5.3.	Importancia de un framework .....	22
2.5.4.	Arquitectura de un framework .....	23
2.5.5.	Componentes de un framework .....	26
2.6.	Frameworks para la creación de aplicaciones web.....	27
2.7.	Frameworks para PHP.....	31
2.8.	Estimación del software .....	33
2.8.1.	Modelo COCOMO.....	34
2.8.2.	Reglas para el conteo de líneas de código.....	36
Capítulo III.....		37
Evaluación y selección del Framework PHP .....		37
3.1.	Identificación de las herramientas .....	37
3.2.	Descripción de los frameworks .....	39
3.2.1.	CakePHP .....	39
3.2.2.	CodeIgniter.....	41
3.2.3.	Yii.....	44
3.3.	Indicadores usados en la selección del framework.....	47

3.3.1.	Indicador 1: Madurez .....	48
3.3.2.	Indicador 2: Adopción.....	48
3.3.3.	Indicador 3: Actividad.....	49
3.3.4.	Indicador 4: Servicios .....	49
3.3.5.	Indicador 5: Empaquetado .....	49
3.3.6.	Indicador 6: Gestión.....	50
3.3.7.	Indicador 7: Técnicas de adaptabilidad.....	50
3.3.8.	Indicador 8: Licencia y derechos de autor .....	51
3.4.	Notación y reglas de puntuación .....	51
3.5.	Evaluación de los indicadores .....	56
3.5.1.	Madurez.....	57
3.5.2.	Adopción .....	58
3.5.3.	Actividad.....	59
3.5.4.	Servicios.....	61
3.5.5.	Empaquetado.....	62
3.5.6.	Gestión .....	63
3.5.7.	Técnicas de adaptabilidad .....	64
3.5.8.	Licencia y derechos de autor.....	66
3.6.	Resultados de la evaluación de los frameworks .....	67
Capítulo IV.....		71
Desarrollo del sistema web CALIAGUA .....		71
4.1.	Definición de la metodología .....	71
4.2.	Formulación del problema.....	71
4.3.	Planificación del proyecto .....	72
4.4.	Requisitos del sistema .....	72
4.4.1.	Introducción .....	73



4.4.2.	Descripción general.....	78
4.4.3.	Requisitos específicos .....	80
4.5.	Ingeniería.....	87
4.5.1.	Diseño arquitectónico.....	87
4.5.2.	Diseño de navegación .....	95
4.5.3.	Diseño de interfaz .....	99
4.6.	Generación de páginas.....	105
4.6.1.	Página de inicio .....	106
4.6.2.	Pantalla de administración de datos para usuario del sistema.....	106
4.6.3.	Pantallas de administrar barrios o comunidades .....	108
4.6.4.	Pantallas de administrar sistemas de agua potable.....	109
4.6.5.	Pantallas de administrar datos de las muestras de agua .....	112
4.6.6.	Pantallas de administración de usuarios.....	115
4.6.7.	Pantallas de consulta de fichas por cantón y parroquia.....	117
4.6.8.	Pantalla de generación de estadísticas.....	118
4.7.	Pruebas .....	121
4.8.	Implementación del sistema web CALIAGUA.....	122
4.8.1.	Archivo de conexión con la base de datos .....	122
4.8.2.	Conexión del módulo de estadísticas con la base de datos .....	123
4.9.	Dominio de la aplicación.....	126
Capítulo V .....		127
Metodología .....		127
5.1.	Tipo de estudio .....	127
5.2.	Población y muestra .....	127
5.2.1.	Población.....	127
5.2.2.	Tamaño de la muestra .....	128

5.3.	Operacionalización de variables.....	128
5.4.	Procedimientos .....	128
5.4.1.	Técnicas de investigación .....	128
5.5.	Hipótesis.....	129
5.5.1.	Hipótesis de investigación.....	129
5.5.2.	Determinación de las variables .....	129
5.6.	Procesamiento y análisis .....	129
5.6.1.	Comprobación de la hipótesis .....	129
5.6.2.	Decisión.....	138
Capítulo VI.....		139
Resultados .....		139
Capítulo VII .....		141
Discusión.....		141
Capítulo VIII.....		142
Conclusiones y recomendaciones .....		142
8.1.	Conclusiones .....	142
8.2.	Recomendaciones .....	143
IX. Bibliografía .....		144
X. APÉNDICES O ANEXOS.....		146
10.1.	Glosario de términos y abreviaturas .....	146
10.2.	Diagrama de gantt.....	147
10.3.	Código fuente de las aplicaciones prototipo .....	148
10.4.	Solicitudes de entrega de manuales del sistema .....	177
10.5.	Fotografías de las capacitaciones realizadas.....	180
10.6.	Acta de entrega del sistema en la Secretaria del Agua .....	184

## INDICE DE TABLAS

Tabla 1. Tecnologías del lado de cliente .....	13
Tabla 2. Arquitecturas de un framework.....	24
Tabla 3. Diferencias entre PHP y JSP (JAVA).....	31
Tabla 4. Características comunes en frameworks PHP.....	32
Tabla 5. Parámetros del método COCOMO .....	36
Tabla 6. Indicadores y parámetros para selección del framework PHP.....	47
Tabla 7. Puntuación de los parámetros .....	51
Tabla 8. Reglas de notación para el Indicador 1 .....	52
Tabla 9. Reglas de notación para el Indicador 2 .....	52
Tabla 10. Reglas de notación para el Indicador 3 .....	53
Tabla 11. Reglas de notación para el Indicador 4 .....	53
Tabla 12. Reglas de notación para el Indicador 5 .....	54
Tabla 13. Reglas de notación para el Indicador 6 .....	55
Tabla 14. Reglas de notación para el Indicador 7 .....	55
Tabla 15. Reglas de notación para el Indicador 8 .....	56
Tabla 16. Evaluación del indicador madurez .....	57
Tabla 17. Promedios para el indicador madurez.....	57
Tabla 18. Indicador adopción.....	58
Tabla 19. Promedio del indicador adopción .....	59
Tabla 20. Indicador actividad.....	60
Tabla 21. Promedio del indicador actividad .....	60
Tabla 22. Indicador servicios .....	61
Tabla 23. Promedio del indicador servicios.....	61
Tabla 24. Indicador empaquetado .....	62
Tabla 25. Promedio del indicador empaquetado.....	63

Tabla 26. Indicador gestión.....	63
Tabla 27. Promedio del indicador gestión .....	64
Tabla 28. Indicador técnicas de adaptabilidad .....	65
Tabla 29. Promedio del indicador técnicas de adaptabilidad.....	65
Tabla 30. Indicador licencia y derechos de autor.....	66
Tabla 31. Promedio del indicador licencia y derechos de autor .....	66
Tabla 32. Resumen de resultados.....	68
Tabla 33. Porcentajes de cada framework.....	69
Tabla 34. Resumen de actividades .....	72
Tabla 35. Operacionalización de variables .....	128
Tabla 36. Líneas de código fuente de la aplicación WebApp 1.....	130
Tabla 37. Líneas de código fuente de la aplicación WebApp 2.....	130
Tabla 38. Líneas totales de código fuente de las aplicaciones de prueba .....	131
Tabla 39. KLDC para las aplicaciones de prueba .....	131
Tabla 40. Valores coeficientes y exponentes .....	132
Tabla 41. Sueldo promedio de un desarrollor de sistemas.....	133
Tabla 42. Resumen de resultados de la estimación del software .....	134
Tabla 43. Intervalo de datos de la WebApp 1 .....	135
Tabla 44. Intervalo de datos de la WebApp 2.....	135
Tabla 45. Líneas de código con y sin framework .....	136
Tabla 46. Resultados de la tabla T-Student.....	138
Tabla 47. Porcentaje de frameworks PHP.....	140

## ÍNDICE DE GRÁFICOS E ILUSTRACIONES

Figura 1. Arquitectura cliente/servidor .....	7
Figura 2. Arquitectura de tres niveles .....	8
Figura 3. Esquema básico de una aplicación web .....	9
Figura 4. Arquitectura de tres niveles .....	12
Figura 5. Arquitectura de las aplicaciones web: todo separado .....	13
Figura 6. Arquitectura de PHP .....	15
Figura 7. Tendencias de búsqueda de PHP a nivel mundial .....	18
Figura 8. Tendencias de búsqueda de PHP en Ecuador .....	19
Figura 9. Ranking de los lenguajes más populares .....	20
Figura 10. Arquitectura genérica de un framework .....	25
Figura 11. Funcionamiento del modelo vista controlador .....	26
Figura 12. Componentes del framework .....	27
Figura 13. Uso de los frameworks PHP .....	29
Figura 14. Uso de los framework ASP.NET .....	30
Figura 15. Uso de los framework java .....	30
Figura 16. Número de votos obtenidos de PHP Frameworks .....	38
Figura 17. Estructura de directorios de CakePHP .....	41
Figura 18. Estructura de directorios de CodeIgniter .....	43
Figura 19. Estructura de directorios de Yii .....	46
Figura 20. Indicador madurez .....	57
Figura 21. Indicador adopción .....	59
Figura 22. Indicador actividad .....	60
Figura 23. Indicador servicios .....	61
Figura 24. Indicador empaquetado .....	62
Figura 25. Indicador gestión .....	64

Figura 26. Indicador técnicas de adaptabilidad.....	65
Figura 27. Indicador licencia y derechos de autor .....	66
Figura 28. Indicadores evaluados, y sus promedios.....	67
Figura 29. Porcentaje de todos los indicadores .....	69
Figura 30. Modelo de proceso IWEB .....	71
Figura 31. Secuencia para realizar consulta de fichas.....	82
Figura 32. Diseño de la Base de Datos .....	88
Figura 33. Diseño de Base de Datos de la tabla "muestras" .....	89
Figura 34. Diagrama general de casos de uso .....	90
Figura 35. Caso de uso "Iniciar sesión" .....	90
Figura 36. Caso de uso - Gestionar barrios y/o comunidades.....	91
Figura 37. Casos de usos - Gestionar de sistemas de agua .....	91
Figura 38. Casos de uso - Gestionar datos de muestras .....	91
Figura 39. Casos de uso - Gestionar datos de usuarios .....	92
Figura 40. Casos de uso - realizar consultas de fichas.....	92
Figura 41. Casos de uso - Porcentajes de tipos de sistemas.....	93
Figura 42. Casos de uso - Gráficas con los parámetros de agua potable .....	93
Figura 43. Casos de uso - Consultar calidad del agua.....	94
Figura 44. Diagrama de red de la UNACH.....	94
Figura 45. Diagrama de navegación - Página inicial .....	95
Figura 46. Diseño de navegación - Página de administración .....	96
Figura 47. Diseño de navegación - Página de registrado .....	96
Figura 48. Diseño de navegación - Módulo de fichas – Administrador .....	97
Figura 49. Diseño de navegación - Módulo de fichas – Registrado .....	97
Figura 50. Diseño de navegación - Módulo de consulta de fichas.....	98
Figura 51. Diseño de navegación - Módulo de estadísticas .....	98

Figura 52. Interfaz de usuario - Página principal .....	99
Figura 53. Interfaz de usuario administración de datos – Administrador .....	101
Figura 54. Interfaz de usuario administración de datos – Registrado .....	102
Figura 55. Interfaz de usuario - administracion de muestras de agua.....	103
Figura 56. Interfaz de usuario - Modulo de consulta de fichas.....	104
Figura 57. Interfaz de usuario - módulo de estadísticas.....	105
Figura 58. Pantalla principal del sistema web Caliagua .....	106
Figura 59. Pantalla de administración – usuario administrador.....	107
Figura 60. Pantalla de administración - usuario registrado.....	107
Figura 61. Pantalla de registrar barrios o comunidades .....	108
Figura 62. Pantalla de actualizar datos de un barrio o comunidad.....	108
Figura 63. Pantalla de registrar sistemas de agua potable (datos básicos).....	109
Figura 64. Pantalla de actualizar sistemas de agua potable (datos básicos).....	109
Figura 65. Pantalla de registrar datos técnicos de los sistemas de agua .....	110
Figura 66. Pantalla de actualizar datos técnicos de sistemas de agua potable ....	111
Figura 67. Pantalla de administración de fichas de muestras de agua potable ...	112
Figura 68. Pantalla para registrar muestras de agua potable.....	113
Figura 69. Pantalla para actualizar datos de muestras de agua potable .....	114
Figura 70. Pantalla de confirmación para eliminar muestra .....	115
Figura 71. Pantalla de registrar nuevo usuario.....	115
Figura 72. Pantalla de registro de nuevo usuario .....	116
Figura 73. Pantalla de mensaje de confirmación al eliminar un usuario .....	116
Figura 74. Pantalla de consulta de fichas por cantón.....	117
Figura 75. Pantalla de consulta de fichas por sistema de agua potable.....	117
Figura 76. Pantalla principal de generación de estadísticas.....	118
Figura 77. Pantalla de generar estadísticas de tipos de sistemas.....	118

Figura 78. Pantalla para generar gráfico de parámetros de calidad de agua .....	119
Figura 79. Pantalla para consultar la calidad del agua según su cantón .....	119
Figura 80. Pantalla de gráfico de calidad del agua según su cantón .....	120
Figura 81. Pantalla de gráfico estadístico de parámetro de calidad del agua.....	120
Figura 82. Pantalla de gráfico estadístico de los tipos de sistemas de agua.....	121
Figura 83. Gráfica de T – Student.....	137
Figura 84. Diagrama de Gantt.....	147
Figura 85. Capacitacion en laboratorios de la UNACH.....	180
Figura 86. Capacitacion en la SENAGUA subzonal Riobamba .....	180
Figura 87. Segunda capacitacion en la SENAGUA subzonal Riobamba .....	181
Figura 88. Capacitacion en laboratorios de la UNACH - foto 2.....	181
Figura 89. Presentacion del sistema en la SENAGUA Latacunga.....	182
Figura 90. Presentacion del sistema en la SENAGUA Latacunga - foto 2.....	182
Figura 91. Firma de Acta de entrega del sistema, Quito .....	183
Figura 92. Firma de Acta de entrega del sistema, Quito - foto 2.....	183



## RESUMEN

El trabajo de investigación se enfocó en el desarrollo de una aplicación web para almacenamiento, control y registro de la información de calidad del agua para consumo doméstico llevado por el departamento de agua potable y saneamiento de la SENAGUA, los datos eran reportados en hojas impresas por los municipios de la provincia de Chimborazo. Para el desarrollo de esta aplicación web se utilizó el lenguaje de programación PHP y un Sistema Gestor de Base de Datos (SGBD) MySQL.

La investigación se enfoca en la utilización de un Framework PHP opensource que facilite la construcción de reportes estadísticos y ayude a los técnicos de los municipios a tomar correctivos en los sistemas de agua potable que no cumplan los parámetros establecidos en las normas de calidad del agua NTE INEN 1 108:2011 Cuarta revisión y TULAS Libro VI, Anexo I, Tablas 1 y 2.

La existencia de muchas herramientas framework dificulta la selección de una en especial para la construcción de reportes estadísticos gráficos, con la ayuda de una metodología de calificación y selección de herramientas de software libre QSOS se elige Yii framework por cumplir en un 92.5 por ciento los requisitos técnicos, funcionales y estratégicos esta herramienta ayuda en la programación del módulo de reportes estadísticos.

La hipótesis de investigación “Un Framework PHP opensource mejorará el desarrollo de una aplicación web para la gestión de reportes estadísticos” se acepta por cuanto las pruebas de aceptación del análisis de la distribución T-Student dieron como resultado que el promedio de líneas de código de la aplicación utilizando el framework es menor que el promedio de líneas de código de la aplicación sin usar el framework lo cual reduce el personal y el costo necesario para la programación de líneas de código y una mayor rentabilidad del presupuesto establecido del proyecto para la empresa de desarrollo.



UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
CENTRO DE IDIOMAS



Lic. Geovanny Armas

18 de marzo de 2015

**SUMMARY**

This research work is focused on developing a web application for storage, control and recording of information about water quality for domestic consumption led by the department of drinking water and sanitation "SENAGUA", some data were reported in printed sheets of paper by the municipalities in the province of Chimborazo. In order to develop this web application, the PHP programming language and (SGBD) MySQL Database Manager System was used.

The research is focused on using a PHP opensource framework that facilitates the construction of statistical reports and helps municipality technicians in taking corrections in drinking water systems that do not meet the parameters established in NTE INEN 1 108:2011 water quality standards, Fourth revision and TULAS Book VI, Appendix I, Tables 1 and 2.

The existence of many framework tools makes it difficult to select one, especially for the construction of graphic statistical reports, with the help of a methodology for qualification and selection of QSOS free software tools and the Yii framework is chosen to fulfill a 92.5% of the technical, functional and strategic requirements. This tool helps in programming the module for statistical reports.

The research hypothesis "A PHP opensource Framework will improve the development of a web application for managing statistical reports" it is accepted because the acceptance testing of the analysis of the T-student distribution resulted in the average code lines for the application using the framework is lower than the average code lines without using the framework, it reduces staff and the cost required for programming code lines and a higher profitability of the budget established from the project for the company in development.



MSc. ANA MALDONADO L.  
COORDINADORA

## INTRODUCCIÓN

Las empresas e instituciones públicas o privadas necesitan llevar un control y almacenamiento de la información concerniente a sus procesos, actualmente toda información debe recuperarse de manera inmediata y en cualquier lugar del mundo por lo cual es necesario estar en Internet, existen muchas tecnologías para la implementación de aplicaciones web, las tecnologías del lado del cliente podemos mencionar HTML, HTML5, JavaScript, etc. y tecnologías del lado del servidor como JSP, PHP, ASP, Ruby, etc.

PHP es un lenguaje netamente para la creación de aplicaciones web de fácil aprendizaje para cualquier desarrollador que tenga conocimientos en lenguajes como C o C++.

Existen herramientas que ayudan al desarrollo de proyectos informáticos llamadas framework que no son más que herramientas que ayudan al desarrollador a gestionar de manera eficiente y rápida las aplicaciones. PHP tiene muchos framework para desarrollo de aplicaciones web de la misma manera que otros lenguaje de programación lo tienen, las razones por las cuales se optó por los framework PHP opensource para el desarrollo del aplicativo se explicó en líneas anteriores.

La importancia que tiene esta investigación es que no todos los frameworks PHP opensource pueden realizar tareas específicas como la generación de reportes estadísticos. Se analizarán tres framework PHP que serán seleccionados al tomar como referencias de empresas de desarrollo que utilicen este tipo de herramientas para sus proyectos y sitios web especializados.

El objetivo de este estudio es demostrar que el uso de frameworks para PHP mejora el desarrollo de aplicaciones web pero tomando en cuenta que se mejora la fase de programación, e implica el uso del personal necesario para el desarrollo del software y aún más importante el costo que involucra para la empresa desarrolladora construir el producto final y la rentabilidad para esta.

Se utilizará PHP como lenguaje núcleo de una aplicación web para manejo de la información de calidad del agua para consumo humano dirigido por la Secretaria del Agua esta información es registrada en una base de datos MySQL por los técnicos laboratoristas de los municipios de la provincia de Chimborazo. Con el estudio de los tres frameworks se elegirá el mejor framework que se adapte al objetivo de generar reportes estadísticos con la información almacenada en la base de datos.

En el primer capítulo de este proyecto de investigación se toman como referencia el anteproyecto donde se muestran los alcances tanto de la investigación como del aplicativo web.

En el segundo capítulo se enfatiza todo el marco teórico con los conceptos necesarios para iniciar el proyecto de investigación.

En el tercer capítulo se resaltan los framework PHP opensource que serán objeto de análisis y con el uso de una metodología de calificación y selección de herramientas de software libre se escogerá la herramienta que más se adapte al objetivo del proyecto que es generar reportes estadísticos.

En el capítulo cuatro se realiza la ingeniería web con los pasos de la metodología “ciclo de vida” establecida por Rogger Pressman: análisis, ingeniería, generación de páginas, pruebas e implementación final.

# CAPÍTULO I

## MARCO REFERENCIAL

### 1.1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

El desarrollo de aplicaciones web se ha vuelto muy popular entre las empresas desarrolladoras de software con lenguajes de programación como PHP, JSP, ASP, RUBY entre otros que brindan muchas ventajas para a la hora de ofertar sus productos y servicios.

Al considerar al lenguaje de programación PHP del lado del servidor como un lenguaje de alto nivel los desarrolladores a menudo eligen este lenguaje como una alternativa para realizar sus aplicaciones, frecuentemente lo hacen con programación estructurada, orientada a objetos que incluye el uso de clases, funciones, procedimientos para desarrollar de la manera más eficiente sus aplicaciones. Todo el trabajo de generar código puede ser simplificado con la utilización de los Framework que ayudan al desarrollo ágil de aplicaciones web, como una ventaja que se considera la más importante es la separación de la aplicación en capas lo cual nos ayuda a tener un dominio completo y mejor entendimiento de lo que se desarrolla.

Al usuario final como beneficiario de la aplicación le interesa la vista de reportes estadísticos que genere el sistema ya que le ayuda en la toma de decisiones; los Framework PHP ayudan a la obtención de este tipo de requerimientos, es por eso que la utilización de este tipo de herramientas es muy aconsejable.

Existen muchos Framework para PHP como por ejemplo: Kumbia, Zend, Symfony, Kohana y muchos otros por esta razón el grupo desarrollador no tiene idea de cuál utilizar por cuanto no se realiza un análisis previo para identificar cual se adapta a las necesidades del cliente es por eso que la selección adecuada de un Framework en este caso para PHP está vinculada a la gestión estadística.

El departamento de agua potable y saneamiento de la Secretaría del Agua (SENAGUA), dirección provincial de Chimborazo, lleva el registro de los datos correspondientes a la calidad del agua para consumo de la provincia, los cuales son registrados en fichas impresas, esto dificulta el acceso y seguimiento de los datos. A la SENAGUA le interesa obtener reportes estadísticos de la calidad del agua que consumen los ciudadanos de la provincia bajo ciertos indicadores de calidad como: pH, aluminio, cloruros, dureza total, calcio, magnesio, fosfato, etc.

La Central Ecuatoriana de Servicios Agrícolas (CESA), es una organización orientada a colaborar con el desarrollo agrario y rural del país, CESA apoya la gestión de los sistemas de producción, las iniciativas de comercialización asociativa de productos campesinos, el manejo técnico y social del agua (riego y agua para consumo).

Agrícolas y Veterinarios Sin Fronteras es una organización sin fines de lucro (VSF-CICDA), es el patrocinador del proyecto, entidad que financia las investigaciones tendientes a cooperar para el desarrollo rural a favor de las agriculturas campesinas en regiones marginadas y contribuir a acciones de cabildeo en el Norte y Sur a favor de las agriculturas. VSF-CICDA implementa y participa en programas de desarrollo rural en 19 países, entre ellos América Latina y Ecuador en particular. Desde el año 1990 VSF-CICDA contribuyo a la construcción del concepto de gestión social del agua a partir del desarrollo de proyectos para mejorar sistemas de riego tradicionales andinos.

La Facultad de Ingeniería desde el 15 de abril del 2010 mantiene un convenio de cooperación con CESA/AVSF, cuyo objeto es “ejecutar conjuntamente proyectos de investigación y desarrollo orientados al mejoramiento de vida de la población de Chimborazo, y la recuperación de los recursos naturales, de acuerdo a las condiciones que expresamente se establecerán entre las partes, mediante convenios específicos”.

## **1.2. FORMULACIÓN DEL PROBLEMA**

¿Un Framework PHP opensource ayuda en el desarrollo de una aplicación web para la gestión de reportes estadísticos en el sistema de agua potable de la provincia de Chimborazo?

## **1.3. OBJETIVOS**

### **1.3.1. OBJETIVO GENERAL**

Analizar los Frameworks PHP en el desarrollo de una aplicación web para la gestión de reportes estadísticos en el sistema de agua potable de la provincia de Chimborazo.

### **1.3.2. OBJETIVOS ESPECÍFICOS**

- ✓ Analizar los Framework PHP opensource utilizando indicadores que permitan establecer el más eficiente.
- ✓ Desarrollar la aplicación utilizando el Framework PHP más apto para la generación de reportes estadísticos.

## **1.4. JUSTIFICACIÓN**

El desarrollo de una aplicación ya sea de escritorio o web, el equipo involucrado en el proyecto utiliza una metodología y herramientas específicas para desarrollar cualquier sistema de información, estas aplicaciones ayudan al desarrollo y progreso de la sociedad.

Existen muchos Framework PHP opensource que se clasifican mediante estudios y comparaciones técnicas que son trascendentales para selección de la mejor herramienta que permita desarrollar una aplicación de calidad, si el grupo desarrollador no elige el Framework adecuado para el sistema en desarrollo el proyecto tendrían resultados que no se esperaban y además tiempos de respuesta lentos.

Luego de realizar un análisis de los Framework PHP opensource se pretende utilizar el que más se adapte a las necesidades de la organización para la toma de

decisiones más efectiva y eficaz en los sistemas de agua potable de la provincia de Chimborazo donde tengan falencias en alguno de sus indicadores.

### **1.5. ANTECEDENTES DEL TEMA**

Al revisar el repositorio digital de la Universidad Nacional de Chimborazo <http://dspace.unach.edu.ec> no se ha encontrado registro de ningún tipo de investigación referente al tema.

### **1.6. HIPÓTESIS**

Un Framework PHP opensource mejorará el desarrollo de una aplicación web para la gestión de reportes estadísticos.



## CAPÍTULO II

### FUNDAMENTACIÓN TEÓRICA

#### 2.1. ARQUITECTURA CLIENTE-SERVIDOR

Una arquitectura cliente/servidor es un tipo de arquitectura de red, donde cada ordenador puede tener las funciones tanto de servidor o cliente, puede recibir peticiones de otros usuarios de la red y también realizar peticiones a otros equipos.

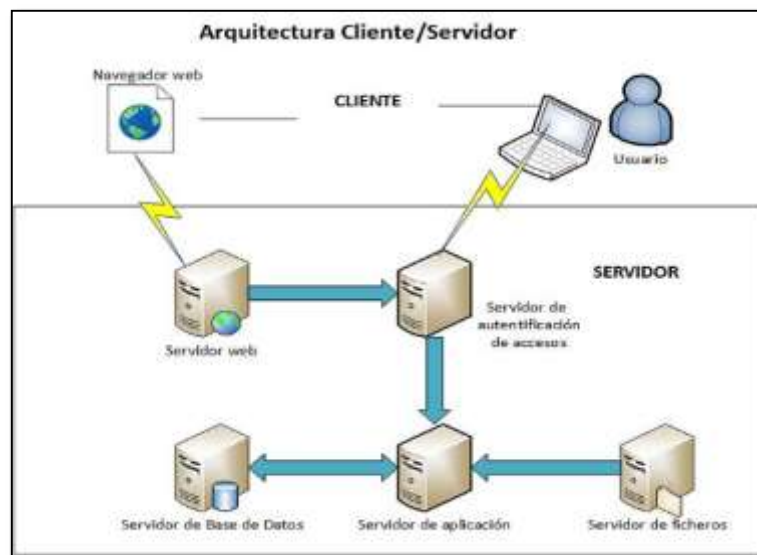


Figura 1. Arquitectura cliente/servidor  
Autor: Juan Carlos Moyota

Los *servidores* son equipos sofisticados para atender a muchas peticiones de usuarios, estos pueden ser servidores de archivos, impresoras, red, bases de datos o servidor de aplicaciones.

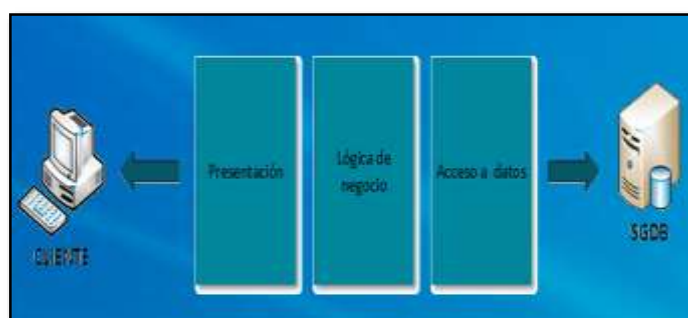
Los *clientes* son equipos con menos recursos de hardware que realizan peticiones de recursos que ofrecen los servidores. Los clientes pueden ser de dos tipos:

**Cientes inteligentes.** Tienen todo el hardware y software para funcionar de manera independiente.

**Cientes tontos.** Estos necesitan de los servidores para operar es decir no pueden funcionar de manera independiente.

### 2.1.1. SEPARACIÓN DE FUNCIONES

La arquitectura cliente/servidor nos permite la separación de funciones en tres niveles:



*Figura 2. Arquitectura de tres niveles  
Autor: Juan C. Moyota*

Las aplicaciones web se ejecutan dentro de un navegador web el cual realiza peticiones al servidor que tiene alojado el contenido, donde se pueden distinguir tres capas para el procesamiento y obtención de los resultados:

***Presentación.*** La lógica de presentación se encarga de la parte visual, es decir la presentación de la interfaz de usuario con la información que se genera con la lógica de negocios.

***Lógica de Negocios.*** La lógica de negocios es la encargada de proporcionar las peticiones que realiza el usuario, mantiene una comunicación con la capa de acceso a datos proporciona información y visualiza en la capa de presentación.

***Acceso a datos.*** La función principal de esta capa es proporcionar acceso a los datos a la capa lógica de negocios, la información que proporciona la extrae de un sistema gestor de base de datos (SGBD).

## 2.2. APLICACIONES INFORMÁTICAS

Una aplicación informática es un tipo de software que permite al usuario realizar uno o más tipos de trabajos, permite la interacción entre el usuario y el computador, existen muchos tipos de aplicaciones: procesadores de texto, hojas de cálculo, navegadores web, bases de datos, aplicaciones web, etc.

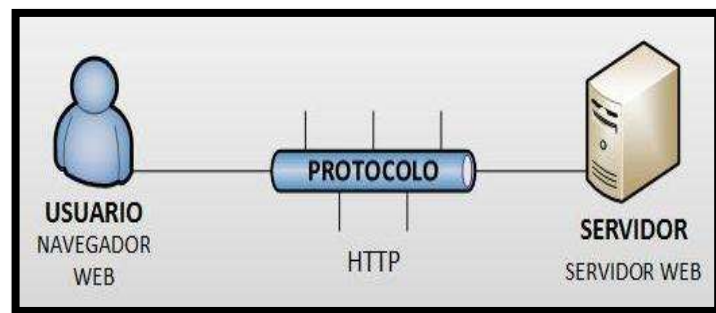
## 2.3. APLICACIONES WEB

Las aplicaciones web son aplicaciones a las que se accede mediante un navegador y están alojadas en servidores dentro de una Intranet o en Internet.

Las aplicaciones web de escritorio sustituyen a los programas que se instalan en los ordenadores y uqe realizan tareas como por ejemplo organizar agendas, calendarios, leer el correo electrónico, etc. (Niño, 2010)

El protocolo **HTTP**<sup>1</sup>, forma parte de la familia de protocolos de comunicación TCP/IP usado para las comunicaciones en Internet.

“Nos referimos a las aplicaciones web como WebApps (aplicaciones web) que son muy diferentes a otras categorías de software web.” (Pressman, 2010)



*Figura 3. Esquema básico de una aplicación web  
Autor: Juan C. Moyota*

Powell resume las diferencias básicas al afirmar que los sistemas basados en Web implican una mezcla de publicación impresa y desarrollo de software, de

---

<sup>1</sup> Protocolo de Transferencia de Híper Texto, protocolo utilizado en internet para la tranferencia de texto, gráficos y video en un solo conjunto o presentación.

marketing e informática, de comunicaciones internas y relaciones externas, y de arte y tecnología. (Powell, 1998, citado en Pressman, 2010)

### **2.3.1. ATRIBUTOS DE UNA APLICACIÓN WEB**

Pressman en el 2010 cita los siguientes como atributos que se pueden encontrar en la mayoría de las aplicaciones web:

***Intensivas de red.*** Las aplicaciones web residen en la red y brindan servicios a una comunidad diversa de usuarios. Las WebApps pueden funcionar en una intranet o una extranet.

***Controlada por el contenido.*** En muchos casos, la función primaria de una WebApp es utilizar hipermedia para presentar al usuario el contenido de textos, gráficos, sonido y vídeo.

***Evolución continúa.*** A diferencia de las aplicaciones convencionales que evolucionan en base a una serie de versiones planificadas, las aplicaciones web están en constante evolución, es usual que algunas WebApps actualicen su contenido cada hora. Algunos argumentan la evolución de las WebApps un trabajo similar a la jardinería. Un cuidado y una alimentación continua permiten que un sitio web crezca. Pero la diferencia es que las aplicaciones web deben satisfacer las necesidades de más de un usuario.

### **2.3.2. CARACTERÍSTICAS DE UNA APLICACIÓN WEB**

Según Pressman, 2010 establece las siguientes características comunes entre todas las aplicaciones web:

***Inmediatez.*** Las aplicaciones basadas en web tienen una inmediatez que no se encuentra en otros tipos de software. Es decir, el tiempo que se tarda en comercializar un sitio web completo puede ser cuestión de días o semanas.

***Seguridad.*** Las aplicaciones web están disponibles a través de la red y por esa razón es difícil controlar la población de usuarios finales que pueden acceder a la aplicación. Con el propósito de proteger la confidencialidad de la información y

los datos, deberán implementarse medidas de seguridad tanto para la infraestructura física que aloja la aplicación web así como la aplicación.

*Estética.* Una parte innegable del atractivo de una WebApp es su apariencia e interacción. Cuando se ha diseñado una aplicación con el fin de comercializarse o vender productos o ideas, la estética puede tener mucho que ver con el éxito del diseño técnico.

### 2.3.3. ARQUITECTURA DE LAS APLICACIONES WEB

Las aplicaciones web utilizan lo que se conoce como clientes livianos (light clients) los cuales no ejecutan demasiadas labores de procesamiento para la ejecución de la aplicación misma. Desde el punto de vista de la arquitectura se distinguen dos lados; uno es el cliente, donde se encuentra el usuario final utilizando la aplicación por medio de un navegador (como Internet Explorer o Mozilla Firefox). A través de este cliente web, el usuario interactúa con la aplicación localizada al otro lado, en el servidor, que es donde residen realmente los datos, reglas y lógica de la aplicación. (Ferrer, 2012)

Ferrer, en el 2012 menciona dos arquitecturas para aplicaciones web:

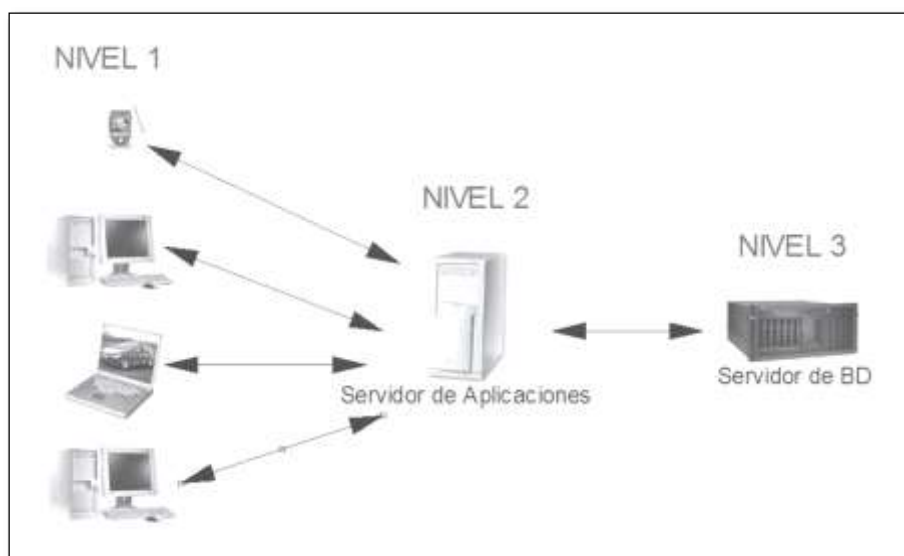
- Arquitectura cliente-servidor
  - Arquitectura de tres niveles
- a) **Cliente – servidor**

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados **servidores**, y los demandantes, llamados **clientes**. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

## b) Arquitectura de tres niveles

En la arquitectura en tres niveles existe un nivel intermedio. Esto significa que la arquitectura generalmente está compartida por:

- Un **cliente**, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
- El **servidor de aplicaciones** (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El **servidor de datos**, que proporciona al servidor de aplicaciones los datos que éste le solicitó.



*Figura 4. Arquitectura de tres niveles*

*Fuente: Ferrer, J (2012). Implementación de aplicaciones web. [Figura]. Madrid. RA-MA Editorial*

### 2.3.4. TECNOLOGÍAS PARA LA CREACIÓN DE APLICACIONES WEB

Existen muchas tecnologías que ayudan la creación de aplicaciones web, de las cuales podemos hablar de lenguajes del lado del cliente como se muestran en la tabla:

Tecnología	Descripción	Ejemplo
CSS, HTML estático	Formeto de las páginas: color, tamaño, capas, efectos, etc.	Estilos de texto
JavaScript , VBScript	Ejecutar eventos en el sistema web, ayuda a dar dinamismo a los sistemas web	Imágenes que cambian según algún evento.
Applets, de Java	Aplicaciones pequeñas para aplicaciones JSP	Puzzles, conectividad con base de datos.
Animaciones Flash	Animaciones como banners en movimiento, botones animados, etc.	Películas interactivas

Tabla 1. Tecnologías del lado de cliente  
Elaborado por: Juan C. Moyota

Las tecnologías del lado del servidor más usadas para la creación de páginas web dinámicas son: ASP, PHP y JSP. En la **Figura 5** se puede observar la arquitectura de cada una de ellas:

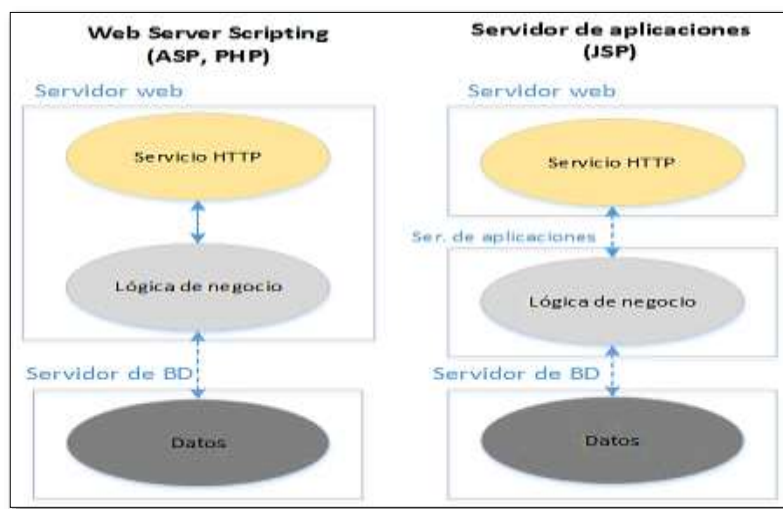


Figura 5. Arquitectura de las aplicaciones web: todo separado  
Elaborado por: Juan C. Moyota

Las tecnologías ASP(Active Server Pages) y PHP(PHP Hypertext Pre-processor) se clasifican dentro de la categoría Web Server Scripting. Estas tecnologías pueden emplear las arquitecturas 1 y 2. En estas tecnologías, el servidor web ofrece el servicio de HTTP y además se encarga de ejecutar los procesos que

poseen la lógica de negocio. La comunicación entre la lógica de negocio y los datos se suele realizar a través de métodos estandarizados como ODBC (Open DataBase Connectivity), lo que introduce una penalización en el tiempo de ejecución. En este tipo de tecnologías, parte de la lógica de negocio se puede desplazar hacia el servidor de bases de datos al emplear procedimientos almacenados y disparadores (triggers) de esta forma se logra una cierta escalabilidad en el sistema.

### **2.3.5. TIPOS DE APLICACIONES WEB**

Dart (citado en Pressman, 2010) sostiene que las categorías de aplicaciones que se enumeran a continuación son las más frecuentes en el trabajo web:

***Informativa:*** se proporciona un contenido sólo de lectura con navegación y enlaces simples;

***Descarga:*** un usuario descarga la información desde el servidor apropiado;

***Personalizable:*** el usuario personaliza el contenido a sus necesidades específicas;

***Interacción:*** la comunicación entre una comunidad de usuarios ocurre mediante un espacio chat (charla), tableros de anuncios o mensajería instantánea;

***Entrada del usuario:*** la entrada basada en formularios es el mecanismo primario de la necesidad de comunicación;

***Orientada a transacciones:*** el usuario hace una solicitud (por ejemplo, la realización un pedido) que es cumplimentado por la WebApp;

***Orientado a servicios:*** la aplicación proporciona un servicio al usuario, por ejemplo, ayuda al usuario a determinar un pago de hipoteca;

***Portal:*** la aplicación canaliza al usuario llevándolo a otros contenidos o servicios Web fuera del dominio de la aplicación del portal;

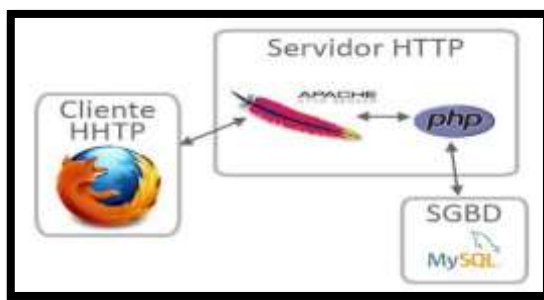


**Acceso a bases de datos:** el usuario consulta en una base de datos grande y extrae información;

**Almacenes de datos:** el usuario hace una consulta en una colección de bases de datos grande y extrae información.

## 2.4. PHP (PHP HYPERTEXT PREPROCESSOR)

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la web, dentro de una página web puede incrustar código PHP que se ejecutará cada vez que se visite una página. El código PHP es interpretado en el servidor web y genera código HTML y otro contenido que será visible por el visitante. (Welling & Thomson , 2005)



*Figura 6. Arquitectura de PHP  
Elaborado por: Juan Carlos Moyota*

PHP puede considerarse como un lenguaje de programación de alto nivel, código abierto, uso libre, además dispone de una gran cantidad de módulos y librerías libre de usar que hacen el trabajo fácil a los desarrolladores, otro aspecto importante de este intérprete es la conexión con bases de datos tanto libres como comerciales. Al ser un lenguaje de programación completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a la información almacenada en una base de datos.

### 2.4.1. BREVE HISTORIA DE PHP

PHP fué creado a principios de 1995 por Rasmus Lerdorf como un conjunto de utilidades escritas en lenguaje Perl (llamadas PHP/FI y posteriormente PHP 2.0) que permitían añadir dinamismo a las páginas web. Zeev Suraski y Andi

Gutmans rediseñaron este conjunto de herramientas con esto crearon el primer interprete que bautizaron con el nombre de PHP 3.0 en el año 1998.

Transcurría el año 2000, cuando se rediseñó por completo el intérprete, se añadió más potencia y funcionalidades con lo cual nació PHP4. A mediados del año 2004, se liberó la primera versión de lo que hoy conocemos como PHP5 la cual incorporaba nuevas funcionalidades como: POO, extensiones, conexión con bases de datos, etc.

#### **2.4.2. CARACTERÍSTICAS DE PHP**

Entre las características de PHP, un lenguaje para la creación de sistemas web, podemos destacar las siguientes:

- ✓ Potente y robusto lenguaje de programación embebido en documentos HTML.
- ✓ Conexión con la mayoría de sistemas gestores de bases de datos como: Oracle, PostgreSQL, Informix, MySQL, SQL Server entre otros con la ayuda de extensiones que permiten una conexión con estos SGBD.
- ✓ Soporte a múltiples protocolos de comunicación en Internet (HTTP, IMAP, FTP, etc.)
- ✓ Gratuito y no necesita realizar ningún desembolso económico para realizar sistemas de información con PHP.
- ✓ Código fuente del intérprete es abierto por lo cual se pueden hacer mejoras o sugerencias en su desarrollo.
- ✓ Lenguaje multiplataforma, existen versiones del intérprete PHP para plataformas como Windows, UNIX, Linux, MacOS, etc. Es por eso que las aplicaciones desarrolladas pueden cambiar de plataforma sin problemas de compatibilidad.
- ✓ Eficiente ya que consume muy pocos recursos del servidor, por esa razón que en equipos sencillos es posible desarrollar aplicaciones complejas.
- ✓ Alta velocidad de desarrollo: PHP permite realizar rápidamente sitios web dinámicos. Proporciona gran cantidad de extensiones con muy buena documentación lo cual ahorra mucho tiempo al desarrollador.

### 2.4.3. VENTAJAS Y DESVENTAJAS

Welling & Thomson (2005) menciona las siguientes ventajas de usar PHP para la producción de proyectos web:

- ✓ Alto rendimiento
- ✓ Interfaces para diferentes sistemas de bases de datos.
- ✓ Bibliotecas incorporadas para muchas tareas web habituales.
- ✓ Bajo coste
- ✓ Facilidad de aprendizaje u uso.
- ✓ Portabilidad
- ✓ Disponibilidad de código abierto.
- ✓ Disponibilidad de asistencia técnica.

Entre las desventajas podemos mencionar las siguientes:

- ✓ Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número pero hoy en día existen servidores con mayores recursos que soportan muchas solicitudes a la vez sin perder eficiencia en la entrega de resultados.
- ✓ La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

### 2.4.4. RAZONES PARA USAR PHP EN PROYECTOS

Welling & Thomson (2005) mencionan las siguientes razones por las cuales usar PHP en la producción de sistemas de información como son:

**Multiplataforma:** PHP puede ser instalado y configurado en plataformas de sistemas operativos como: Windows, Linux, MacOS, cuando se decida cambiar de plataforma, las aplicaciones no sufren ningún daño y pueden seguir en funcionamiento sin mayores problemas.

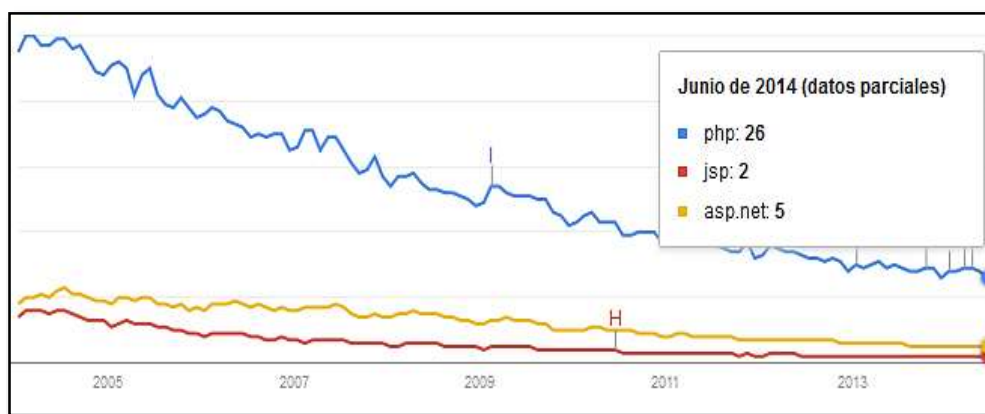
**Gratis:** el intérprete PHP es gratuito no requiere de ningún tipo de pago económico para ser usado en la creación de aplicaciones.

**Conexión con bases de datos:** PHP dispone de extensiones para manejar bases de datos como PostgreSQL, MySQL, SQL Server, Oracle, etc. Con esto de hacer uso de servidores de bases de datos ya instalados en otros equipos o en el mismo servidor que aloja el intérprete PHP.

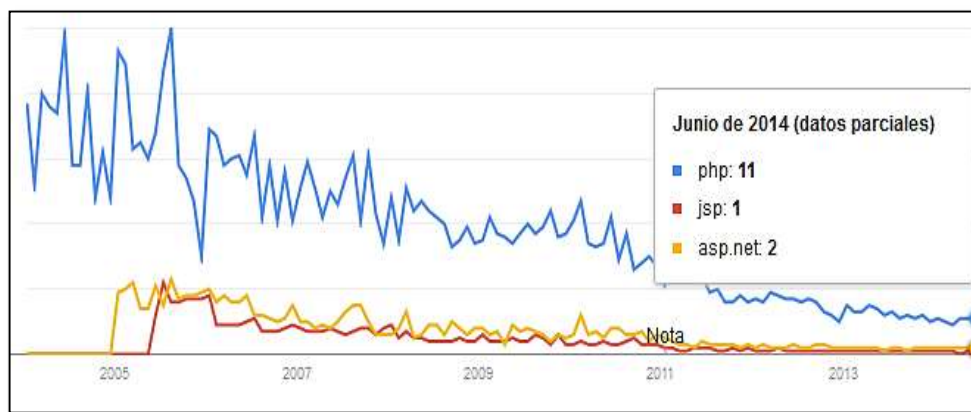
**Facilidad de aprendizaje:** la sintaxis de PHP se basa en otros lenguajes de programación, principalmente en C y Perl. Si ya conoce C o Perl, o un lenguaje de tipo C como C++ o Java, no tardará nada en utilizar PHP de manera productiva.

#### 2.4.5. POPULARIDAD DE PHP

Con el uso de la herramienta Google Trends se obtuvo la tendencia de búsqueda a nivel mundial y local (Ecuador) de las tecnologías del lado del servidor para la creación de aplicaciones web de gran tráfico como: PHP, JSP y ASP .NET indicadas en las **Figuras 7 y 8**.



*Figura 7. Tendencias de búsqueda de PHP a nivel mundial  
Elaborado por: Juan C. Moyota*



*Figura 8. Tendencias de búsqueda de PHP en Ecuador  
Elaborado por: Juan C. Moyota*

Los resultados que muestra la herramienta Google Trends son resultados relativos o porcentajes que representan cada búsqueda de un término y siempre con puntuaciones de 1 a 100 por ciento.

Los resultados que se tomaron como referencia son datos de junio de 2014 como se describe: 26% de búsquedas para el término PHP, un 2% para las búsquedas de JSP y un 5% para las búsquedas de ASP.NET. En la **Figura 8** vemos datos sólo de Ecuador donde acotamos que existe un 11% de búsquedas del término PHP, un 1% para JSP y un 2% de búsquedas para el término ASP.NET.

Concluimos de esta manera que PHP es considerado un lenguaje muy popular en el ámbito web como muestran los resultados de Google Trends tanto a nivel mundial como en nuestro país.

TIOBE<sup>2</sup> en sus artículos muestra la popularidad de los lenguajes de programación, se actualiza una vez al mes. Las calificaciones se basan en el número de ingenieros cualificados en todo el mundo, cursos y proveedores de terceros.

En el sitio oficial de TIOBE encontramos un artículo en inglés titulado “**TIOBE Index for January 2014**” que en español se traduce **Índice TIOBE para enero**

---

<sup>2</sup> TIOBE es una inversión de la empresa suiza SYNSPACE especializada en la evaluación y el seguimiento de la calidad del software iniciada el 1 de octubre de 2000. Dirección web: <http://www.tiobe.com/>

2014 donde se ilustra una tabla con los lenguajes de programación más populares como muestra la **Figura 9**, podemos interpretar que entre los lenguajes que no presentan ni altas ni bajas de popularidad esta PHP ubicado en el puesto número 6 a nivel mundial con lo cual se demuestra ser un lenguaje natural para la creación de sistemas de información web.

Enero 2014	Enero 2013	Cambio	Lenguaje de Programación	Valoraciones	Cambio
1	1		C	17.871%	+0.02%
2	2		Java	16.499%	-0.92%
3	3		Objective-C	11.098%	+0.82%
4	4		C++	7.548%	-1.59%
5	5		C#	5.855%	-0.34%
6	6		PHP	4.627%	-0.92%
7	7		(Visual) Basic	2.989%	-1.76%
8	8		Python	2.400%	-1.77%
9	10	▲	JavaScript	1.569%	-0.41%
10	22	▲▲	Transact-SQL	1.559%	+0.98%

*Figura 9. Ranking de los lenguajes más populares*  
Fuente: <http://www.comoprogramar.org/ranking-lenguajes-de-programacion/>

## 2.5. FRAMEWORK

“En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado” (Gutierrez, Tejeda, Silveira, & Caballero, 2013)

El término framework (marco de trabajo) se emplea en muchos ámbitos de desarrollo de aplicaciones no solo del tipo web, es así que pueden existir Frameworks para un sin número de ámbitos como por ejemplo: aplicaciones médicas, desarrollo de juegos, virtualización de escenarios, interfaces de usuario, etc.

### **2.5.1. CARACTERÍSTICAS DE UN FRAMEWORK**

Larman (citado en Polo 2014) menciona las siguientes características comunes entre un framework:

- Disponen de un conjunto cohesivo de interfaces y clases que colaboran para proporcionar los servicios de la parte central e invariable de un subsistema lógico.
- Contienen clases concretas y abstractas que definen las interfaces a las que ajustarse, interacciones de objetos en las que participar, y otras invariantes.
- Normalmente, requieren que el usuario del framework defina subclases de las clases que se incluyen en él para utilizar, adaptar y extender los servicios del framework.

### **2.5.2. CLASIFICACIÓN DE LOS FRAMEWORK**

Polo, 2014 clasifica a los Frameworks en tres grupos:

- Frameworks para desarrollo de interfaces
- Frameworks de persistencia para mapeo objeto relacional
- Frameworks para desarrollo de aplicaciones web

#### ***2.5.2.1. FRAMEWORKS PARA DESARROLLO DE INTERFACES***

Son herramientas que ayudan a diseñar interfaces de usuario con la ayuda de componentes visuales, mientras el usuario agrega elementos como cajas de texto, botones, etc. en su entorno de desarrollo se genera código correspondiente al aspecto de la ventana que luego el programador puede personalizar.

Entre los ejemplos de Frameworks de este tipo están JSF y ASP.NET.

#### ***2.5.2.2. FRAMEWORKS DE PERSISTENCIA PARA MAPEO OBJETO RELACIONAL***

En el mundo del desarrollo de las aplicaciones existen gestión de persistencia de los objetos es así que los objetos y las tablas son muy diferentes.

Es conveniente mantener diseños equivalentes entre la estructura de clases y la estructura de tablas, que puede lograrse mediante la aplicación de patrones de transformación, como el patrón “Una clase, una tabla” y en segundo lugar se deben asignar las responsabilidades de persistencia de forma que cumplan los principios básicos de diseño de software. (Polo, 2014)

Existen frameworks que ayudan a resolver la gestión de persistencia los cuales se conocen como frameworks ORM: object-relational mapping los cuales proporcionan:

- Una API para gestionar las operaciones CRUD.
- Un lenguaje para especificar las consultas que se refieren a las clases y a sus propiedades.
- Una forma de definir y especificar los metadatos del mapeo.
- Una técnica para la implementación del mapeo que permita la interacción con objetos transaccionales y soporta diversas funciones.

### **2.5.2.3. FRAMEWORKS PARA DESARROLLO DE APLICACIONES WEB**

Es una estructura definida, reusable en el que sus componentes facilitan la creación de aplicaciones web, en cierto sentido podemos afirmar que nos proveen una capa de abstracción sobre la arquitectura original ocultándola o adaptándola para no tener que utilizar el protocolo http de manera nativa y así acelerar los tiempos de desarrollo y mantenimiento. (Digiovannini, 2007)

### **2.5.3. IMPORTANCIA DE UN FRAMEWORK**

Zabala & Ochoa, 2008 mencionan la importancia de los framework e indican lo siguiente:

**Modularidad y reducción de complejidad.** La aplicación está formada por subsistemas especializados en distintos aspectos de toda la aplicación.

**Documentación.** La documentación del framework promueve el uso correcto del mismo y disminuye el esfuerzo para las tareas de mantenimiento.



***Distribución de funciones.*** Permite distribuir el trabajo de desarrollo, ya que la solución puede desarrollarse con un conjunto de piezas independientes que encajan en el framework usado.

***Eficiencia.*** El desarrollo puede concentrarse en los requerimientos funcionales de la aplicación.

***El código se ejecuta en varios tipos de plataformas.*** Los desarrolladores de los frameworks son unos genios, y además de algún modo cuentan con la capacidad de probar sus frameworks en varios tipos de plataforma y asegurarse de que se ejecuten en cada una de ellas.

***Seguridad de la aplicación.*** La mayoría de los frameworks web traen incorporados mecanismos de seguridad, en la forma de procedimientos de sanitación y validación de datos que permite filtrar todo lo que es enviado dentro de la aplicación y que generalmente sirve como vector de ataque. Pero puede constituirse en una espada de doble filo para los desarrolladores web. Por una parte corta rápidamente el problema de la seguridad de las aplicaciones con la utilización de los mecanismos de seguridad del framework; pero por otra parte, hay el riesgo de confiarse de los mecanismos de los frameworks web, ni preocuparse por la seguridad lógica de esta, ni estar al tanto de las actualizaciones de seguridad de nuestro framework.

#### **2.5.4. ARQUITECTURA DE UN FRAMEWORK**

Los frameworks presentan muchas arquitecturas, pero los siguientes son patrones básicos para la construcción de aplicaciones web (Digiovannini, 2007), de las cuales podemos mencionar las siguientes:

<b>Arquitectura</b>	<b>Definición</b>
Modelo I	Se trata de una página JSP que carece de un controlador accede directamente al modelo, se aplica la lógica de negocio y seleccionar la vista.
Modelo II	Un Servlet actúa como controlador central que actúa como controlador central que recibe las peticiones y datos de entrada. Las vistas no tienen relación con el modelo.
Modelo Vista Controlador	Separación de la aplicación en tres capas: modelo, controlador y la vista de usuario con la existencia de una comunicación entre las tres capas.

*Tabla 2. Arquitecturas de un framework  
Elaborado por: Juan Carlos Moyota*

En el presente trabajo se enfocará en el patrón MVC (Modelo – Vista - Controlador) que facilita la implementación de cambios en cualquier parte del sistema sin afectar al resto de la aplicación. Este patrón de diseño es el más usado en los frameworks web por características que le permite al desarrollador usar e incorporar código de terceros por lo cual ayuda en el desarrollo rápido y eficiente de aplicaciones web. Por eso la mayoría de los frameworks en su documentación oficial se menciona que fueron desarrollados bajo el patrón de diseño MVC.

#### **2.5.4.1. MODELO VISTA CONTROLADOR (MVC)**

El modelo vista controlador (MVC) o en inglés Model View Controller, es el patrón de diseño de software más usado en las aplicaciones web ya que tiene la característica de separar la lógica de negocio de la interfaz de usuario.

Facilita la funcionalidad, mantenibilidad y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite “no mezclar lenguajes de programación en el mismo código”.



Figura 10. Arquitectura genérica de un framework  
Elaborado por: Juan Carlos Moyota

McArthur, 2008 menciona que el Modelo Vista Controlador (MVC) es un patrón de diseño que simplifica el desarrollo de aplicaciones y mantenimiento. Esto se logra mediante la separación de la aplicación en tres componentes lógicos:

- ✓ *Modelo*
- ✓ *Vista*
- ✓ *Controlador*

#### 2.5.4.1.1. Componentes

Los tres componentes del patrón MVC son:

**Modelo:** representa la lógica de negocios, es la encargada de acceder a los datos al operar como intermediario con la base de datos.

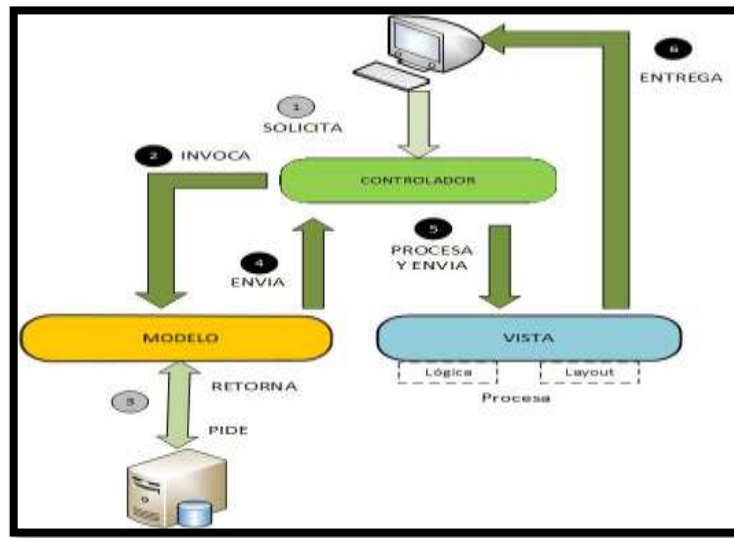
**Vista:** se encargar de mostrar la información de manera gráfica, visible y entendible para el usuario.

**Controlador:** actúa como medio entre la vista y el modelo, controla las peticiones del usuario solicita datos y los muestra en la vista de forma entendible y legible para cualquier usuario.

### 2.5.4.1.2. *Funcionamiento*

El funcionamiento del modelo vista controlador se presenta en la **Figura 11**, con el siguiente flujo:

- ✓ El usuario realiza la petición HTTP.
- ✓ El controlador captura el evento.
- ✓ Hace llamada al modelo(s) se realizan las modificaciones pertinentes sobre el modelo.
- ✓ El modelo se encarga de la comunicación e interacción con la base de datos de manera directa o con el uso de cualquier medio de comunicación con la base de datos y retorna la información al controlador.
- ✓ El controlador recibe la información y la envía a la vista.
- ✓ La vista procesa esta información, crea de abstracción para la lógica y la otra para la interfaz de usuario.



*Figura 11. Funcionamiento del modelo vista controlador*  
Autor: Juan C. Moyota

### 2.5.5. COMPONENTES DE UN FRAMEWORK

Para complementar la arquitectura escogida y enfrentar diferentes problemas de interoperabilidad, seguridad, facilidad de acceso y desempeño, se definieron los siguientes componentes:

**Controlador:** Se encarga de procesar las solicitudes que se realizan desde la interfaz gráfica y las re-direcciona a los procesos especializados que devuelven su respuesta mediante el mismo controlador.

**Gestor de Seguridad:** Encargado de validar los permisos del usuario en sesión para realizar la autenticación y autorización del mismo, este componente es accedido por el controlador en el momento de requerir una acción determinada así se logra en el momento que la sesión del usuario tenga cualquier cambio, que este se vea reflejado al usuario.

**Gestor de formularios:** Realiza todo el trabajo relacionado con la carga, validación y operaciones CRUD sobre los formularios, de esta manera se desacopla la operación central del Framework de lo relacionado con acceso a datos y gestión de seguridad.

**Gestor de acceso a datos:** Este componente se encarga de realizar el acceso a las Bases de datos que se hayan configurado, así se centraliza el acceso y se permite el uso de diferentes tipos de base de datos.

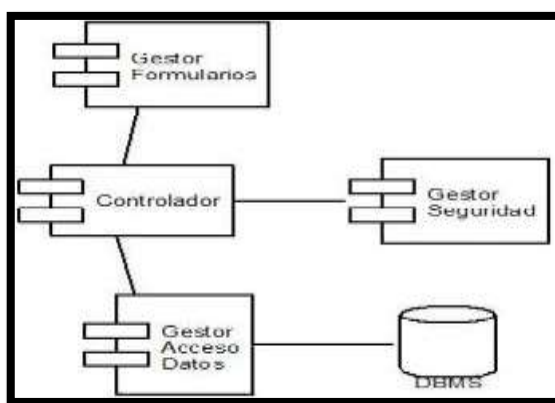


Figura 12. Componentes del framework

Fuente: <http://revistas.utp.edu.co/index.php/revistaciencia/article/viewFile/1817/11>

## 2.6. FRAMEWORKS PARA LA CREACIÓN DE APLICACIONES WEB

Los frameworks web son estas herramientas que podemos encontrar bajo muchos de los lenguajes de programación entre los que se destacan:

- PHP
- JSP(Java)
- ASP .NET
- Perl
- Phyton, etc.

Los lenguajes de programación antes mencionados son los más usados para la creación de aplicaciones web dinámicos, existen frameworks para cada uno de estos lenguajes de programación web.

Actualmente en el mercado encontramos frameworks para desarrollar aplicaciones web, que es el ámbito más importante donde se usan los frameworks. La mayoría de estos utilizan el patrón de diseño MVC del cual se habló previamente.

El objeto de escoger el framework más adecuado para el desarrollo y producción de nuestros proyectos web, conlleva a muchos factores que influyen en la implementación final, aquí podemos hablar de:

- El tipo de aplicación que se va a crear, si hablamos de una aplicación sencilla o una verdadera aplicación web, que tendrá actualizaciones, correcciones y mejoras a futuro.
- Con el fin de aprovechar las ventajas que ofrece el patrón de diseño MVC, el cual separa la aplicación en tres capas y tener un mayor dominio de la aplicación, donde cada capa es independiente y no exista mayor problema para poder realizar cambios o mejoras.
- Aprovechar las ventajas adicionales que nos ofrece el framework para la integración con otras herramientas.

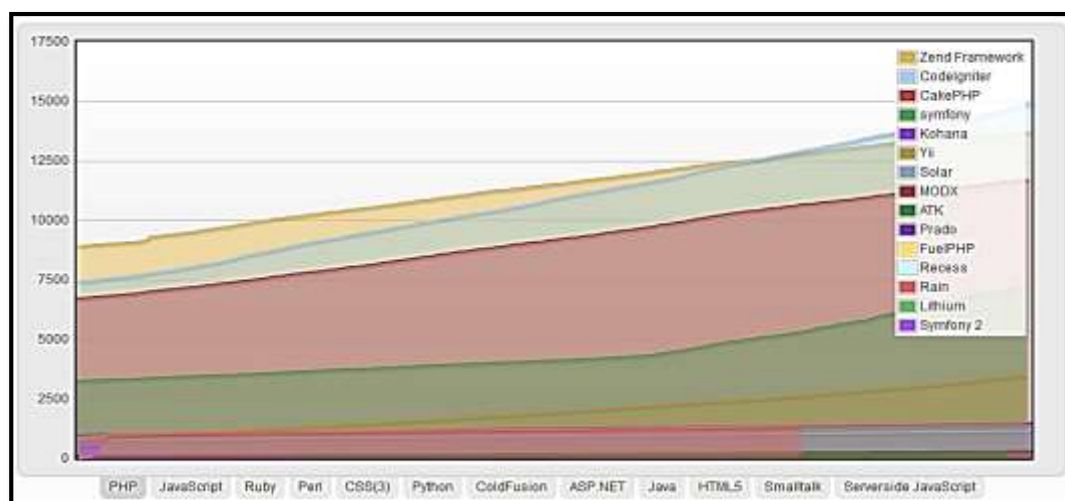
En el sitio web de **BESTWEBFRAMEWORKS**<sup>3</sup> se muestra el uso de los framework web bajo tres de los lenguajes más populares de la web: PHP, JAVA y

---

<sup>3</sup> **BESTWEBFRAMEWORKS**: Es un sitio web donde se puede comparar diferentes frameworks web para cada uno de los lenguajes de programación web o lenguaje de scripting. Dirección web <http://www.bestwebframeworks.com>

ASP .NET, las gráficas que a continuación se presentan son elaborados con antecedentes del 2011 tomándose como referencia para el análisis de los frameworks. Los resultados se basan en el API Stack Exchange que es una red de 89 sitios de preguntas y respuestas sobre diversos temas orientados al desarrollo de software, con lo cual podemos decir que los datos representan la cantidad de usuarios que dieron su preferencia al usar una herramienta framework específica.

La **Figura 13** hace referencia al uso de los frameworks PHP, se consideran a Zend Framework, CodeIgniter, CakePHP como frameworks destacados, el más utilizado tiene una utilización por encima de los 7500 usuarios y alcanza los 15000 usuarios, con esto podemos decir que estas herramientas framework PHP son usadas mucho para la creación de aplicaciones web dinámicas.



*Figura 13. Uso de los frameworks PHP*

*Tomado de la fuente: <http://www.bestwebframeworks.com/web-frameworks-usage/#php>*

En la **Figura 14**, podemos apreciar dos no muy conocidos frameworks para la plataforma ASP.NET: DotNetNuke y MonoRail, de los cuales el primero tiene un uso que sobrepasa los 600 usuarios y llega a superar los 1200 usuarios, con esto podemos decir que estas herramientas framework son también usadas para la implementación de aplicaciones web dinámicas bajo entornos Windows con lenguajes como Visual Basic o Visual C#.

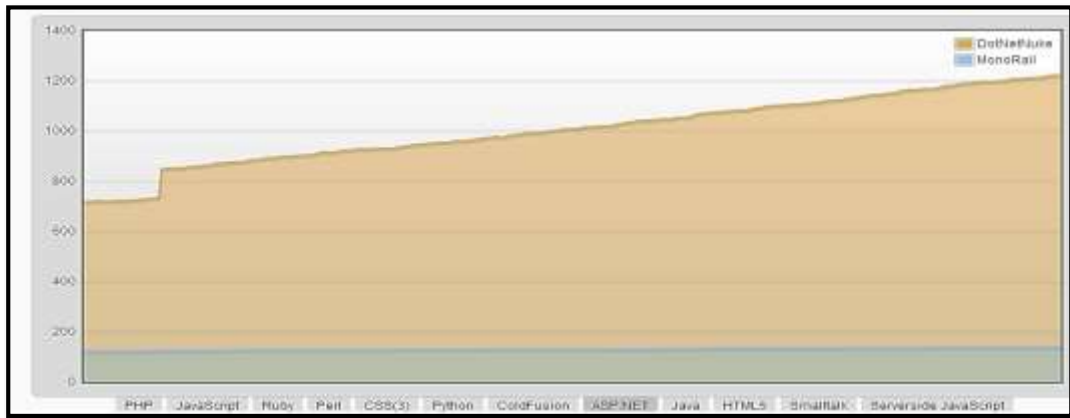


Figura 14. Uso de los framework ASP.NET

Tomado de la fuente: <http://www.bestwebframeworks.com/web-frameworks-usage/#asp-net>

Para terminar con el uso de los frameworks, la **Figura 15** observamos la cantidad de usuarios que usan los frameworks JSP (Java), un lenguaje web que a tomado mucha popularidad en los últimos años para la creación de aplicaciones web dinámicas y empresariales, en el gráfico se presentan dos de los frameworks JSP: Hibernate y Spring, los cuales tienen 10000 usuarios y superan los 20000 usuarios, concluyendo que los frameworks JSP se utilizan con frecuencia para la generación de sistemas web.

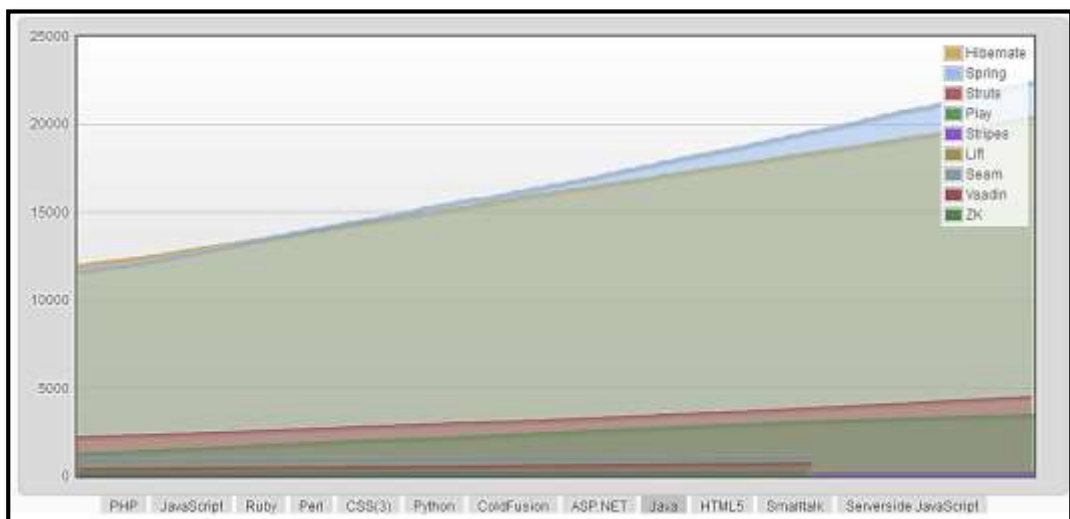


Figura 15. Uso de los framework java

Tomado de la fuente: <http://www.bestwebframeworks.com/web-frameworks-usage/#java>

Al observar las Figuras 17,18 y 19, llegamos a la conclusión que los Frameworks PHP y JSP son los más usados para desarrollo de aplicaciones en internet; los



sitios web que se encuentran a menudo podrían ser desarrollados con el uso de estos frameworks.

En la siguiente tabla se mencionan algunas de las características más frecuentes entre los frameworks web, lo que ayudará a seleccionar la plataforma más adecuada entre PHP y JSP (JAVA):

<b>Características</b>	<b>PHP</b>	<b>JSP</b>
Open source	Si	Si
Facilidad de aprendizaje	Si	No
Orientación a objetos	Si	Si
Multiplataforma	Si	Si
Conexión con base de datos	Si	Si
Documentación	Si	Si
Capacidad de incorporar extensiones	Si	Si

*Tabla 3. Diferencias entre PHP y JSP (JAVA)  
Elaborado por: Juan Carlos Moyota*

En la **Tabla 3** se puede observar una clara diferencia entre los lenguajes de programación PHP y JSP, la codificación JSP es más compleja que PHP, ya que para desarrollar aplicaciones web en JSP es necesario dominar el lenguaje Core Java lo cual nos llevaría un mayor tiempo en la entrega del sistema a desarrollarse, PHP es un lenguaje de fácil aprendizaje utiliza POO para el desarrollo de aplicaciones web dinámicas al igual que JSP. Por la facilidad de aprendizaje de PHP optamos por analizar los frameworks PHP.

## **2.7. FRAMEWORKS PARA PHP**

En internet encontramos muchos sitios web construidos con el lenguaje de programación PHP, algunos de estos sitios pueden ser desarrollados con el uso de frameworks, en la **Tabla 4** se resume las características que la mayoría de los frameworks PHP tienen:

<b>PHP framework</b>	<b>PHP4</b>	<b>PHP5</b>	<b>MVC</b>	<b>Multiple DB's</b>	<b>ORM</b>	<b>DB Objects</b>	<b>Templates</b>	<b>Caching</b>	<b>Validation</b>	<b>Ajax</b>	<b>Auth Module</b>	<b>Modules</b>	<b>EDP</b>
Yii	-	X	X	X	X	X	X	X	X	X	X	X	X
Zend	-	X	X	X	X	X	X	X	X	X	X	X	-
Symfony	-	X	X	X	X	X	-	X	X	X	X	X	-
Seagull	X	X	X	X	X	X	X	X	X	X	X	X	-
Prado	-	X	X	X	X	X	X	X	X	X	X	X	X
CodeIgniter	X	X	X	X	-	X	X	X	X	-	-	X	-
CakePHP	X	X	X	X	X	X	-	X	X	X	X	X	-
Zend	-	X	X	X	X	X	X	X	X	X	X	X	-
ZooP	X	X	X	X	-	X	X	X	X	X	X	-	-
QPHP	X	X	X	X	-	X	X	-	X		X	X	X

*Tabla 4. Características comunes en frameworks PHP  
Fuente: Tomado de <http://www.phpframeworks.com/>*

Los siguientes ítems describen las características que se presentan en los frameworks PHP presentes en la **Tabla 4**:

- ✓ **MVC**: Indica si el framework viene con soporte incorporado para una configuración de Modelo-Vista-Controlador.
- ✓ **Múltiple DB**: Indica si el framework es compatible con múltiples bases de datos sin tener que cambiar nada.
- ✓ **ORM**: Indica si el framework es compatible con un asignador de objeto de registro, por lo general una implementación de ActiveRecord.
- ✓ **Objetos DB**: Indica si el framework incluye otros objetos de base de datos, como un TableGateWay.
- ✓ **Plantillas**: Indica si el framework tiene un motor de plantillas incorporado.
- ✓ **Almacenamiento en caché**: Indica si el framework incluye un objeto de caché o de alguna manera otra forma de almacenamiento en caché.
- ✓ **Validación**: Indica si el framework tiene una validación incorporado o componente de filtrado.
- ✓ **Ajax**: Indica si el framework viene con soporte incorporado para el Ajax.
- ✓ **El módulo de autenticación**: Indica si el framework tiene un módulo incorporado para el manejo de la autenticación de usuarios.
- ✓ **Módulos**: Indica si el framework tiene otros módulos, como un analizador de RSS, módulo PDF o cualquier otra cosa (útil).
- ✓ **EDP** : Programación dirigida por eventos

## 2.8. ESTIMACIÓN DEL SOFTWARE

Las métricas del software orientadas al tamaño provienen de la normalización de las medidas de calidad y/o productividad considerando el “tamaño” del software que se haya producido. (Pressman, 2010)

Para la estimación del software podemos utilizar las líneas de código fuente (LDC) o los puntos de función (PF).

Las líneas de código usadas son instrucciones de programa que se escriben en un lenguaje de programación determinado para realizar una tarea específica del sistema. Cuanto menos líneas de código fuente se utilicen para ver los resultados esperados es mucho mejor de esta manera existe ahorro en recursos hardware como espacios de memoria ocupados por el número de variables presentes en el bloque de código.

Los defensores de la medida LDC afirman que la LDC es un “artificio” que se puede calcular fácilmente para todos los proyectos de desarrollo de software, que muchos modelos de estimación del software existente utilizan LDC o KLDC como clave de entrada.

En todas las fases de desarrollo del software se puede estimar el proyecto de software con la ayuda de modelos empíricos de cálculo de costos y esfuerzos.

### **2.8.1. MODELO COCOMO**

Barry Boehm (Pressman, 2010), en su libro sobre “Economía de la Ingeniería del Software”, menciona una escala de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Costo). La escala de modelos de Boehm incluye:

- Modelo 1. El modelo COCOMO básico calcula el esfuerzo (y el costo) del desarrollo de software en función del tamaño del programa, expresado en las líneas estimadas de código (LDC).
- Modelo 2, El modelo COCOMO intermedio calcula el esfuerzo del desarrollo de software en función del tamaño del programa y de un conjunto de “conductores de costo” que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.
- Modelo 3, El modelo COCOMO avanzado incorpora todas las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, diseño, etc.) del transcurso de ingeniería del software.

Pressman menciona que los modelos COCOMO están establecidos para tres prototipos de proyectos de software que con el uso de la terminología de Boehm son:

- ***modo orgánico:*** aquellos proyectos de software que son respectivamente pequeños y sencillos en donde trabajan pequeños equipos que poseen buena experiencia en la aplicación, sobre un conjunto de requisitos poco rígidos.
- ***modo semiacoplado:*** son los proyectos de software intermedios hablando de tamaño y complejidad, en donde los equipos tienen diversos niveles de experiencia, y además deben satisfacer requerimientos poco o medio rígidos.
- ***modo empotrado:*** son proyectos de software que deben ser desarrollados en un conjunto de hardware, software y restricciones operativas muy restringido. (Pressman, 2010)

Para poder calcular la estimación del software se puede utilizar el modelo de estimación de costos y esfuerzos COCOMO básico por cuanto este método se basa en la LDC o PF como datos para calcular el esfuerzo, tiempo y costo del desarrollo del proyecto.

En la siguiente tabla se describen los factores que se pueden calcular al usar el modelo COCOMO para la estimación del software en base a LDC:

<b>Parámetro</b>	<b>Descripción</b>
<i>Esfuerzo aplicado</i>	Este parámetro establece la cantidad de personas en meses que se estima para poder programar un determinado número de líneas de código.
<i>Tiempo de desarrollo</i>	Tiempo establecido en meses que se estima es necesario para poder programar un número de líneas de código.
<i>Costo del desarrollo del software</i>	El valor monetario que cuesta el desarrollo del proyecto, de esta forma se puede establecer cuanto resta para la empresa del presupuesto general del proyecto.

*Tabla 5. Parámetros del método COCOMO  
Elaborado por: Juan Carlos Moyota*

Las ecuaciones para medir los parámetros de la estimación del software con el uso del modelo COCOMO básico son:

$$E = a_b KLDC^{b_b} \quad (f1)$$

$$D = C_b E^{d_b} \quad (f2)$$

E = es el esfuerzo aplicado en personas-mes, D = es tiempo de desarrollo en meses cronológicos y KLDC = es número estimado de líneas de código distribuidas (en miles)

### **2.8.2. REGLAS PARA EL CONTEO DE LÍNEAS DE CÓDIGO**

Las siguientes reglas se establecen para el conteo de líneas de código:

- Líneas de código creadas por el personal del proyecto y la herramienta de generación de código.
- Una instrucción es una línea de código.
- Las declaraciones de variables se toman como instrucciones.
- Los comentarios en el código fuente no se cuentan como instrucciones.

## CAPITULO III

### EVALUACIÓN Y SELECCIÓN DEL FRAMEWORK PHP

#### 3.1. IDENTIFICACIÓN DE LAS HERRAMIENTAS

Los frameworks PHP en su mayoría son open source, pero estas herramientas también existen corporativas como es el caso de **Script Case**, esta herramienta ayuda en la generación de reportes, formularios de datos, exportación a archivos en diferentes formatos, soporte para muchas bases de datos, soporte para Google Maps, lector de código de barras, etc.

Al buscar referencias en la web se han identificado dos sitios especializados y el sitio web de una gran empresa de implementación de Tecnologías de la Información, se tomará como referencia y diferentes puntos de vista al trabajar con estos frameworks opensource, para seleccionar tres de estas herramientas se explican las razones a continuación:

- En el sitio **PHP Frameworks**<sup>4</sup> se encuentra una lista de los mejores frameworks PHP los cuales se listan a continuación:
  - Yii
  - CodeIgniter
  - Zend
  - CakePHP
  - Symfony

En el siguiente gráfico se presenta el número de votos para cada uno de los framework PHP del listado anterior, se identifica a Yii y Codeigniter como las herramientas más populares.

---

<sup>4</sup> <http://www.phpframeworks.com>

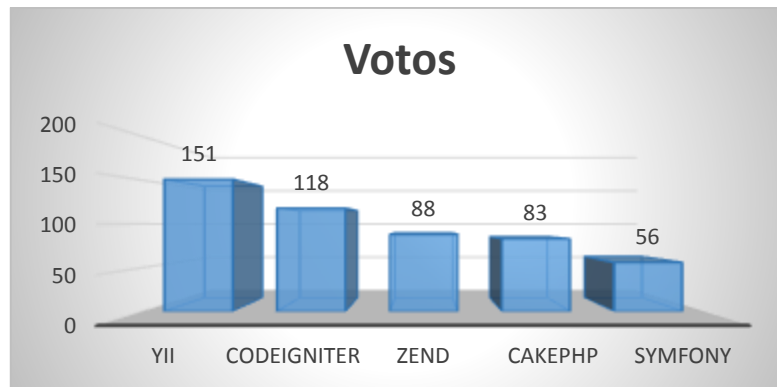


Figura 16. Número de votos obtenidos de PHP Frameworks  
Elaborado por: Juan Carlos Moyota

- En el sitio web LinuxLinks.com presenta una lista de todo el software disponible para el sistema operativo Linux, en este sitio web se encuentra un artículo titulado **10 Best Free Web Application Frameworks** (Los 10 mejores frameworks gratuitos para aplicaciones web), donde se muestra un listado de los mejores frameworks bajo los lenguajes: Ruby, Python y PHP, de esta lista podemos mencionar los siguientes frameworks PHP:
  - CodeIgniter
  - CakePHP
  - Zend Framework
  - Yii
  - Symfony
- En el portal de la empresa **Zfort Group**<sup>5</sup> se encuentra una infografía que ha tomado de foros y comunidades información lo cual determina 5 de los frameworks más populares para PHP:
  - Yii
  - CodeIgniter
  - Zend
  - Cake PHP

---

<sup>5</sup> Zfort Group es una empresa de consultoría con sede en Ucrania, Europa del Este. Zfort Group ofrece soluciones de desarrollo web, consultoría y asistencia en la implementación de TI para el crecimiento de empresas y organizaciones. Con más de 150 especialistas en el equipo y un departamento de reclutamiento fuerte, proporciona soluciones únicas para todas las tareas y necesidades de TI. Dirección web <http://www.zfort.com/>



- Symfony

Con la información presente en la **tabla 4** y al utilizar como referencia a los sitios web antes mencionados se realiza el análisis de los frameworks PHP eligiendo a: Yii, CodeIgniter y CakePHP por las siguientes razones:

- Los tres framework están presentes en los tres sitios web especializados y el sitio web de la empresa de implementación de Tecnologías de la Información tomados como referencia, con esto indicaríamos que los tres tienen la popularidad necesaria para ser elegidos por los desarrolladores.
- Utilizan el patrón de diseño MVC (Modelo Vista Controlador), divide la aplicación en tres capas con un mayor manejo y control.
- Tienen una documentación en línea muy buena para el manejo de cada uno de los framework, los tres framework tiene un sitio web con documentación propia.
- Pueden fácilmente conectarse a los más conocidos SGBD como: MySQL, MS SQL, PostgreSQL, Oracle y SQLite.
- Los tres frameworks tienen la capacidad de incorporar plugins, módulos, extensiones y librerías que ayudan a realizar tareas adicionales.

### **3.2. DESCRIPCIÓN DE LOS FRAMEWORKS**

A continuación se analiza cada uno de los tres framework PHP: CakePHP, CodeIgniter y Yii.

#### **3.2.1. CAKEPHP**

CakePHP es un framework libre, de código abierto, para el desarrollo rápido de aplicaciones para PHP. Es una estructura fundamental para ayudar a los programadores a crear aplicaciones web, permite trabajar de forma estructurada y rápida y sin pérdida de flexibilidad. (Cake Software Foundation, 2014)

##### **3.2.1.1. BREVE HISTORIA**

En 2005, Michal Tatarynowicz escribió una mínima versión de un framework rápido para aplicaciones en PHP. Encontró que ese fue el principio de

un muy buen framework. Michal publicó el framework bajo licencia MIT, mezclando Cake, y abriéndose a una comunidad de desarrolladores, quienes ahora mantienen Cake bajo el nombre de CakePHP.

### **3.2.1.2. REQUISITOS DEL SERVIDOR**

Los requisitos mínimos del servidor para poder instalar y usar CakePHP para el uso en el desarrollo de nuestros proyectos web son:

- Servidor web: Apache, LightHTTPD o Microsoft IIS
- Interprete PHP 5.2.6 o superior
- Permisos de los ficheros de configuración y carpetas
- Para conexión con bases de datos, los SGBD soportados son: MySQL, PostgreSQL, SQLite , SQL Server

### **3.2.1.3. CARACTERÍSTICAS**

Entre muchas de las características importantes de este framework se mencionan las siguientes:

- Comunidad activa y amigable.
- Compatible con la mayoría de entornos de servidores.
- Arquitectura modelo vista controlador (MVC)
- Integración con AJAX, Java script, Formularios HTML y mas
- URL limpias y personalizadas
- Validación de formularios incorporada
- Plantillas rápidas y flexibles
- Listas de control y acceso flexible
- Manejo de seguridad, sesiones y componentes para manejo de peticiones.
- Interacción con bases de datos mediante el CRUD integrado.

### **3.2.1.4. ESTRUCTURA PRINCIPAL DE DIRECTORIOS**

Luego descargar y descomprimir el paquete de CakePHP disponible en su página oficial <http://cakephp.org>, se tendrá una estructura de directorios parecida al siguiente gráfico:



*Figura 17. Estructura de directorios de CakePHP  
Elaborado por: Juan Carlos Moyota*

- **app**, este directorio almacena los archivos de la aplicación en construcción.
- **lib**, es el directorio núcleo de CakePHP, es aconsejable no modificar nada de este directorio.
- **vendors**, directorio donde se almacenan las bibliotecas de terceros para uso de nuestra aplicación.

#### **3.2.1.5. EXTENSIONES PARA GRÁFICOS ESTADÍSTICOS**

CakePHP al igual que muchos frameworks tienen la capacidad de incorporar extensiones de terceros para realizar cualquier tipo de tareas como: exportar datos a PDF, enviar correo, exportar datos a Excel entre otras tareas, una de estas tareas es crear gráficos estadísticos configurables. CakePHP tiene la posibilidad de trabajar con extensiones como: fusion charts, jgraph, Google Chart, etc.

#### **3.2.1.6. LICENCIA**

CakePHP está bajo la licencia MIT que lo hace perfecto para su uso en aplicaciones comerciales. Está permitido usar, copiar, modificar, publicar, distribuir, sub-licenciar y/o vender copias del software siempre y cuando tenga una copia del acuerdo de licencia publicado por Cake Software Foundation, Inc.

#### **3.2.2. CODEIGNITER**

CodeIgniter es un framework open source enfocado al desarrollo de aplicaciones web, utiliza un conjunto de herramientas para los desarrolladores que usan PHP.

Su objetivo es permitirle desarrollar proyectos mucho más rápido que lo que podría hacer si escribiera el código desde cero, proveyéndole un rico conjunto de bibliotecas para tareas comunes, así como y una interfaz sencilla y una estructura lógica para acceder a esas bibliotecas. CodeIgniter le permite enfocarse creativamente en su proyecto al minimizar la cantidad de código necesaria para una tarea dada. (CodeIgniter, 2012)

### **3.2.2.1. BREVE HISTORIA**

EllisLab es una empresa dedicada al desarrollo de aplicaciones web basadas en PHP, ubicada en la ciudad de Obregón (USA), esta empresa desarrolló la primera versión de CodeIgniter en febrero 28 del 2006, esta versión fue la 1.0 que hasta el 10 de marzo del mismo año con la versión 1.2 fueron beta. El 21 de marzo del 2006 se lanzó la primera versión estable CodeIgniter 1.2, hoy en día la última versión estable de este framework es la 2.1.4 publicada el 13 de julio del 2013.

### **3.2.2.2. REQUISITOS DEL SERVIDOR**

Los requisitos del servidor para la instalación de CodeIgniter son:

- Interprete de código PHP 5.0 o superior.
- Las bases de datos que se soportan actualmente son: MySQL (4.1 o superior), MS SQL, PostgreSQL, Oracle, SQLite, y ODBC.

### **3.2.2.3. CARACTERÍSTICAS**

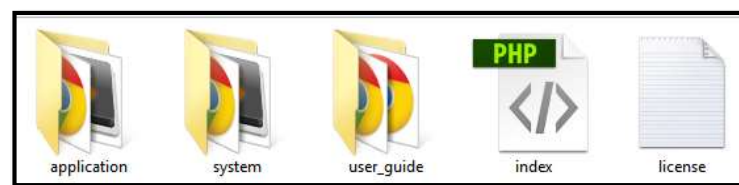
Entre las características de CodeIgniter se destacan las siguientes:

- Documentación, soporte y comunidad estable
- Basado en modelo vista controlador
- Compatibilidad como servidores bajo plataformas Windows, Linux o MacOS.
- Extremadamente liviano
- Formularios y validación
- Integración con AJAX, Java script, Formularios HTML y mas

- Manejo de sesiones
- Librerías de clases para: enviar correos mediante protocolos (Sendmail, SMTP, and Mail), manipulación de imágenes (cortar, redimensionar, rotar, etc.) y carga de archivos.
- Incorporación de plugins, clases y extensiones.
- URL amigables y a motores de búsqueda.
- Encriptación de datos
- Manejo de plantillas.

#### 3.2.2.4. ESTRUCTURA PRINCIPAL DE DIRECTORIOS

La estructura típica de directorios, luego de la descarga de la última versión estable disponible en <http://ellislab.com/codeigniter/download> es como como se muestra en la **Figura 22**:



*Figura 18. Estructura de directorios de CodeIgniter  
Elaborado por: Juan Carlos Moyota*

- ✓ **application:** Se encuentran todos los archivos de nuestro proyecto y es donde trabajaremos.
- ✓ **system:** Aquí se encuentran todas las librerías del Framework. No es nada recomendable modificarlos.
- ✓ **user\_guide:** La guía de usuario que podemos borrar sin problemas si lo vemos un gasto innecesario de espacio, ya que tenemos una guía online que está muy bien estructurada.

#### 3.2.2.5. EXTENSIONES PARA GRÁFICOS ESTADÍSTICOS

CodeIgniter tiene la característica de incorporar extensiones para realizar tareas comunes como exportación a formatos conocido, envío de correos, generación de gráficas estadística, entre muchas tareas que puede hacer este

sencillo pero completo framework. Como es objeto de esta investigación la generación de gráficas estadísticas se podrían incorporar librerías como: highcharts, jgraph, open flash chart.

#### **3.2.2.6. LICENCIA**

CodeIgniter es un framework con licencia Apache/BSD, es decir propietaria de la empresa EllisLab la cual permite usar, copiar, modificar y distribuir CodeIgniter y su documentación, con o sin modificaciones, para cualquier propósito, siempre y cuando se cumplan las siguientes condiciones:

- Una copia del acuerdo de licencia debe ser incluida con la distribución.
- Las redistribuciones del código fuente y redistribuciones en formato binario deben mantener el copyright de la empresa EllisLab.
- Cualquier archivo que haya sido modificado deberá llevar consigo notificaciones aclarando la naturaleza del cambio y los nombres de aquellos que lo hayan cambiado.
- Los productos derivados del Software no deben ser llamados "CodeIgniter", ni debería aparecer "CodeIgniter" en su nombre, sin previo permiso escrito por parte de EllisLab, Inc.

#### **3.2.3. YII**

Yii es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo (Yii Software LLC, 2008). Yii es un framework muy liviano de correr y equipado con soluciones de cacheo sofisticadas adecuado para la creación de sitios de gran tráfico como portales, foros, CMS, e-commerce, etc.

##### **3.2.3.1. BREVE HISTORIA**

Yii es un framework open source iniciado el 8 de enero del 2008 por Qiang Xue con el fin de solucionar algunos problemas con el framework Prado. En octubre del 2008 la primera versión alfa de Yii fue lanzada, el 3 de diciembre

del 2008 se presentó formalmente Yii 1.0. Hasta la fecha de este estudio la última versión estable y con soporte es la 1.1.14 lanzada el 11 de agosto del 2013.

“Tiene un equipo creciente de desarrollo y continúa ganando popularidad entre el día de rutina de desarrolladores PHP” (Winesett, 2010)

### **3.2.3.2. REQUISITOS DEL SERVIDOR**

Yii un framework orientado al desarrollo de aplicativos web y se necesita las siguientes características del servidor:

- Servidor web Apache con soporte para PHP.
- Lenguaje de programación PHP 5.1.0 o superior.
- Soporta bases de datos: MySQL 4.1 o superior, PostgreSQL 7.3 o superior, SQLite 2 y 3, Microsoft SQL Server 2000 o superior y Oracle

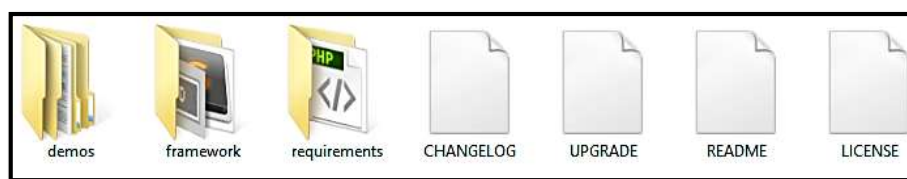
### **3.2.3.3. CARACTERÍSTICAS**

Hernandez, 2013 describe las siguientes características de este framework:

- Simple, sencillo y humilde
- Funciona en plataformas Windows, Linux y MacOS.
- Utiliza el patrón MVC
- Generación de Código, CRUD, Validaciones
- Soporte para Ajax, JQuery incluido.
- Gestión de errores y logging
- Manejo de urls Amigables
- Soporte para código de terceros e inclusión de extensiones
- Documentación y soporte bien documentado
- Soporte para temas y skins.

### **3.2.3.4. ESTRUCTURA DE DIRECTORIOS**

La estructura de directorios clásica de directorios de Yii antes de crear un proyecto de desarrollo se asemeja a como se muestra en la siguiente ilustración:



*Figura 19. Estructura de directorios de Yii  
Elaborado por: Juan Carlos Moyota*

**Demos:** directorio con ejemplos sencillos hechos con Yii.

**Framework:** directorio núcleo de Yii, contiene los archivos del framework para la construcción de nuestros proyectos.

**Requirements:** directorio que sirve para hacer un test al servidor y comprobar que este sea apto para ejecutar los archivos del Core Yii Framework.

### **3.2.3.5. EXTENSIONES PARA GRÁFICOS ESTADÍSTICOS**

Yii al igual que CodeIgniter y CakePHP brindan la posibilidad de incorporar extensiones para realizar tareas comunes entre las aplicaciones web, una de esas tareas es la generación dinámica de gráficos estadísticos, Yii tiene en su sitio web sus propias extensiones entre las que destacan: pCharts, Highcharts, flot-charts, open flash chart, chartjs, etc.

### **3.2.3.6. LICENCIA**

Según Wikipedia Yii es liberado bajo la Nueva Licencia BSD (Cláusula 3 de la licencia). Esto significa que es posible utilizar de forma gratuita para desarrollar cualquier aplicación web de código abierto o software privativo.

El texto que figura en la documentación oficial de Yii tiene licencia para el público bajo la GNU Free Documentation License (GFDL). En general, el contenido de la documentación Yii puede ser copiado, modificado y redistribuido siempre y cuando la nueva versión de subvenciones de las mismas libertades a los demás y reconozca a los autores del artículo de la documentación utilizada en Yii.



### 3.3. INDICADORES USADOS EN LA SELECCIÓN DEL FRAMEWORK

Después de haber conocido características de los tres frameworks PHP: CakePHP, CodeIgniter y Yii, con la ayuda de indicadores se evalúa, califica y selecciona el mejor framework PHP para realizar la aplicación de gestión de reportes estadísticos.

Los parámetros a evaluar se basan en la metodología QSOS<sup>6</sup> (*evaluación y selección de software libre*) estos se indican en la siguiente tabla:

<b>Indicador</b>	<b>Parámetro</b>
<i>Madurez</i>	Edad Estabilidad Historial de problemas conocidos
<i>Adopción</i>	Popularidad Referencias Contribuciones de la comunidad Libros Manuales y tutoriales
<i>Actividad</i>	Actividad en errores Actividad en funcionalidades Actividad en publicaciones
<i>Servicios</i>	Capacitación Soporte
<i>Empaquetado</i>	Código fuente Debian RedHat/Fedora MacOS X Windows
<i>Gestión</i>	Facilidad de uso Administración /supervisión
<i>Técnicas de adaptación</i>	Modularidad Modificación del código fuente Extensión del código
<i>Licencia y derechos de autor</i>	Licencia Propietarios del Copyright Patrocinador

Tabla 6. Indicadores y parámetros para selección del framework PHP  
Elaborado por: Juan Carlos Moyota

---

<sup>6</sup> Qualification and Selection of Open Source Software, metodología que permite identificar si el software cumple los requisitos técnicos, funcionales y estratégicos, mediante la clasificación y comparación de los diferentes productos, con el fin de tomar una decisión final.

### 3.3.1. INDICADOR 1: MADUREZ

Este indicador permite conocer aspectos como la edad en la que el software empezó a utilizarse con una versión estable y conocer un historial de errores bien documentados nos ayudan a ver el grado de madurez que tiene el framework.

***Edad:*** indica el año de lanzamiento de la primera versión estable del framework.

***Estabilidad:*** la existencia de versiones estables que incluyen correcciones de errores en versiones anteriores.

***Historial de problemas conocidos:*** conocer la existencia de documentación de los errores publicadas en páginas web especializadas, libros, revistas, etc.

### 3.3.2. INDICADOR 2: ADOPCIÓN

La adopción enfoca aspectos de referencias por usuarios que ya han utilizado una herramienta de software, documentación oficial y no oficial y soporte de la herramienta.

***Popularidad:*** identificar un número de usuarios, empresas desarrolladoras que utilizan un framework para sus proyectos.

***Referencias:*** existen de proyectos desarrollados con un framework.

***Contribuciones de la comunidad:*** existencia de comunidades de desarrollo que trabajan en mejoras del software.

***Libros:*** documentación en libros impresos o digitales.

***Manuales y tutoriales:*** documentación para el usuario en manuales, tutoriales, páginas web, documentos de internet.

### **3.3.3. INDICADOR 3: ACTIVIDAD**

La actividad es el movimiento de la comunidad o grupo de desarrollador que implementan mejoras en las funcionalidades del framework PHP.

*Actividad de errores:* medios para que el grupo de desarrollador de seguimiento a los errores en las funcionalidades del framework.

*Actividad en funcionalidades:* incorporación de nuevas funcionalidad por parte del desarrollador o grupo desarrollador.

*Actividades en publicaciones:* el desarrollador o grupo de desarrollo planifica nuevas versiones a futuro del framework.

### **3.3.4. INDICADOR 4: SERVICIOS**

Los servicios contemplan la existencia de capacitación, cursos, conferencias, soporte por parte de terceros y todo tipo de capacitación del framework.

*Capacitación:* oferta de cursos de capacitación, tutoriales en video, o cualquier otra forma de capacitación.

*Soporte:* otro tipo de soporte adicional al de la comunidad o foro de desarrolladores.

### **3.3.5. INDICADOR 5: EMPAQUETADO**

El empaquetado tiene que ver con el paquete fuente que sea fácil de instalar, compatible con los sistemas operativos comunes.

*Código fuente:* Grado de dificultad en la instalación del código fuente, configuraciones adicionales, etc.

*Debian:* compatibilidad del framework con la distribución Linux Debian

**RedHat/Fedora:** compatibilidad del framework con la distribución Red/Hat o derivados.

**MacOS:** compatibilidad del framework con MacOS X

**Windows:** compatibilidad del framework con Windows.

### **3.3.6. INDICADOR 6: GESTIÓN**

La gestión establece el grado de dificultad de manejar y utilizar el paquete de software con la ayuda de interfaces gráficas y herramientas de administración que ayuden a un mayor dominio del framework como generación de código, gestión de operaciones CRUD, etc.

**Facilidad de uso:** capacidad del software para permitir al usuario manejar correctamente sus funciones.

**Administración y supervisión:** existencia de herramientas internas o externas de administración.

### **3.3.7. INDICADOR 7: TÉCNICAS DE ADAPTABILIDAD**

Facilidad de incorporar algún tipo de complementos para integrarse con otras tecnologías como Ajax, JavaScript, etc. o aspectos que ayuden a mejorar el desarrollo de proyectos web.

**Modularidad:** existencia de módulos, plugins, extensiones para adaptarse a otras tecnologías.

**Modificación del código fuente:** facilidad de modificar el código fuente con ayuda de herramientas editores de código.

**Extensión del código:** Existencia de algún complemento como plugin, extensión que ayude a la gestión dinámica de contenido.

### 3.3.8. INDICADOR 8: LICENCIA Y DERECHOS DE AUTOR

La licencia y derechos de autor identifican la licencia de código abierto que utiliza el framework para no tener problemas al implementar proyectos desarrollados con el uso de un framework. Nos enfocamos en los parámetros: derechos de autor, tipo de licencia y el patrocinador del proyecto.

**Licencia:** tipo de licencia open source que posee el framework PHP.

**Propietarios del Copyright:** empresa, grupo o persona propietarios de los derechos de autor.

**Patrocinador:** si el framework es patrocinado por una persona, empresa, organización que patrocina el desarrollo del framework.

### 3.4. NOTACIÓN Y REGLAS DE PUNTUACIÓN

El método QSOS se centra en los criterios para estimar los riesgos incurridos por un usuario al usar un software. Para cada parámetro se establece la norma de puntuación siguiente:

<b>Equivalencia</b>	<b>Puntuación</b>
No cubierta	0
Parcialmente cubierta	1
Completamente cubierta	2

*Tabla 7. Puntuación de los parámetros  
Elaborado por: Juan Carlos Moyota*

Los siguientes cuadros se detallan cada una de estas categorías, especifican la regla de notación que se utilizará para cada criterio, esta información fué tomada del sitio web de la metodología QSOS:

<b>Indicador</b>	<b>Parámetro</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Madurez</b>	Edad	Menos de 3 meses	Entre 3 meses y 3 años	Mayor a 3 años
	Estabilidad	Software inestable con numerosas versiones o parches que generan efectos secundarios	Existen versiones de producción, estables pero antiguas.	Las versiones proporcionan correcciones de errores, pero sobre todo nuevas funcionalidades
	Historial, problemas conocidos	Se conoce varios problemas de software que pueden ser prohibitivos	No hay problemas conocidos o crisis	Buena gestión histórica de situaciones críticas

*Tabla 8. Reglas de notación para el Indicador 1  
Elaborado por: Juan Carlos Moyota*

<b>Indicador</b>	<b>Parámetro</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Adopción</b>	Popularidad	Muy pocos usuarios identificados	Uso detectable mediante Internet	Numerosos usuarios, numerosas referencias
	Referencias	Ninguno	Pocas referencias, usos no críticos	A menudo implementado para aplicaciones.
	Contribuciones de la comunidad	Ninguna comunidad o sin actividad real (foro, lista de correo...)	Comunidad existente con una notable actividad	Fuerte comunidad: gran actividad en los foros, numerosos colaboradores y defensores
	Libros	Ningún libro sobre el software	Al menos de cinco libros sobre el software disponibles	Más de 5 libros sobre el software están disponibles en varios idiomas
	Manuales y tutoriales	Muy pocos o ningún manual o tutorial sobre el software, o los tutoriales son poco relevantes y confusos.	Manuales o tutoriales con contenido medianamente relevante y poco confuso	Manuales o tutoriales sobre el tema con contenido relevante y disponibles en varios idiomas

*Tabla 9. Reglas de notación para el Indicador 2  
Elaborado por: Juan Carlos Moyota*

<b>Indicador</b>	<b>Parámetro</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Actividad</b>	Actividad en errores	Lenta o ninguna reacción en el foro o en la lista de correo con respecto a las correcciones de errores en las notas de lanzamiento	Actividad detectable, pero sin un proceso Claramente expuesto, tiempo de reacción y resolución largo	Reacción fuerte, basado en roles y asignación de tareas
	Actividad en funcionalidades	Ninguna o pocas funcionalidades nuevas	Progreso del software impulsado por el equipo central o por solicitud del usuario sin ningún tipo de proceso claramente explicado	Herramienta(s) para gestionar las peticiones de características, una fuerte interacción con el plan de trabajo
	Actividades en publicaciones	Muy débil actividad en ambas versiones, producción y desarrollo	Actividad en versiones de producción y desarrollo. Frecuentes versiones menores (corrección de errores)	Importante actividad con frecuentes versiones menores (corrección de errores) y versiones principales planificada en relación con el plan de trabajo

*Tabla 10. Reglas de notación para el Indicador 3  
Elaborado por: Juan C. Moyota*

<b>Indicador</b>	<b>Parámetros</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Servicios</b>	Capacitación	Ninguna oferta de capacitación identificada	Existe oferta, pero está limitada geográficamente, en términos de lenguaje o suministrado por un proveedor único	Amplia oferta, brindada por varios proveedores en varios idiomas y se divide en módulos de niveles graduales
	Soporte	Ninguna oferta de soporte, excepto a través de foros públicos y listas de correo	Existe oferta, pero se limita a un único proveedor	Múltiples proveedores de servicios con un fuerte compromiso

*Tabla 11. Reglas de notación para el Indicador 4  
Elaborado por: Juan C. Moyota*

<b>Indicador</b>	<b>Parámetro</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Empaquetado</b>	<b>Código fuente</b>	El software no se puede instalar desde el código fuente sin mucho trabajo	Instalación desde la fuente es limitada y depende de condiciones muy estrictas	La instalación desde el código fuente es fácil
	<b>Debian</b>	El software no está empaquetado para Debian	Existe un paquete Debian, pero tiene problemas importantes o no tiene soporte oficial	El software se empaqueta en la distribución
	<b>RedHat/Fedora</b>	El software no está empaquetado para RedHat/Fedora	Existe un paquete, pero tiene problemas importantes o no tiene soporte oficial	El software se empaqueta en la distribución
	<b>MacOS X</b>	El software no está empaquetado para MacOS X	Existe un paquete, pero tiene problemas importantes o no tiene soporte oficial	Existe un paquete oficial de instalación para MacOS X
	<b>Windows</b>	El proyecto no se puede instalar en Windows	Existe un paquete, pero es limitado o tiene problemas importantes o sólo cubre versiones específicas de Windows	Es totalmente compatible con Windows.

*Tabla 12. Reglas de notación para el Indicador 5  
Elaborado por: Juan C. Moyota*



<b>Indicador</b>	<b>Parámetro</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Gestión</b>	Facilidad de uso	Difícil de usar, requiere un conocimiento en profundidad de la funcionalidad del software	Facilidad de uso dada por la asistencia entre usuarios. Presencia de Interfaces Hombre - Máquina.	Software muy orientado al usuario: ayuda contextual, interfaz de usuario atractiva
	Administración y supervisión	Ninguna funcionalidad de administración o de supervisión	Existen funcionalidades pero están incompletas y requieren ser mejoradas	Funcionalidades administrativas y supervisión, completas y fáciles de usar.

*Tabla 13. Reglas de notación para el Indicador 6  
Elaborado por: Juan Carlos Moyota*

<b>Indicador</b>	<b>Parámetro</b>	<b>0</b>	<b>1</b>	<b>2</b>
<b>Técnicas de adaptabilidad</b>	Modularidad	Software Monolítico	Presencia de módulos de alto nivel que permiten un primer nivel de adaptación de software	Modular, lo que permite una fácil adaptación del software mediante la selección de módulos o incluso el desarrollo de módulos nuevos
	Modificación del código fuente	Todo a mano	Recompilación posible, pero compleja sin ninguna herramienta o documentación	Recompilación con herramientas y documentación proporcionada
	Extensión del código	Cualquier modificación requiere la re compilación del código	Arquitectura diseñada para extensión estática pero requiere recompilación	Principio de plugin, arquitectura diseñada para la extensión dinámica sin necesidad de recompilar

*Tabla 14. Reglas de notación para el Indicador 7  
Elaborado por: Juan Carlos Moyota*

<b>Indicador</b>	<b>Parámetros</b>	<b>0</b>	<b>1</b>	<b>2</b>
Licencia y derechos de autor	Licencia	Licencia muy permisiva como BSD o Apache License	Licencia moderadamente permisiva, situada entre ambos extremos (GPL y BSD), licencia dual en función del tipo de usuario (persona, empresa.) o de sus actividades	Licencia muy estricta, como la GPL
	Propietarios del Copyright	Derechos en manos de unos pocos individuos o entidades, por lo que es más fácil cambiar la licencia	Derechos en manos de muchas personas, poseen el código de una manera homogénea, lo que hace muy difícil modificar la licencia	Derechos en manos de una persona jurídica en quien confía la comunidad (por ejemplo, la FSF o ASF)
	Patrocinador	Software no tiene un patrocinador, el equipo principal no es remunerado	El software tiene un patrocinador único el cual determina las estrategias	El software es patrocinado por la industria

*Tabla 15. Reglas de notación para el Indicador 8  
Elaborado por: Juan Carlos Moyota*

### 3.5. EVALUACIÓN DE LOS INDICADORES

Para utilizar el framework adecuado para el desarrollo de la aplicación web que genere estadísticas, se procede a realizar un proceso de calificación.

Los framework PHP tienen características que se pueden calificar con valores de escala cuantitativa descrita en la **Tabla 6** para la evaluación de los indicadores.

Cada parámetro es comentado la razón por la cual se le da un valor en la escala de 0 a 2, las razones se basan en la documentación e investigación de sus características.

### 3.5.1. MADUREZ

En la **Tabla 16** se califica el indicador madurez con sus parámetros: edad, estabilidad e historial de problemas conocidos, estos nos ayudarán a conocer la evolución del framework en base a sus versiones actuales, conocer los problemas o fallos que tuvo desde su inicio hasta la actualidad y si estos errores fueron documentados por los desarrolladores para corregir dichos errores.

Parámetro	CakePHP	CodeIgniter	Yii
Edad	2 Fecha de lanzamiento 2005	2 Fecha de lanzamiento 2006	2 Fecha de lanzamiento 2008
Estabilidad	2 Versión estable 2.5.1	2 Versión estable 2.1.4	2 Versión estable 1.1.14
Historial, problemas conocidos	2 Historial de problemas conocidos documentados en el sitio web de cada framework		

Tabla 16. Evaluación del indicador madurez  
Elaborado por: Juan Carlos Moyota

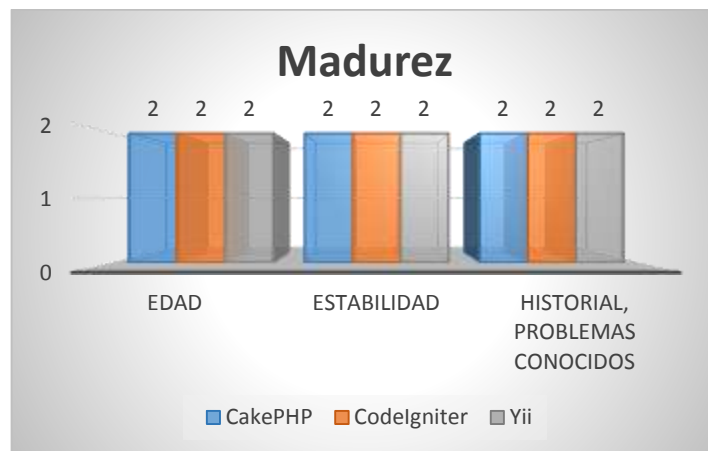


Figura 20. Indicador madurez  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	2.0	2.0	2.0

Tabla 17. Promedios para el indicador madurez  
Elaborado por: Juan Carlos Moyota

**Comentario:** Con los resultados de la tabla 17 nos atrevemos a decir que los tres frameworks tienen el mismo nivel de madurez ya que los tres tienen un promedio de 2.0 en sus parámetros de calificación.

### 3.5.2. ADOPCIÓN

La adopción nos ayudará a conocer si la herramienta ha sido o es usada por empresas desarrolladoras, la existencia de proyectos desarrollados con el uso de un determinado framework, foros de comunidades enfocadas exclusivamente a un framework para ayudar a dar soluciones a problemas comunes y la documentación oficial mediante libros, revistas, medios electrónicos, etc. La siguiente tabla muestra la puntuación del indicador *adopción*:

Parámetro	CakePHP	CodeIgniter	Yii
Popularidad	1 Uso detectable en Github	2 Empresas como: GF sistemas, ITD y Acrux software Factory	2 Empresas como: GF sistemas, comunidades en redes sociales, etc.
Referencias	2	2	2
Contribuciones de la comunidad	2 Comunidad activa, foros, grupos en google.	2 Comunidad activa, foros, grupos en google.	2 Comunidad activa, foros, grupos en google, redes sociales.
Libros	2	2	2
	Existe más de 5 libros disponibles en Amazon		
Manuales y tutoriales	2 Existe documentación en línea varios idiomas.	2 Guías de usuario, tutoriales en YouTube.	2 Tutoriales y documentación en línea en varios idiomas, videos en YouTube, viveo, etc.

*Tabla 18. Indicador adopción  
Elaborado por: Juan Carlos Moyota*

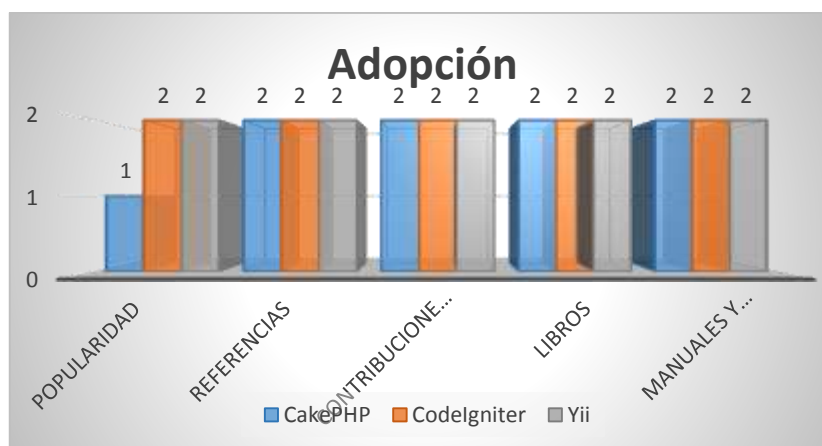


Figura 21. Indicador adopción  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	1.8	2	2

Tabla 19. Promedio del indicador adopción  
Elaborado por: Juan Carlos Moyota

**Comentario:** Al hablar de adopción mencionamos la existencia de referencias, documentación en libros, manuales y tutoriales que ayudan como guía al desarrollador, contribuciones por parte de su comunidad y el uso por parte de empresas en sus proyectos, la Tabla 19 se muestran el promedio obtenido por las tres herramientas framework, donde CodeIgniter y Yii tienen un promedio de 2.

### 3.5.3. ACTIVIDAD

Cualquier herramienta de desarrollo como es el caso de un framework debe estar en constante evolución, hablamos de la actividad en la solución de errores, nuevas funcionalidades y la producción de futuras versiones del framework.

En la Tabla 20 se califica los parámetros para el indicador actividad:

Parámetro	CakePHP	CodeIgniter	Yii
Actividad en errores	2	2	2
	Seguimiento de errores a través de foros disponibles en sus respectivos sitios web.		
Actividad en funcionalidades	1 La evolución del producto está a cargo del grupo de CakePHP	1 La evolución del producto está a cargo de la empresa EllisLab	2 Foro para petición y contribución de nuevas funcionalidades.
Actividades en publicaciones	1 Existe actividad documentada en las versiones de producción y desarrollo.	1 Existe actividad documentada en las versiones de producción y desarrollo.	2 Producción de la nueva versión 2.0 en marcha.

Tabla 20. Indicador actividad  
Elaborado por: Juan Carlos Moyota

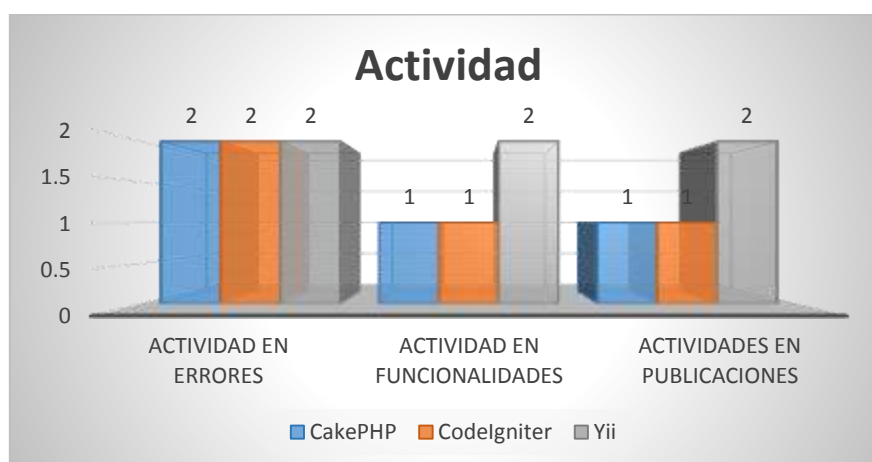


Figura 22. Indicador actividad  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	1.5	1.5	2

Tabla 21. Promedio del indicador actividad  
Elaborado por: Juan Carlos Moyota

**Comentario:** En la tabla 21 se muestran los promedios del indicador actividad el cual es un factor importante para el desarrollador ya que contribuye a corregir posible errores encontrados en la versión actual de un determinado paquete, añadir nuevas funcionalidades que ayudan a desarrollar de manera más eficiente nuestros proyectos, la puntuación promedio de los tres parámetros determina que Yii con

2.0 de calificación tiene un mayor grado de actividad en errores, funcionalidades y publicaciones recientes.

### 3.5.4. SERVICIOS

Los servicios son herramientas de ayuda adicional a la documentación del framework utilizados en el desarrollo de proyectos, estas pueden ser de dos tipos: la existencia de instituciones de capacitación en el uso de una herramienta y el soporte de empresas de consultoría en dichas herramientas. En la Tabla 22 calificamos los parámetros que involucran los servicios:

Parámetros	CakePHP	CodeIgniter	Yii
Capacitación	2	2	2
	Ofertas de cursos por empresas en muchos países, cursos en varios idiomas, video tutoriales en YouTube.		
Soporte	2	2	2
	Soporte de las comunidades, foros, blogs y en redes sociales		

Tabla 22. Indicador servicios  
Elaborado por: Juan Carlos Moyota

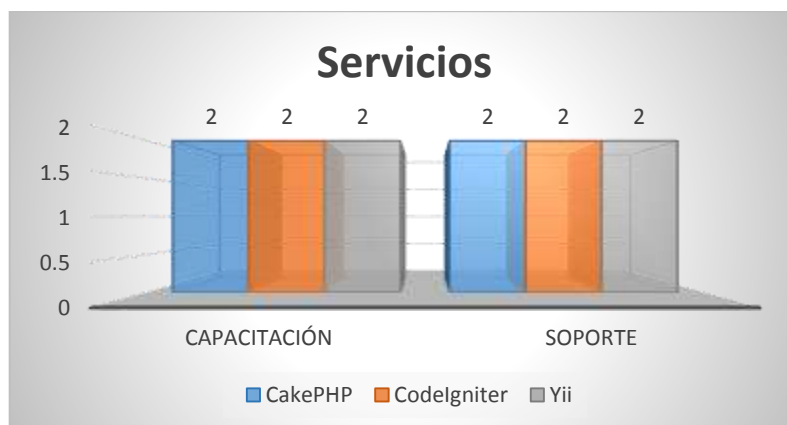


Figura 23. Indicador servicios  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	2	2	2

Tabla 23. Promedio del indicador servicios  
Elaborado por: Juan Carlos Moyota

**Comentario:** En la tabla 23 se indica la puntuación promedio del indicador servicios, en este caso los tres frameworks tienen 2.0 de promedio y cuentan con

el suficiente soporte por empresas de capacitación en estas herramientas y existencia de videos tutoriales en canales como YouTube y otros.

### 3.5.5. EMPAQUETADO

El empaquetado se refiere a la facilidad de instalación del código fuente y a su compatibilidad con varias plataformas de sistemas operativos. La compatibilidad es muy importante por cuanto las aplicaciones web tienden a migrar de un servidor a otro y estos pueden estar en diferentes plataformas de sistemas operativos. Por otro lado el código fuente puede depender de sistemas operativos, archivos de configuración o bibliotecas adicionales que impidan que el framework funcione de forma correcta. En la tabla 24 se establece la calificación de los parámetros del *empaquetado*:

Parámetro	CakePHP	CodeIgniter	Yii
Código fuente	1	1	1
	Necesita de configuraciones en el servidor.		
Debian	2	2	2
	Paquetes disponibles para distribuciones Linux		
RedHat/Fedora	2	2	2
MacOS X	2	2	2
	Paquetes disponibles para sistemas operativos MacOS		
Windows	2	2	2
	Paquetes disponibles para sistemas operativos Windows		

Tabla 24. Indicador empaquetado  
Elaborado por: Juan Carlos Moyota

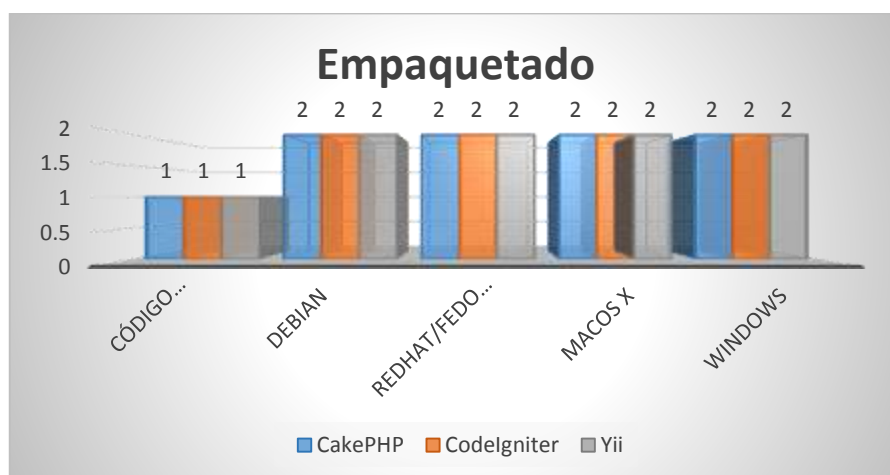


Figura 24. Indicador empaquetado  
Elaborado por: Juan Carlos Moyota



	<b>Framework</b>		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	1.8	1.8	1.8

*Tabla 25. Promedio del indicador empaquetado  
Elaborado por: Juan Carlos Moyota*

**Comentario:** en la **Tabla 25** se establece la puntuación para el indicador empaquetado, con este indicador establecemos la facilidad de instalación del paquete y su compatibilidad con la mayoría de sistemas operativos más comunes en computadores actuales, el promedio obtenido por los tres frameworks es 1.8 al existir algunas configuraciones en el servidor web específicamente para su instalación y los tres son compatibles con las plataformas de sistemas operativos más usados.

### 3.5.6. GESTIÓN

La gestión ayuda al desarrollador a facilitar la generación de código mediante el uso de interface gráfica y la existencia de herramientas de administración y supervisión de la aplicación lo que ayuda a la generación de código y a su vez el cumplimiento del plan de trabajo planteado. En la Tabla 26 se califica los parámetros de gestión y uso de la aplicación:

<b>Parámetro</b>	<b>CakePHP</b>	<b>CodeIgniter</b>	<b>Yii</b>
Facilidad de uso	1 Posibilidad de incorporar interfaz de CRUD como BROWNIE PHP	1 Posibilidad de incorporar interfaces de CRUD como GROCERY CRUD	2 Módulo de administración de CRUD incorporado
Administración y supervisión	2 Posibilidad de incorporar interfaz gráfica.	2 Posibilidad de incorporar interfaz gráfica.	2 Módulo para generación de código.

*Tabla 26. Indicador gestión  
Elaborado por: Juan Carlos Moyota*

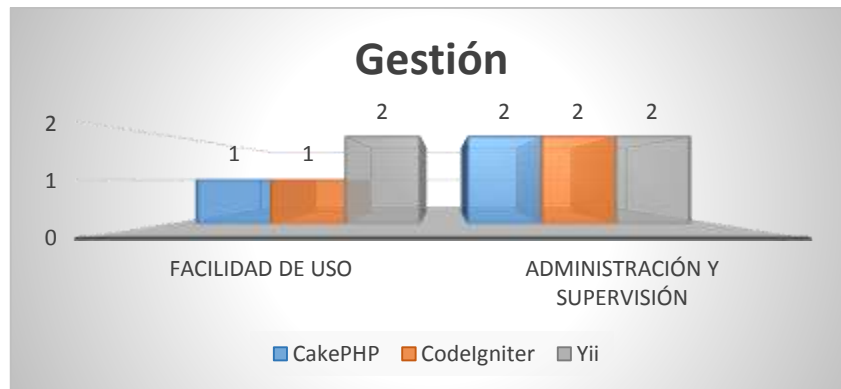


Figura 25. Indicador gestión  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	1.5	1.5	2

Tabla 27. Promedio del indicador gestión  
Elaborado por: Juan Carlos Moyota

**Comentario:** en la tabla 27 se identifica a Yii con una puntuación de 2.0 como la herramienta que posee facilidad de uso por incorporar en su paquete una interfaz gráfica de usuario para la generación de código y operaciones CRUD, que a diferencia de los otros se puede integrar herramientas para operaciones CRUD y generación de código.

### 3.5.7. TÉCNICAS DE ADAPTABILIDAD

Las técnicas de adaptabilidad se refiere a la incorporación de nuevos módulos, plugins o librerías que ayuden a realizar ciertas tareas adicionales a las comunes, por ejemplo enviar correos, exportar a formatos conocidos como Excel y PDF, generar gráficas estadísticas, etc. También puede existir la posibilidad de interactuar con otras tecnologías como: Ajax, JavaScript, JQuery, Bootstrap, etc., los programadores avanzados podrían modificar el código fuente del framework y mejorar ciertas funcionalidades. En la Tabla 28 establecemos las puntuaciones para los parámetros tomados en cuenta al referirnos a las técnicas de adaptabilidad:

Parámetro	CakePHP	CodeIgniter	Yii
Modularidad	2	2	2
	Incorporación con tecnologías como Ajax, JQuery, Bootstrap, etc.		
Modificación del código fuente	2	2	2
	Para la modificación del código fuente existe documentación proporcionada por el sitio web del Framework.		
Extensión del código	2	2	2
	Instalación de plugins, extensiones para realizar tareas adicionales.		

Tabla 28. Indicador técnicas de adaptabilidad  
Elaborado por: Juan Carlos Moyota

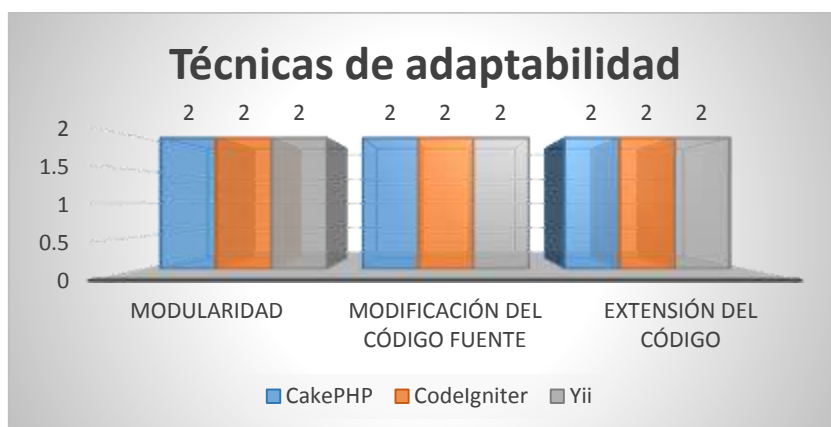


Figura 26. Indicador técnicas de adaptabilidad  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	2	2	2

Tabla 29. Promedio del indicador técnicas de adaptabilidad  
Elaborado por: Juan Carlos Moyota

**Comentario:** en la Tabla 29 se indica el promedio del indicador técnicas de adaptabilidad donde se puede observar que los tres frameworks poseen características similares basándose en la puntuación promedio de 2.0 para cada uno, estas técnicas de adaptabilidad nos ayuda a extender los aplicativos a realizar otras funciones, integrar con otras tecnología web y la facilidad de modificación del código fuente.

### 3.5.8. LICENCIA Y DERECHOS DE AUTOR

Existen muchas licencias de uso de software libre que permiten usar de forma libre el framework pero además modificar el código fuente con el uso de algún tipo de herramienta, además las herramientas framework tienen propietarios de copyright y un patrocinador del proyecto. La Tabla 30 muestra la puntuación del indicador licencia y derechos de autor.

Parámetros	CakePHP	CodeIgniter	Yii
Licencia	2 Licencia MIT	2 Licencia Apache/BSD	2 Licencia BSD
Propietarios del Copyright	0	0	0
Patrocinador	1 <b>Cake Development Corporation</b> propietaria del proyecto.	1 El software el patrocinado por la empresa <b>EllisLab</b>	1 <b>Yii Software LLC</b> patrocinador del proyecto.

Tabla 30. Indicador licencia y derechos de autor  
Elaborado por: Juan Carlos Moyota

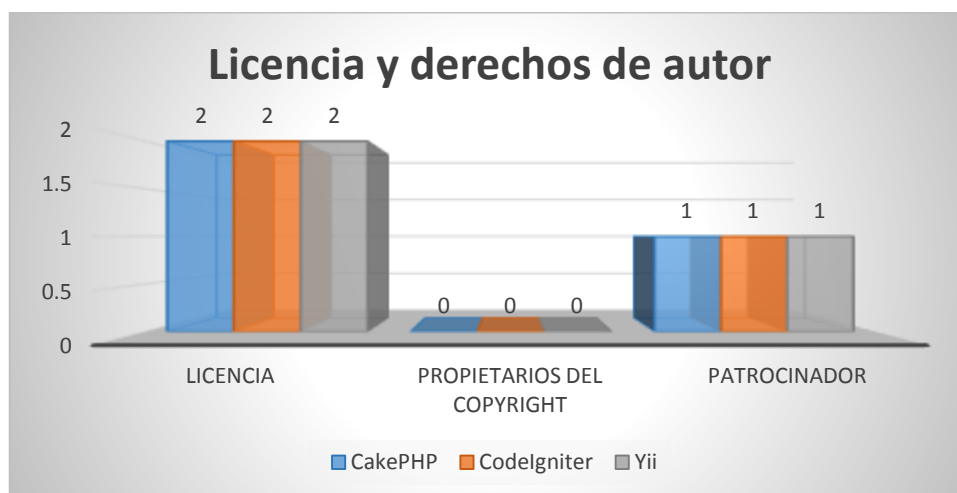


Figura 27. Indicador licencia y derechos de autor  
Elaborado por: Juan Carlos Moyota

	Framework		
	CakePHP	CodeIgniter	Yii
<b>Promedio</b>	1	1	1

Tabla 31. Promedio del indicador licencia y derechos de autor  
Elaborado por: Juan Carlos Moyota

**Comentario:** al hablar de software libre la licencia y los derechos de autor nos indican que podemos cambiar en el software para adaptar a nuestras necesidades, en la tabla 31 encontramos que los tres frameworks tienen el mismo promedio en la calificación de los indicadores de licencia y derechos de autor por lo cual podríamos utilizar cualquiera de los tres para el desarrollo del sistema web sin tener ningún tipo de problema legal al utilizar cualquiera de las herramientas.

### 3.6. RESULTADOS DE LA EVALUACIÓN DE LOS FRAMEWORKS

Los resultados de los indicadores utilizados para la selección de las herramientas framework: CakePHP, CodeIgniter y Yii se basan en investigación teórica, referencias de los sitios web oficiales de cada framework, libros de diferentes autores y sitios web especializados. Estos resultados son producto de los indicadores establecidos con anterioridad, donde cada indicador se establece bajo parámetros para cada framework, con esto se realiza una comparación cualitativa expresada en valores cuantitativos.

En la **Figura 28** se establece la comparación de las tres herramientas con su respectiva puntuación promedio según la puntuación de la **Tabla 6**:

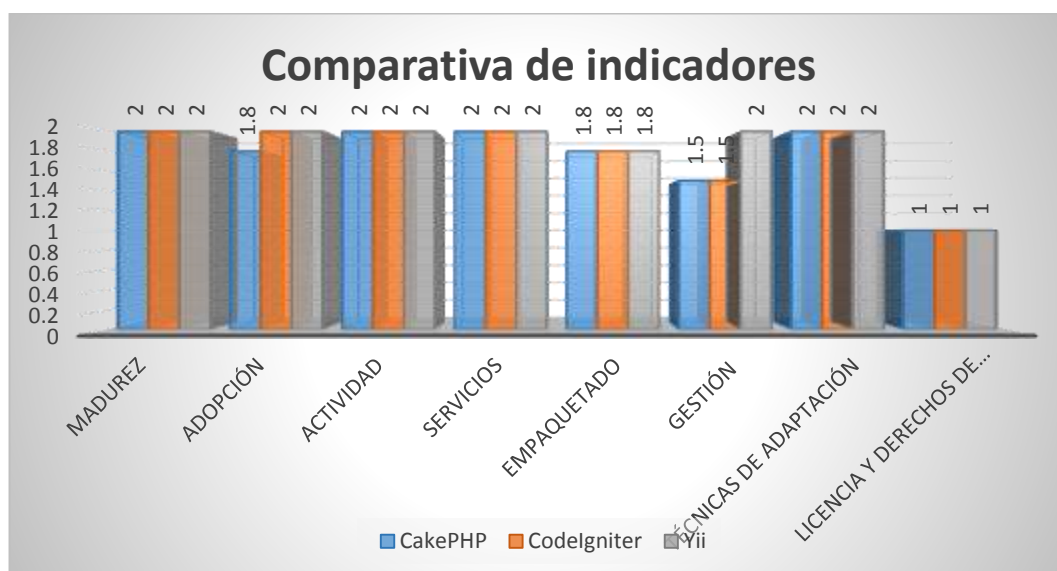


Figura 28. Indicadores evaluados, y sus promedios  
Elaborado por: Juan Carlos Moyota

En la tabla 32 se resume toda la puntuación de los indicadores y parámetros para los tres framework: CakePHP, CodeIgniter y Yii, además se observa los promedios de cada indicador:

<b>Indicador</b>	<b>Parámetro</b>	<b>CakePHP</b>	<b>CodeIgniter</b>	<b>Yii</b>
Madurez	Edad	2	2	2
	Estabilidad	2	2	2
	Historial de problemas conocidos	2	2	2
	<b>Promedio</b>	<b>2</b>	<b>2</b>	<b>2</b>
Adopción	Popularidad	1	2	2
	Referencias	2	2	2
	Contribuciones de la comunidad	2	2	2
	Libros	2	2	2
	Manuales y tutoriales	2	2	2
	<b>Promedio</b>	<b>1.8</b>	<b>2</b>	<b>2</b>
Actividad	Actividad en errores	2	2	2
	Actividad en funcionalidades	1	1	2
	Actividad en publicaciones	1	1	2
	<b>Promedio</b>	<b>2</b>	<b>2</b>	<b>2</b>
Servicios	Capacitación	2	2	2
	Soporte	2	2	2
	<b>Promedio</b>	<b>2</b>	<b>2</b>	<b>2</b>
Empaquetado	Código fuente	1	1	1
	Debian	2	2	2
	RedHat/Fedora	2	2	2
	MacOS X	2	2	2
	Windows	2	2	2
	<b>Promedio</b>	<b>1.8</b>	<b>1.8</b>	<b>1.8</b>
Gestión	Facilidad de uso	1	1	2
	Administración /supervisión	2	2	2
	<b>Promedio</b>	<b>1.5</b>	<b>1.5</b>	<b>2</b>
Técnicas de adaptación	Modularidad	2	2	2
	Modificación del código fuente	2	2	2
	Extensión del código	2	2	2
	<b>Promedio</b>	<b>2</b>	<b>2</b>	<b>2</b>
Licencia y derechos de autor	Licencia	2	2	2
	Propietarios del Copyright	0	0	0
	Patrocinador	1	1	1
	<b>Promedio</b>	<b>1</b>	<b>1</b>	<b>1</b>

*Tabla 32. Resumen de resultados  
Elaborado por: Juan Carlos Moyota*

En la **Tabla 33** se puede observar los porcentajes que representa cada promedio para cada framework PHP.

Indicador	Framework		
	CakePHP	CodeIgniter	Yii
Madurez	2	2	2
Adopción	1.8	2	2
Actividad	2	2	2
Servicios	2	2	2
Empaquetado	1.8	1.8	1.8
Gestión	1.5	1.5	2
Técnicas de adaptación	2	2	2
Licencia y derechos de autor	1	1	1
<b>Promedio</b>	<i>1.7625</i>	<i>1.7875</i>	<i>1.85</i>
<b>Porcentaje</b>	<i>88.13%</i>	<i>89.38%</i>	<i>92.50%</i>

Tabla 33. Porcentajes de cada framework  
Elaborado por: Juan Carlos Moyota

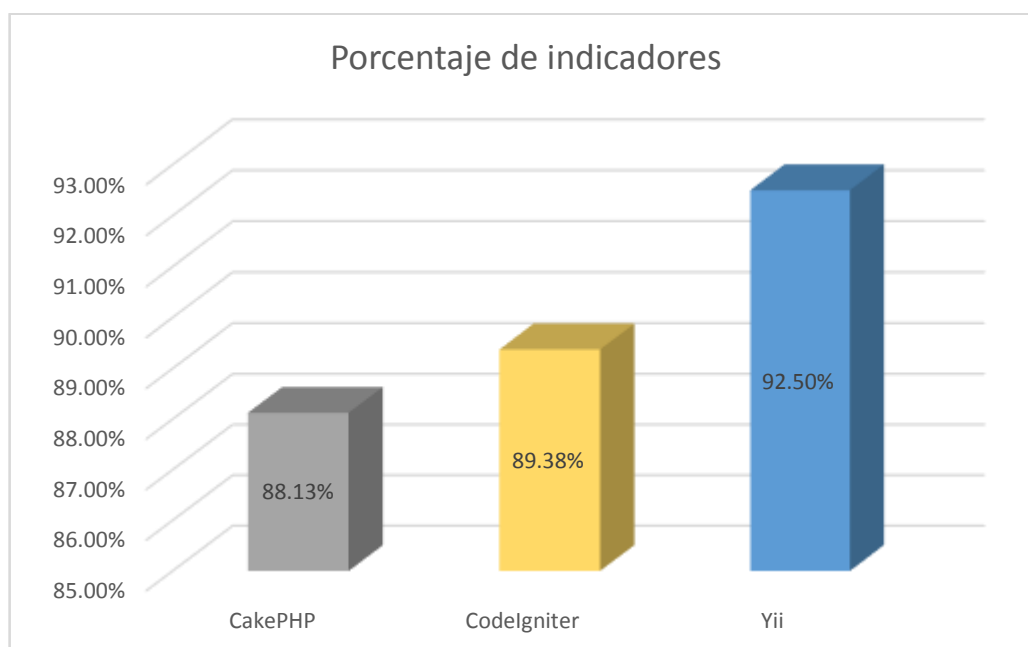


Figura 29. Porcentaje de todos los indicadores  
Elaborado por: Juan Carlos Moyota

Con los porcentajes obtenidos, podemos hacer las siguientes afirmaciones:

- El framework CakePHP obtuvo 1.76 que representa el 88.13% en promedio de todos los indicadores evaluados.
- La puntuación para el framework CodeIgniter es de 1.78 lo que representa el 89.38% del promedio de todos los indicadores evaluados.
- Existe un 92.5% que equivale a 1.85 de promedio de todos los indicadores evaluados para el framework Yii.

Con los resultados de la **Figura 29** podemos afirmar que de las tres herramientas frameworks PHP, Yii es la más adecuada con un 92.50% en todos los indicadores, por consiguiente se utilizará el framework Yii para realizar el sistema web para la gestión de reportes estadísticos.



## CAPITULO IV

### DESARROLLO DEL SISTEMA WEB CALIAGUA

#### 4.1. DEFINICIÓN DE LA METODOLOGÍA

El desarrollo del sistema web se desarrolla con ingeniería web establecida por Pressman, seguiremos los pasos que se presentan en la **Figura 30** nos indica el proceso para desarrollar un sistema web.

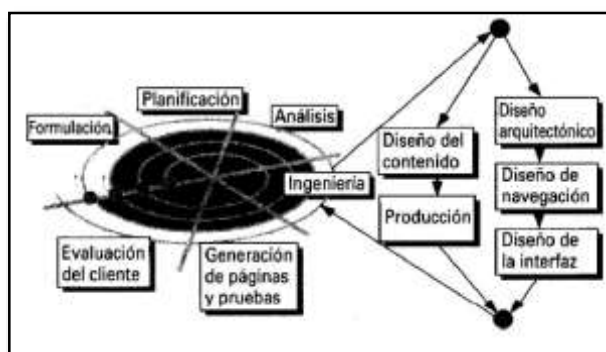


Figura 30. Modelo de proceso IWEB

Fuente: Pressman, R. (2010). *Ingeniería de Software un enfoque práctico*. [Figura]. Madrid. McGRAW-HILLDNTERAMERICANA DE ESPAÑA

#### 4.2. FORMULACIÓN DEL PROBLEMA

La SENAGUA con su sede en Riobamba lleva un registro de los análisis de agua reportados por los laboratorios de los GAD municipales de la provincia de Chimborazo, estos reportes se los realiza de forma impresa y no se entrega a tiempo los reportes a las oficinas del departamento de agua potable y saneamiento, además no se cuenta con una herramienta que proporcione información específica que ayude a la toma de decisiones cuando los sistemas de agua potable no cumplen con todos los requerimientos de calidad del agua.

Por consiguiente se ve la necesidad de automatizar el proceso de entrega de reportes de análisis de agua para consumo doméstico mediante la implementación

de una aplicación que esté al alcance de los técnicos de los laboratorios, se considera que debe ser una aplicación web.

La aplicación web debe ser administrada por la SENAGUA y además la información debe ser registrada por los técnicos de municipios de la provincia, como petición del departamento de agua potable y saneamiento la información también se pretende publicarla de manera pública y abierta para todos los ciudadanos del territorio nacional.

#### 4.3. PLANIFICACIÓN DEL PROYECTO

Para poder estimar los tiempos que tomará cumplir con cada una de las etapas del desarrollo del sistema con el uso de la metodología IWeb citada en Pressman (2010), se resumen las actividades más relevantes en la siguiente tabla:

ID	ACTIVIDADES	Número de semanas
1	Requisitos	1
2	Ingeniería	
2.1	Diseño arquitectónico	2
2.2	Diseño de navegación	2
2.3	Diseño de interfaz	3
3	Generación de páginas y pruebas	20
4	Evaluación del cliente	4
Total de semanas		32

*Tabla 34. Resumen de actividades  
Elaborado por: Juan Carlos Moyota*

Las duración de cada una de las cuatro etapas del proceso de ingeniería web IWeb se muestran en la tabla 34 no se ha contemplado las etapas de formulación del problema y planificación. En el **literal 10.2** se describe detalladamente con un diagrama de Gantt.

#### 4.4. REQUISITOS DEL SISTEMA

Para establecer los requisitos iniciales para la producción del sistema web para manejo de calidad del agua para consumo doméstico de la provincia de Chimborazo se realiza una especificación de requisitos IEEE 830.

#### **4.4.1. INTRODUCCIÓN**

“La Central Ecuatoriana de Servicios Agrícolas” en adelante CESA, es una fundación privada sin fines de lucro reconocida por el Estado Ecuatoriano mediante acuerdo ministerial N° 09025 del 19 de mayo de 1967, expedida por el entonces Ministerio de Provisión Social y Trabajo, que trabaja conjuntamente con la población rural y gobiernos locales para facilitar procesos sostenibles de Desarrollo Humano. CESA apoya la gestión de los sistemas de producción, las iniciativas de comercialización asociativa de productos campesinos, el manejo técnico y social del agua, el manejo de los recursos naturales y la apropiación de estos procesos por parte de la población.

“Agrónomos y Veterinarios Sin Fronteras”, en adelante AVSF, es una organización de solidaridad internacional Francesa cuya misión es trabajar para el desarrollo rural de los campesinos en las zonas marginales y contribuir a las acciones afectan el Norte y el Sur de estos agricultores, a través de la movilización de competencias específicas en el ámbito de la agricultura, la ganadería y la gestión de recursos naturales. La cooperación AVSF centrado en el reconocimiento de las organizaciones de agricultores en la sociedad. AVSF implementa y participa en programas de desarrollo rural en 19 países, entre ellos varios países de América latina y el Ecuador en particular.

La Secretaria del Agua realiza un análisis constante de la calidad del agua para el consumo doméstico, esta información se registra en informes impresos en papel por lo cual se ve la necesidad de almacenar dicha información en digital y que mejor manera que en una base de datos para acceso de técnicos que deseen conocer información de la calidad del agua que consumen a diario.

Este documento presenta el conjunto de características necesarias de especificaciones de requisitos para el análisis, diseño, programación e implementación del sistema web para control de calidad del agua para consumo humano de los 10 cantones de la provincia de Chimborazo. Así mismo, presenta el formato de Especificación de Requisitos Software según la versión de 1998 del estándar IEEE 830.

#### **4.4.1.1. PROPÓSITO**

El propósito de este documento es especificar los requerimientos de software (ERS) para cumplir el fin del proyecto el cual es automatizar el proceso de registro de fichas de control de calidad del agua para consumo humano que realiza la SENAGUA con el apoyo del Comité de Gestión de la subcuenca del Chambo, con estos datos se pretende realizar consultas de datos de los sistemas de agua de la provincia que difieran a los límites permisibles, tanto físico – químico, características químicas y análisis bacteriológico, bajo esta premisa servirá como base para la fase de diseño y desarrollo del sistema.

El sistema web para gestión de datos de calidad del agua para consumo humano está dirigido a los técnicos de los municipios de los 10 cantones de la provincia de Chimborazo, dichos técnicos se consideran usuarios que manipularán el sistema, además va dirigida a todas las personas que necesiten alguna información de la calidad del agua que consumen.

#### **4.4.1.2. ÁMBITO DEL SISTEMA**

Para el futuro el sistema tendrá el nombre de “**CALIAGUA**”.

El sistema a desarrollar permitirá a la SENAGUA la toma de decisiones en base a los resultados que se obtengan de las consultas, además de llevar la información digitalmente se reduce la presencia de papeles u hojas y un respaldo histórico de la información generada.

El sistema podrá ser accesible desde cualquier parte del país por tratarse de un sistema web, siempre y cuando se tenga acceso a internet.

El sistema estará estructurado en grupos que se especifican de la siguiente manera:

- ❖ **Módulos de gestión de barrios o comunidades**, donde se podrá registrar y actualizar datos de las comunidades o barrios de la provincia.
- ❖ **Módulo de gestión de sistemas de agua**, este módulo del sistema se refiere específicamente al registro y actualización de los sistemas de agua

potable o entubada de los cuales posteriormente se registrarán los resultados de las muestras.

- ❖ **Módulo de gestión de muestras**, este será la parte esencial del sistema ya que aquí se podrá registrar y actualizar los resultados de la muestra de los sistemas de agua, dicha información alimentará a la base de datos para poder entregar estadísticas en gráficos fáciles de comprender para cualquier persona que sepa del tema de agua.
- ❖ **Módulo de reportes y búsquedas**, en este módulo existen las opciones de búsquedas de datos de las muestras tomadas en determinado sistema. Además desde este módulo se podrá exportar a PDF el reporte o la ficha de determinado sistema.
- ❖ **Módulo de gestión de usuarios**: en este módulo se podrán registrar, eliminar o actualizar información vinculada a usuarios y únicamente estará abierto para el técnico de la SENAGUA quien administrará los usuarios.
- ❖ **Inicio de sesión para usuarios registrados**, este módulo será el acceso para dos tipos de usuarios: “registrados” los cuáles serán los encargados de los municipios que ingresarán la información. Además otro usuario que tendrá acceso será el “administrador” del sistema web.

El objetivo principal del diseño y programación del sistema web “**CALIAGUA**” es llevar la información en una base de datos relacional que ayude a la obtención y recuperación de datos de las muestras tomadas en diferentes partes de los sistemas de agua de la provincia y además la generación de reportes estadísticos con la información alimentada por los técnicos de los municipios de la provincia.

#### **4.4.1.3. PERSONAL INVOLUCRADO**

<b>Nombre</b>	Marlene Barba
<b>Rol</b>	Coordinadora del proyecto
<b>Categoría profesional</b>	Ingeniera
<b>Responsabilidades</b>	Responsable CESA
<b>Información de contacto</b>	0999710893
<b>Aprobación</b>	

<b>Nombre</b>	José Luis Coba
<b>Rol</b>	Técnico de CESA/AVSF
<b>Categoría profesional</b>	Ingeniero
<b>Responsabilidades</b>	Co - responsable
<b>Información de contacto</b>	0984748509
<b>Aprobación</b>	

<b>Nombre</b>	Aníbal Armijos
<b>Rol</b>	Técnico del departamento de agua potable y saneamiento.
<b>Categoría profesional</b>	Ingeniero
<b>Responsabilidades</b>	Técnico responsable de la SENAGUA.
<b>Información de contacto</b>	0991473842
<b>Aprobación</b>	

<b>Nombre</b>	Anita Congacha
<b>Rol</b>	Tutora de proyecto
<b>Categoría profesional</b>	Ingeniera
<b>Responsabilidades</b>	Tutorías para el desarrollo del sistema.
<b>Información de contacto</b>	0994320667
<b>Aprobación</b>	

#### **4.4.1.4. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS**

<b>ACRÓNIMO</b>	<b>DEFINICIÓN</b>
CALIAGUA	CALIDAD DEL AGUA, sistema web para la gestión de datos de la calidad del agua de consumo humano de la subcuenca del Chambo.
ERS	Especificación de requisitos de software
IEEE	Institute of Electrical & Electronics Engineers
Base de datos.	Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
PK	Clave principal, es el identificador de la tabla de la base de datos.
SENAGUA	Secretaria del agua
CESA	Central Ecuatoriana de Servicios Agrícolas
AVSF	Agrónomos y Veterinarios Sin Fronteras

#### **4.4.1.5. REFERENCIAS**

El sistema tendrá como referencia el documento de Especificación de Requisitos de Software ERS IEEE 830 como estándar para la obtención de requisitos de sistema.

#### **4.4.1.6. VISIÓN GENERAL DEL DOCUMENTO**

Este documento especifica los requisitos específicos para el desarrollo del software en su proceso de construcción. La organización del documento se describe brevemente de la siguiente forma:

En la parte de descripción general se explicará de manera sencilla las principales tareas que el sistema realizará como son el contenido de: Módulo de reportes y búsquedas, módulo de generación de gráficas estadísticas, módulo de gestión de muestras, módulos de gestión de barrios o comunidades y módulo de inicio de sesión sistema para usuarios registrados que tendrán acceso al sistema.

En la parte de descripción específica se detalla los datos de cada tabla de la Base de Datos del sistema que se utilizará para la utilización del sistema así como las PK de las tablas las cuales deben ser únicas de cada registro.

#### **4.4.2. DESCRIPCIÓN GENERAL**

##### **4.4.2.1. PERSPECTIVA DEL PRODUCTO**

El Sistema de Manejo de Datos de Calidad del Agua de Consumo Chimborazo es un sistema independiente de otros sistemas que tiene la SENAGUA por tanto es el único sistema que registrará información de control de calidad del agua para consumo a nivel nacional.

##### **4.4.2.2. FUNCIONES DEL PRODUCTO**

El Sistema de Control de Calidad del Agua Chimborazo será un sistema web eficaz que tendrá las siguientes características y tareas:

- ✓ El acceso a la administración de datos será sólo para personal autorizado, estos usuarios serán los técnicos de los municipios de cada cantón.
- ✓ Será desarrollado en un lenguaje de programación libre para evitar costos en la adquisición de licencias de uso.
- ✓ Interfaz gráfica de usuario agradable para el usuario.
- ✓ Administrará datos de los barrios, sistemas y usuarios estos módulos se podrán realizar tareas como: registrar y cambiar información de los registros.
- ✓ Ejecutará la tarea primordial del sistema que es el almacenamiento y posterior recuperación de los datos de las muestras de los sistemas de agua para consumo, se permitirá la impresión de un reporte que se asemejará a la ficha que actualmente se usa.
- ✓ El módulo de administración de usuarios será una tarea específica del técnico de la SENAGUA.
- ✓ Generación de estadísticas con los datos que se almacena en la base de datos ya sean en gráficos y/o datos.



#### **4.4.2.3. CARACTERÍSTICAS DE LOS USUARIOS**

A continuación se describen las tareas de los diferentes niveles de acceso para usuarios:

- El sistema tendrá dos partes: la pública que será libre para todos ya que es un sistema web, pero no podrán administrar datos ya sea de las muestras o los barrios y la parte privada reservado para los usuarios registrados en el sistema.
- Los usuarios registrados tendrán acceso a administrar datos de las muestras y los barrios, estos usuarios son los encargados de los municipios de la provincia.
- El técnico de la SENAGUA será quien administre los usuarios ya que este asignará usuarios y claves a cada municipio.

#### **4.4.2.4. RESTRICCIONES**

Limitación en el hardware:

- ❖ Una impresora para la impresión de la ficha.

Limitación en la información:

- ❖ El programador se encargará de ingresar la información de los cantones y parroquias.
- ❖ El resto de información como registro de los resultados de las muestras, barrios y sistemas de agua potable será ingresada por los municipios.

Limitación del sistema operativo:

- ❖ El sistema funcionará en las plataformas Windows, Linux y MacOS.

#### **4.4.2.5. SUPOSICIONES Y DEPENDENCIAS**

- ❖ Los usuarios autorizados por la SENAGUA sólo podrán acceder con su nombre de usuario y clave únicos.

- ❖ Las interfaces del diseño pueden o no estar acorde con los gustos de los usuarios es decir gusto de colores, iconos, acceso al sistema entre otros puntos.
- ❖ Si el modelado de la base de datos para el sistema no es confiable se prestará a cambios durante el proceso de desarrollo

#### **4.4.2.6. REQUISITOS A FUTURO**

Si la Secretaria del agua considera el sistema muy útil para sus tareas se considerará ampliar el sistema para el resto de provincias del país.

#### **4.4.3. REQUISITOS ESPECÍFICOS**

##### **4.4.3.1. INTERFACES EXTERNAS**

**Interfaz de usuario:** el sistema contará con una interfaz totalmente gráfica, controles de fácil manejo para el usuario, un sistema de navegación sencilla y experiencia interactiva que no requiera recargar la página cada vez.

**Interfaz de hardware:** para la utilización del sistema se utilizarán equipos que tengan una conexión a internet con una tarjeta de red o conexión inalámbrica que es un requisito importante de hardware.

**Interfaz de software base:** entre el recurso de software a utilizarse para la programación del sistema web podemos mencionar:

- Sistema operativo: Windows 8 de 64 bits
- Software base: PHP 5.3
- Sistema gestor de base de datos: MySQL 5
- Servidor local de prueba: Apache 2

Cabe mencionar que en el software XAMPP podemos encontrar integrado los paquetes: Apache, PHP y MySQL, se utilizará este paquete para la producción del sistema web.

#### 4.4.3.2. FUNCIONES

El futuro sistema se detalla lo siguiente en sus módulos y funciones:

**Inicio de sesión:** para acceder al sistema y realizar tareas de administración de datos de barrios, sistemas y muestras el usuario contará con datos únicos para el acceso al sistema. Estos datos serán asignados por la SENAGUA para evitar que usuarios ajenos se registren en el sistema y puedan manipular los datos.

**Módulo de administración de usuarios:** el administrador del sistema podrá administrar datos de los usuarios, refiriéndonos a administrar datos al registro, eliminación y actualización. Los datos de cada usuario deben ser los siguientes:

- ✓ Código de usuario
- ✓ Nombre de usuario
- ✓ Contraseña
- ✓ Cantón al que pertenecen
- ✓ Perfil de usuario
- ✓ Correo electrónico

**Operaciones como usuario visitante:** el usuario visitante del sistema podrá únicamente realizar consultas y ver gráficos estadísticos con los datos registrados en las tablas de muestras.

**Operaciones como usuario registrado:** el usuario registrado una vez que se verifica sus datos tendrá acceso a las tareas de administrar (actualizar y añadir) los datos de barrios, sistemas y muestras, además de las operaciones que hace el usuario visitante. Un detalle que se omite es que este usuario no podrá cambiar o actualizar los datos del módulo muestras.

**Operaciones como usuario “administrador”:** el usuario considerado como administrador será quien realice las tareas tanto como usuario visitante y usuario registrado, pero además se encarga de administrar (registrar, eliminar y actualizar) usuarios y cambiar las claves de los usuarios. Este usuario será el encargado de la SENAGUA.

**Módulo de consulta de fichas:** para las tareas de este módulo se realizará consultas de fichas, además de la posterior exportación a formato PDF e impresión de la ficha para cualquier tipo de usuario.

- a) **Consulta por sistema:** En el diagrama se muestran los pasos para buscar las fichas de un determinado sistema de agua potable:



*Figura 31. Secuencia para realizar consulta de fichas  
Elaborado por: Juan Carlos Moyota*

El usuario al seleccionar los datos en el orden de la figura anterior el sistema buscará los registros correspondientes al sistema seleccionado mostrándose organizados en una tabla, con la opción de seleccionar un registro para mostrar toda la información y exportar a formato PDF para su impresión.

- b) **Consulta de fichas de manera masiva:** esta consulta de fichas tiene como finalidad mostrar todos los registros de muestras de un cantón, el usuario solo deberá seleccionar el cantón y se mostrará una tabla con todos los registros y de la misma forma que la búsqueda por sistema se podrá ver toda la ficha y exportar a formato PDF.

**Módulo de administración de datos de barrios:** el usuario autorizado para administrar datos realizará las tareas de:

- Registro: ingresar datos de barrios.
- Modificación: cambiar algún dato de los barrios o comunidades.

Los datos de los barrios o comunidad son los siguientes:

- ✓ Código de barrio

- ✓ Cantón
- ✓ Nombre del barrio

**Módulo de administración de sistemas de agua:** este módulo registra, elimina y actualiza datos de los sistemas a agua de la provincia. Los datos de los sistemas son:

- ✓ Código del sistema
- ✓ Barrio
- ✓ Nombre del sistema de agua
- ✓ Tipo de sistema

**Módulo de administración de muestras:** este módulo registra y actualiza datos de las muestras. Los datos de la muestra deben ser los siguientes:

- ✓ Datos principales:
  - Institución
  - Departamento
  - Técnico responsable de recolección
  - Nombre del cantón
  - Nombre de la parroquia
  - Nombre del barrio o comunidad
  - Nombre del sistema de agua
  - Sitio exacto de toma de la muestra
  - Fecha y hora de recolección
  - Fecha y hora de análisis
- ✓ Parámetros de calidad del agua potable establecidos por la norma **NTE INEN 1108 Quinta revisión 2014** y de la norma **TULAS Libro VI, Anexo I, Tablas 1 y 2:**

- PH
- Color
- Olor
- Turbiedad
- Temperatura
- Sólidos totales disueltos
- Conductividad
- Dureza
- Cloro libre residual
- Hierro
- Nitratos
- Nitritos
- Sulfatos
- Fosfatos
- Manganeseo
- Fluoruros
- Amoniacó
- Coliformes totales
- Coliformes fecales

**Módulo de gráficos estadísticos:** Este módulo permitirá ver gráficas de: porcentaje de tipos de sistemas en un cantón, evolución de un parámetro de calidad del agua para consumo doméstico de un sistema en particular, además una lista de barrios o comunidades que cumplen los límites permisibles del agua para consumo doméstico, un listado de barrios que no cumplan dichos límites permisibles y sistemas sin fichas registradas.

**a) Porcentaje de tipo de sistemas por cantón**

Esta gráfica muestra el porcentaje de tipos de sistemas que tiene un cantón o parroquia, el usuario debe seleccionar el nombre del cantón o parroquia según sea el caso y podrá ver una gráfica de pastel con el porcentaje de tipos de sistemas.

## b) Parámetros de calidad del agua

El usuario selecciona un sistema determinado y un parámetro que puede ser:

- PH
- Color
- Olor
- Turbiedad
- Temperatura
- Sólidos totales disueltos
- Conductividad
- Dureza
- Cloro libre residual
- Hierro
- Nitratos
- Nitritos
- Sulfatos
- Fosfatos
- Manganeseo
- Fluoruros
- Amoniacó
- Coliformes totales
- Coliformes fecales

Con los datos seleccionados previamente el usuario generará una gráfica de líneas con los resultados que se encuentran en la base de datos y las fechas de recolección de las muestras.

## c) Calidad del agua

El sistema realizará una búsqueda de los sistemas que tienen una buena calidad del agua, mala calidad del agua y sistemas que no tienen datos registrados, para obtener esta información el usuario deberá seleccionar el cantón. Los sistemas que no cumplen los límites permisibles establecidos en la norma **NTE INEN 1108 Quinta revisión 2014** y de la norma **TULAS Libro VI, Anexo I, Tablas 1 y 2** se mostrarán los parámetros en color rojo esto con los últimos resultados que se encuentran en la base de datos.

**Control de datos:** el sistema será capaz de controlar la información validándola tanto números como letras para un correcto registro de los datos en todos los módulos de ingreso y actualización de datos.

#### **4.4.3.3. REQUISITOS DE RENDIMIENTO**

- ✓ Impresión de documentos, antes de imprimir documentos se mostrará un PDF del documento que se va a imprimir extraídos del sistema.
- ✓ Mensajes de error al ingresar datos incorrectos.
- ✓ Mensajes de error cuando existan algún formato de datos que sea inválido antes de registrar en la base datos.
- ✓ Consultas por parámetros establecidos definidos.

#### **4.4.3.4. RESTRICCIONES DE DISEÑO**

Para la creación de la base de datos se utilizará el software:

 MySQL 5.0.51a

Para el diseño, programación de la aplicación así como la conexión a la base de Datos, pantallas entre otras se utilizará los lenguajes de programación para la web:

- ✓ PHP 5.3.8
- ✓ HTML
- ✓ JavaScript
- ✓ CSS

Como editor de código se usará:

- ✓ NetBeans IDE 7.4
- ✓ Adobe Dreamweaver CS6

Para la edición y diseño de imágenes y logos se utiliza como herramienta:

- ✓ Adobe Photoshop CS6.
- ✓ GIMP 2

#### **4.4.3.5. ATRIBUTOS**

**Seguridad:** En cuanto se refiere la seguridad y acceso a los módulos se dijo con anterioridad que los usuarios registrados deberán ingresar con su **Login** y



**Password,** para tener un mayor nivel de seguridad en el acceso al sistema se utilizará el algoritmo de encriptación más seguro para la encriptación de claves.

**Portabilidad:** el sistema deberá ser instalable en cualquier sistema operativo Windows o Linux que tenga características de software iguales o superiores a las especificadas en las restricciones de diseño donde se especifica el software que se utilizará para la programación.

**Mensajes de error:** se mostrarán mensajes de error cuando el usuario registrado ingrese datos incorrectos como por ejemplo:

- ✚ El usuario ingrese letras o caracteres en campos que sólo se deben ingresar números.
- ✚ El usuario ingrese números en campos que debe ingresar letras.
- ✚ El usuario ingrese caracteres incorrectos en campos como fecha y hora.

**Impresión de documentos:** la impresión de las fichas con información de las muestras será una tarea de cualquier usuario ya que la información es de dominio público.

## **4.5. INGENIERÍA**

El siguiente paso del proceso de ingeniería web es la fase de ingeniería donde se realiza una descripción del diseño arquitectónico, navegación e interfaz.

### **4.5.1. DISEÑO ARQUITECTÓNICO**

#### **4.5.1.1. DISEÑO DE BASE DE DATOS**

Con los datos obtenidos en la especificación de requisitos se plantea el siguiente diseño de base de datos relacional:

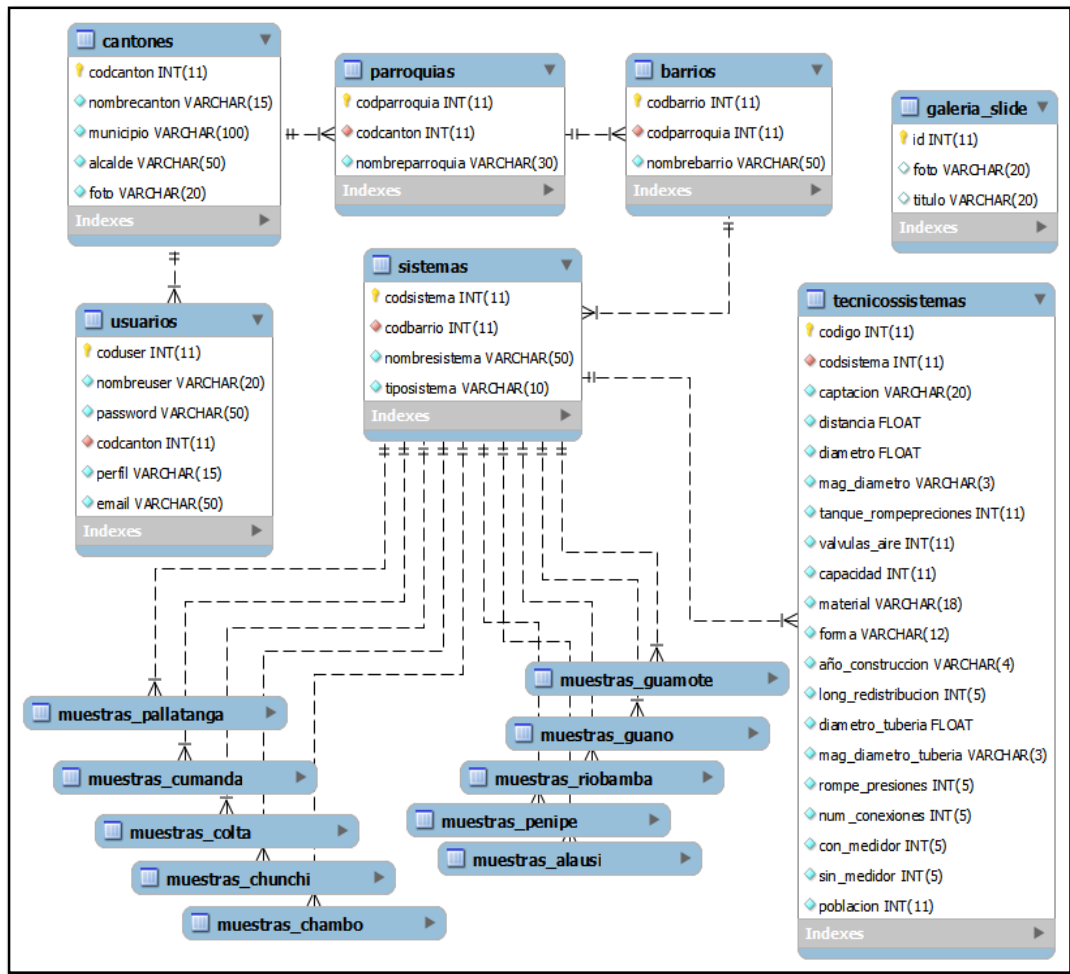


Figura 32. Diseño de la Base de Datos  
Elaborado por: Juan Carlos Moyota

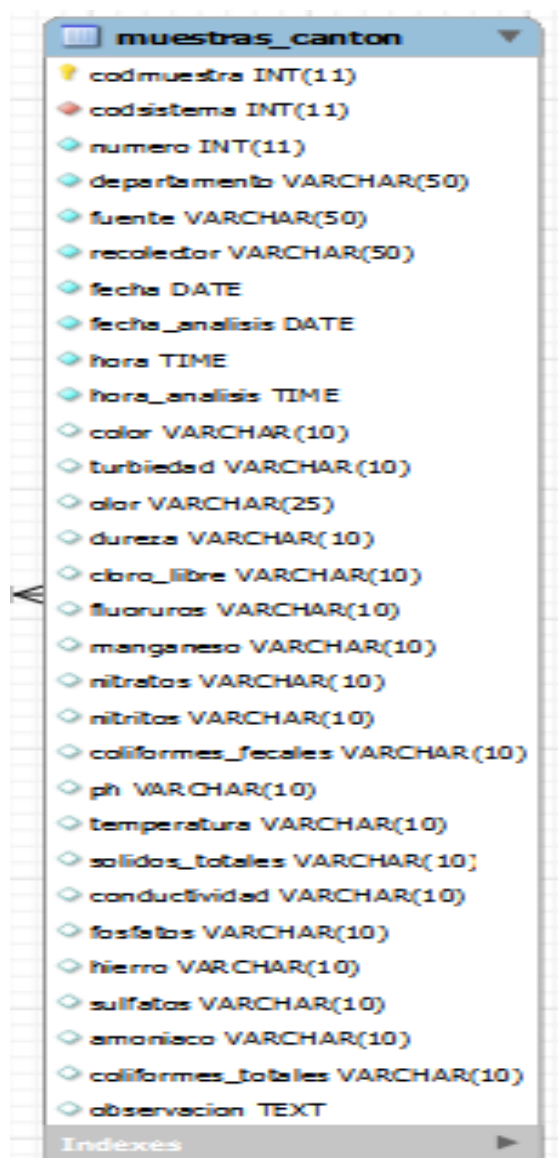


Figura 33. Diseño de Base de Datos de la tabla "muestras"  
Elaborado por: Juan Carlos Moyota

El esquema de la base de datos planteado en el grafico 33 muestra un total de 17 tablas las cuales se describen en el *Manual Técnico*.

#### 4.5.1.2. DEFINICIÓN CASOS DE USO DEL SISTEMA

En la **Figura 34** podemos tener una expectativa general de las funciones específicas del sistema web en desarrollo, a continuación el siguiente diagrama de casos de uso del sistema web en producción.

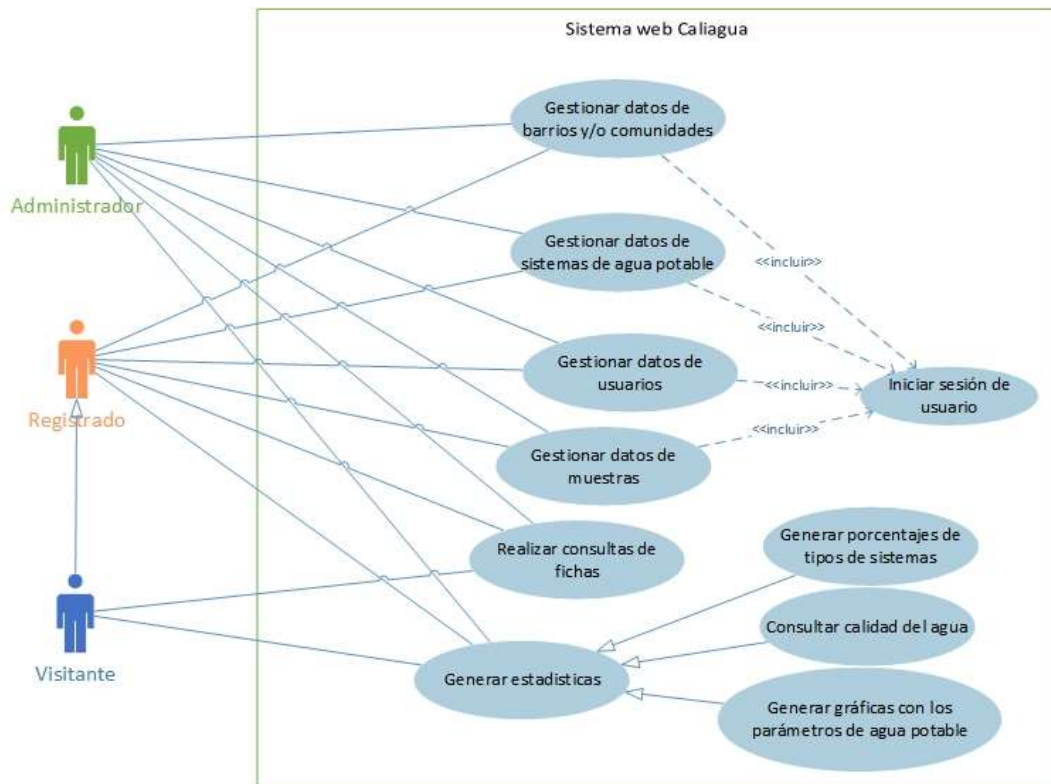


Figura 34. Diagrama general de casos de uso  
Elaborado por: Juan Carlos Moyota

#### 4.5.1.2.1. Casos de uso “Iniciar sesión”

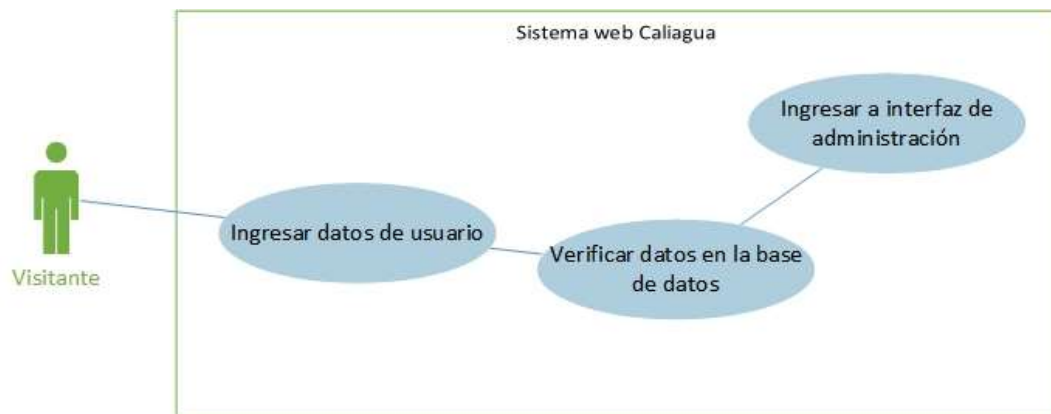


Figura 35. Caso de uso "Iniciar sesión"  
Elaborado por: Juan Carlos Moyota

**4.5.1.2.2. Casos de uso “Gestionar datos de barrios y/o comunidades”**



Figura 36. Caso de uso - Gestionar barrios y/o comunidades  
Elaborado por: Juan Carlos Moyota

**4.5.1.2.3. Casos de uso “Gestionar datos de sistemas de agua potable”**



Figura 37. Casos de usos - Gestionar de sistemas de agua  
Elaborado por: Juan Carlos Moyota

**4.5.1.2.4. Casos de uso “Gestionar datos de muestras”**



Figura 38. Casos de uso - Gestionar datos de muestras  
Elaborado por: Juan Carlos Moyota

#### 4.5.1.2.5. Casos de uso “Gestionar datos de usuarios”

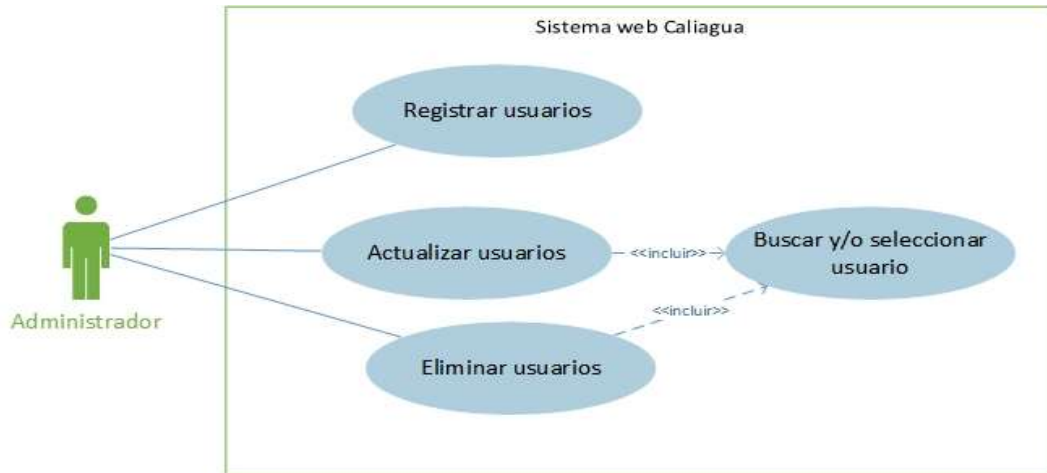


Figura 39. Casos de uso - Gestionar datos de usuarios  
Elaborado por: Juan Carlos Moyota

#### 4.5.1.2.6. Casos de uso “Realizar consultas de fichas”

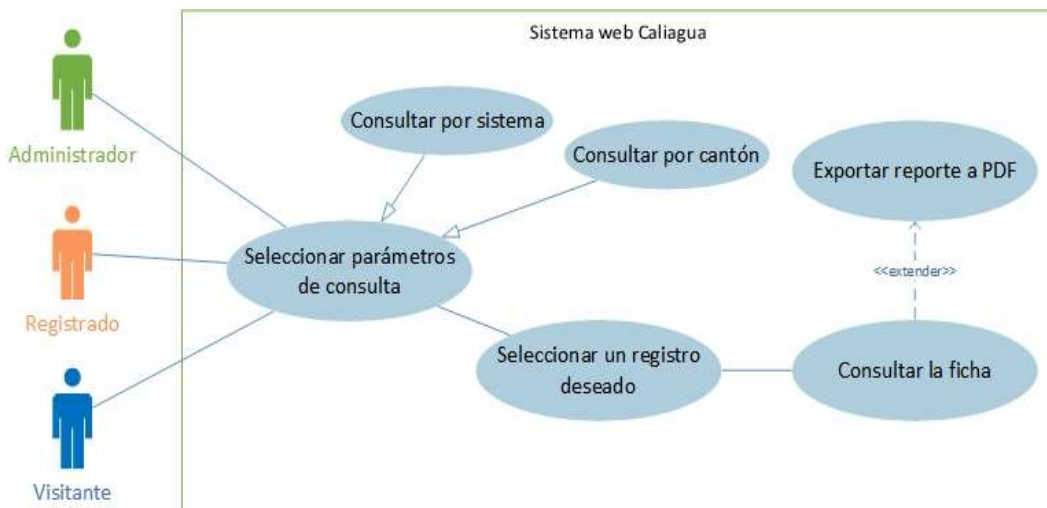


Figura 40. Casos de uso - realizar consultas de fichas  
Elaborado por: Juan Carlos Moyota

**4.5.1.2.7. Casos de uso “Generar porcentajes de tipos de sistemas”**



Figura 41. Casos de uso - Porcentajes de tipos de sistemas  
Elaborado por: Juan Carlos Moyota

**4.5.1.2.8. Casos de uso “Generar gráficas con los parámetros de agua potable”**



Figura 42. Casos de uso - Gráficas con los parámetros de agua potable  
Elaborado por: Juan Carlos Moyota



#### 4.5.1.2.9. Casos de uso “Consultar calidad del agua”



Figura 43. Casos de uso - Consultar calidad del agua  
Elaborado por: Juan Carlos Moyota

En el *Manual técnico* se presenta una descripción completa de cada caso de uso del sistema web.

#### 4.5.1.3. DISEÑO DE RED

El siguiente gráfico presenta el diagrama de red institucional de la UNACH ya que la aplicación y la base de datos se alojarán en un servidor de su propiedad.

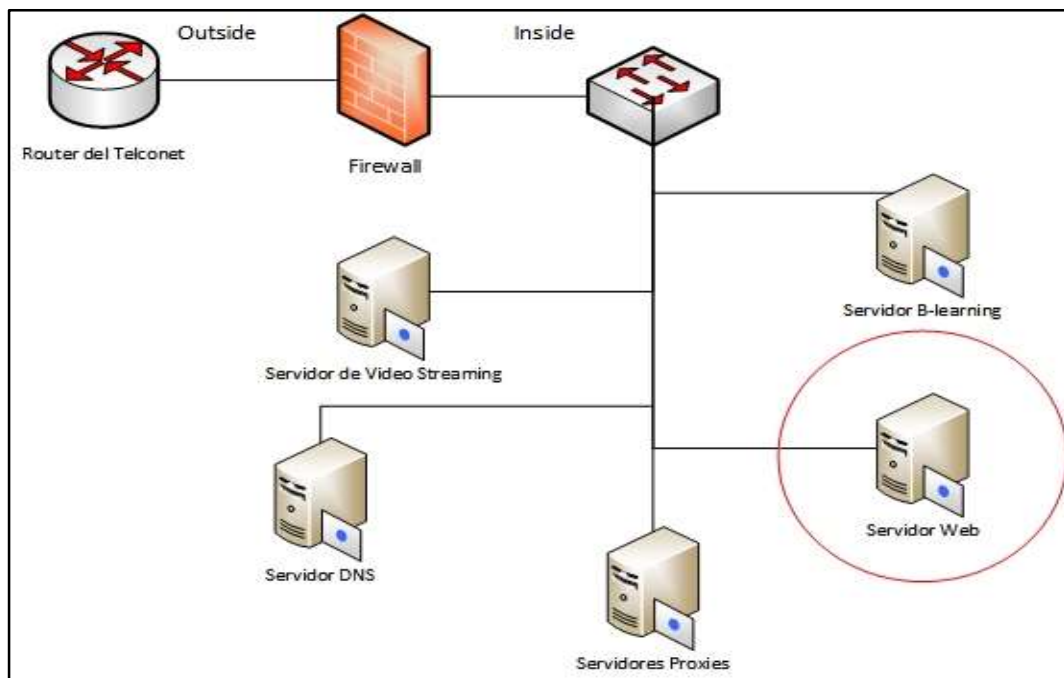


Figura 44. Diagrama de red de la UNACH  
Fuente: Centro de cómputo de la UNACH



#### 4.5.2. DISEÑO DE NAVEGACIÓN

En el siguiente gráfico se presenta el diseño de navegación de la página principal del sistema la cual tendrá acceso para cualquier usuario ya sea este administrador, usuario registrado o un visitante que acceda al sitio web.

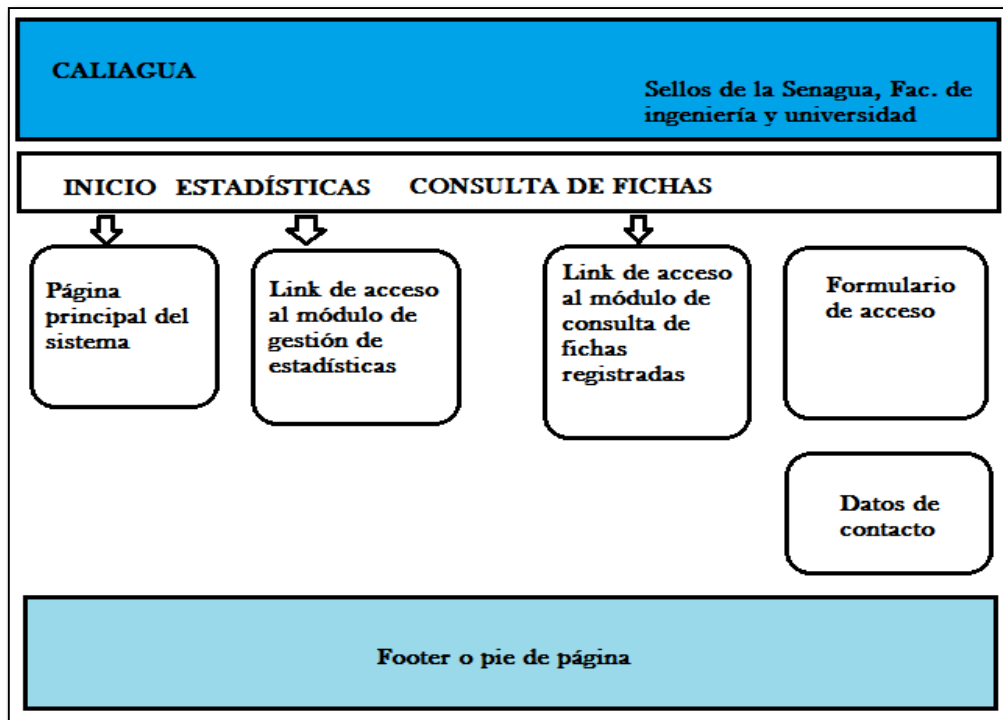


Figura 45. Diagrama de navegación - Página inicial  
Elaborado por: Juan Carlos Moyota

En las **Figuras 46 y 47** se muestra el diseño de navegación para los usuarios administrador y registrado. Los usuarios con el perfil de administrador tienen acceso a más opciones de menú que un usuario registrado ya que estos se encargan de registrar la información en la base de datos y con esta información los usuarios visitantes y los usuarios del sistema podrán generar gráficos estadísticos con los datos registrados en la base de datos.

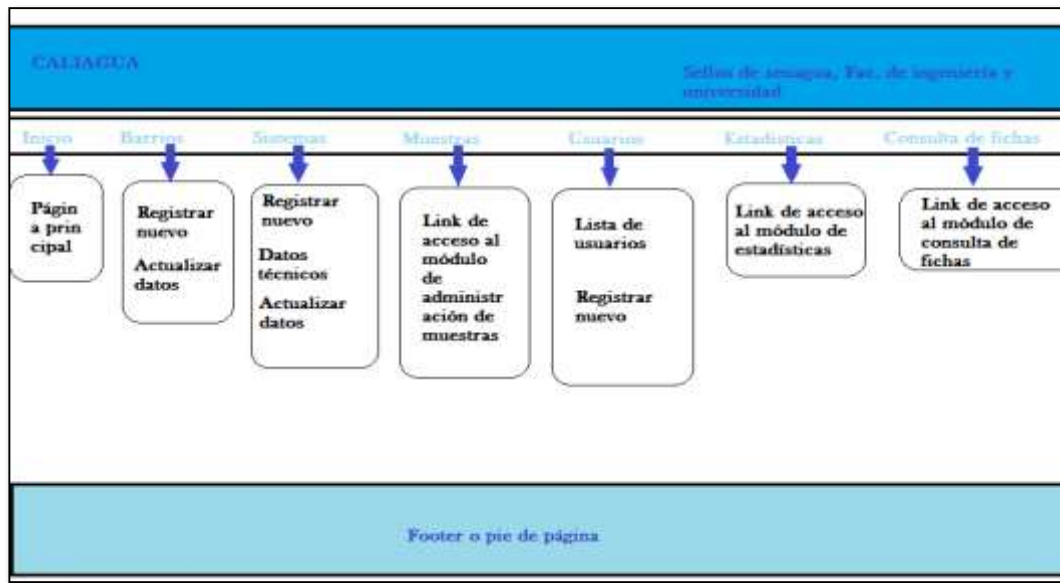


Figura 46. Diseño de navegación - Página de administración  
Elaborado por: Juan Carlos Moyota

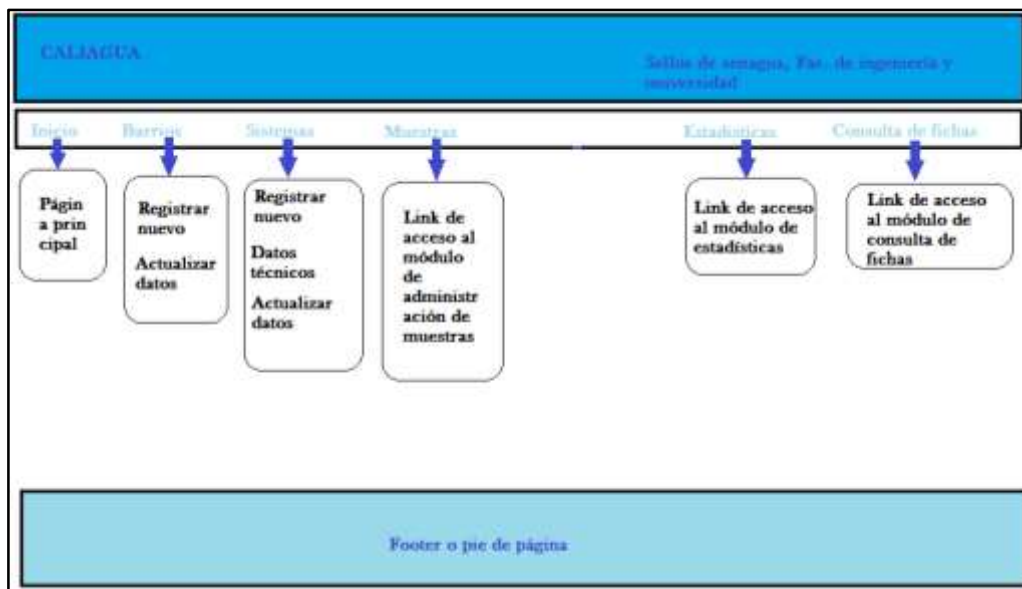


Figura 47. Diseño de navegación - Página de registro  
Elaborado por: Juan Carlos Moyota

Las Figuras 48 y 49 presentan el diseño de navegación del *módulo de administración de muestras* de los dos tipos de usuarios que pueden realizar tareas administrativas con los datos.

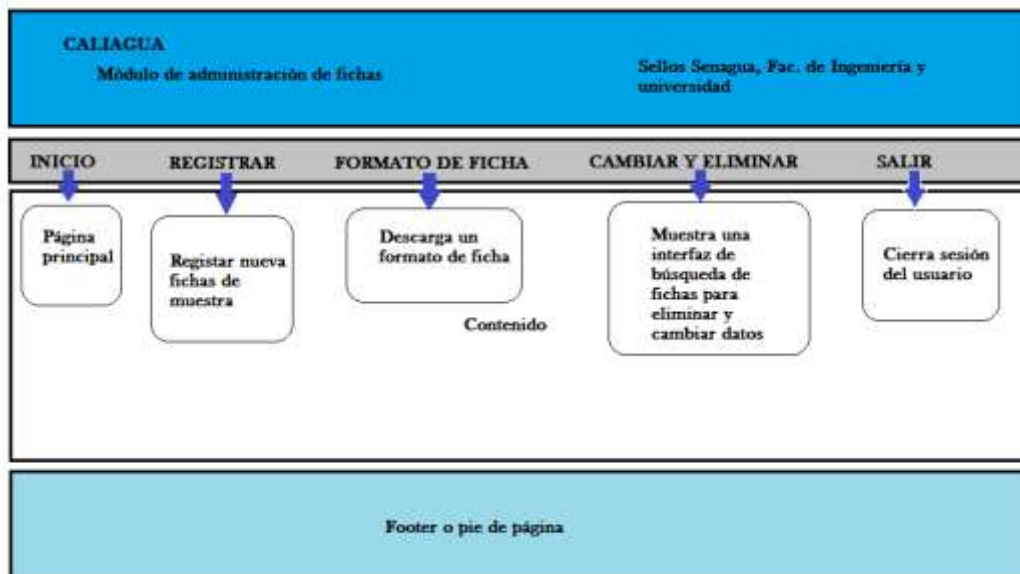


Figura 48. Diseño de navegación - Módulo de fichas – Administrador  
Elaborado por: Juan Carlos Moyota

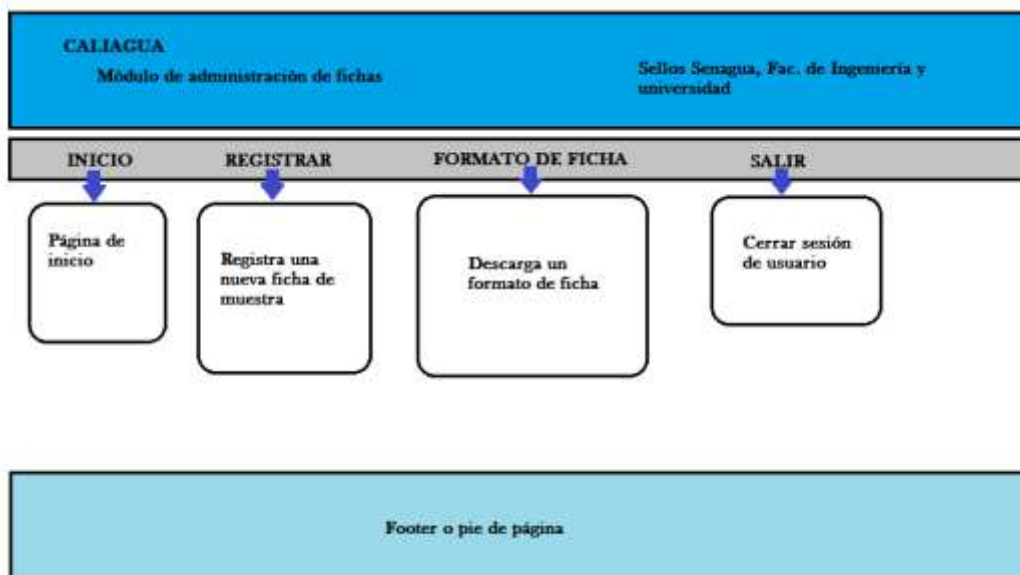


Figura 49. Diseño de navegación - Módulo de fichas – Registrado  
Elaborado por: Juan Carlos Moyota

La **Figura 50** muestra el diseño de navegación para la página de *consulta de fichas* por cantón y sistema considerada como una parte de dominio público para cualquier persona pueda consultar las fichas de análisis de agua potable registradas en la base de datos del sistema web.

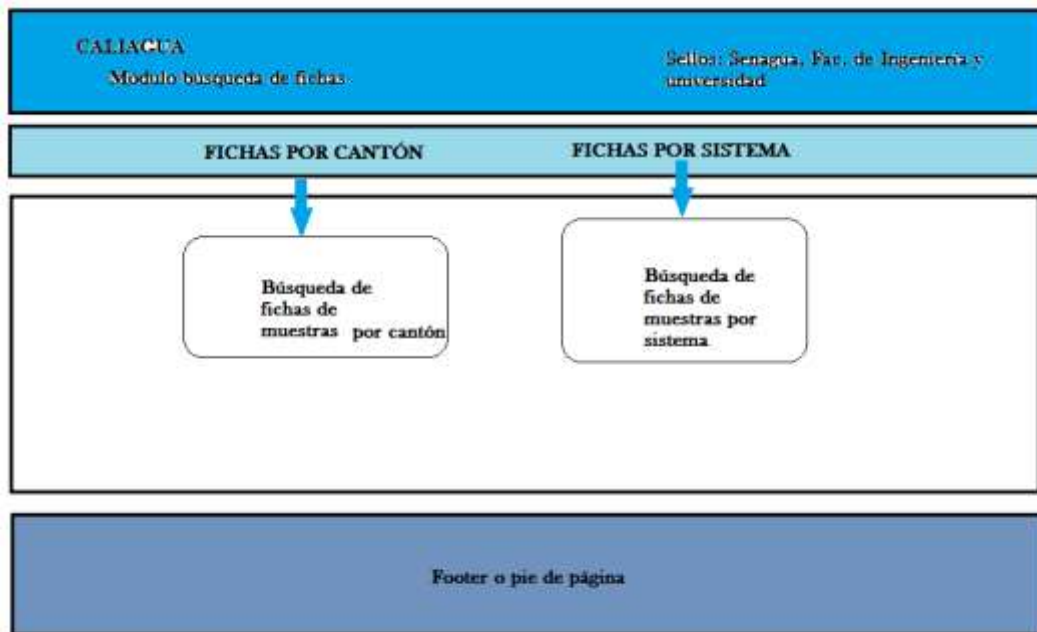


Figura 50. Diseño de navegación - Módulo de consulta de fichas  
Elaborado por: Juan Carlos Moyota

La **Figura 51** muestra el diseño de navegación del módulo de generación de estadísticas.

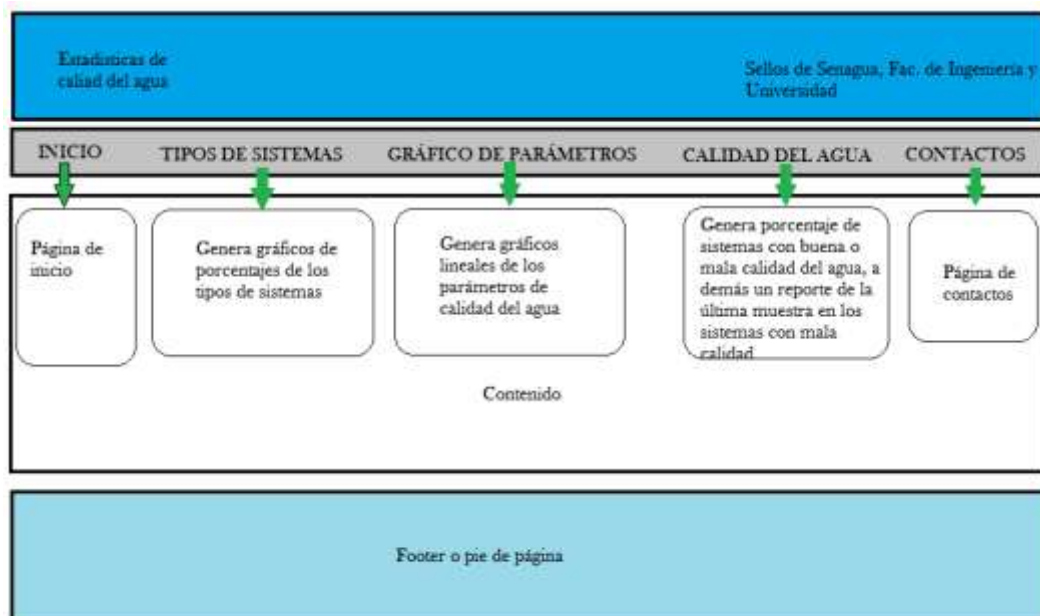


Figura 51. Diseño de navegación - Módulo de estadísticas  
Elaborado por: Juan Carlos Moyota

### 4.5.3. DISEÑO DE INTERFAZ

A continuación se describe la fase de diseño de interfaz de usuario de todo el sistema web. Estos diseños son los últimos que se implementaron en la aplicación web para la gestión de datos de calidad del agua de consumo doméstico de la provincia de Chimborazo.

#### 4.5.3.1. INTERFAZ DE PÁGINA PRINCIPAL

La interfaz principal tendrá una vista de usuario como se muestra en la siguiente figura:

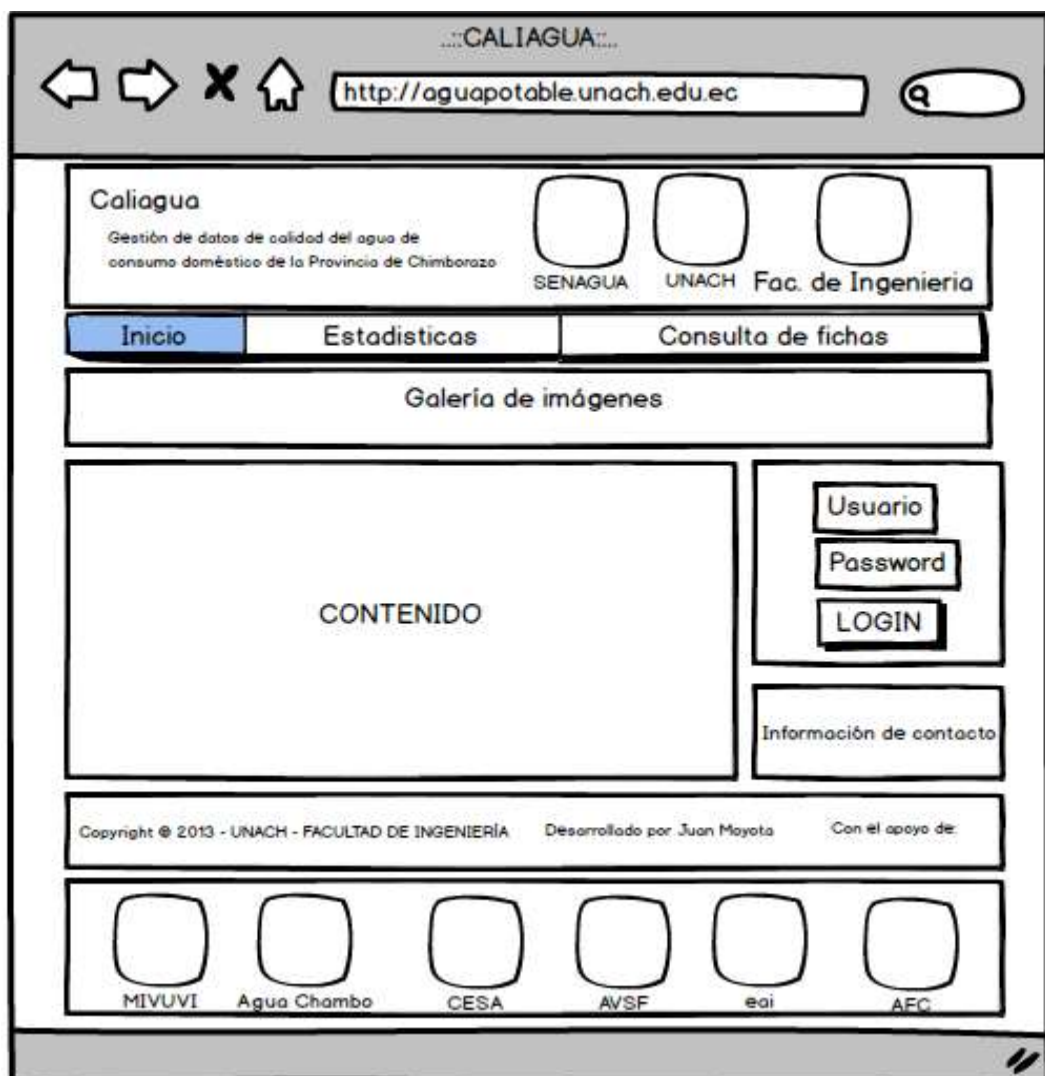


Figura 52. Interfaz de usuario - Página principal  
Elaborado por: Juan Carlos Moyota

La página principal del sistema constará de:

- ❖ Cabecera
- ❖ Menú
- ❖ Galería de fotos
- ❖ Cuerpo o contenido
- ❖ Pie de página (copyright, derechos de autor y auspiciantes)

**Cabecera:** la cabecera contendrá una marquesina y una imagen de cabecera con los logos de las instituciones y organizaciones que ayudaron a la ejecución del proyecto, además estará el nombre de la aplicación con el título CALIAGUA y una pequeña descripción del sistema.

**Menú:** las opciones de menú de primera vista serán tres: inicio, consulta de fichas y estadísticas.

**Galería de imágenes:** es una transición de imágenes para darle dinamismo al sistema web y no sea tan estático.

**Cuerpo o contenido:** estará estructurado por dos partes, los módulos de acceso y contactos que están ubicados en la posición lateral derecha del cuerpo y el contenido que es donde se visualizarán los contenidos de las páginas secundarios.

**Pie de página:** también conocido como footer contiene los derechos reservados del sistema, el nombre del desarrollador, y los logos de las instituciones u organismos que contribuyeron a la ejecución de la aplicación.

#### **4.5.3.2. INTERFAZ DE PÁGINA DE ADMINISTRACIÓN DE SISTEMA**

El sistema al tener dos tipos de usuarios que manipularán datos, existe una interfaz de usuario para la administración parecida a la página principal, con la diferencia que no existe la galería de imágenes por cuanto está colocado en la parte lateral derecha de la página, otra diferencia es las opciones de menú que son diferentes para los dos tipos de actores que interactúan con el sistema web, en el siguiente gráfico se muestra la interfaz de usuario para la administración de datos.

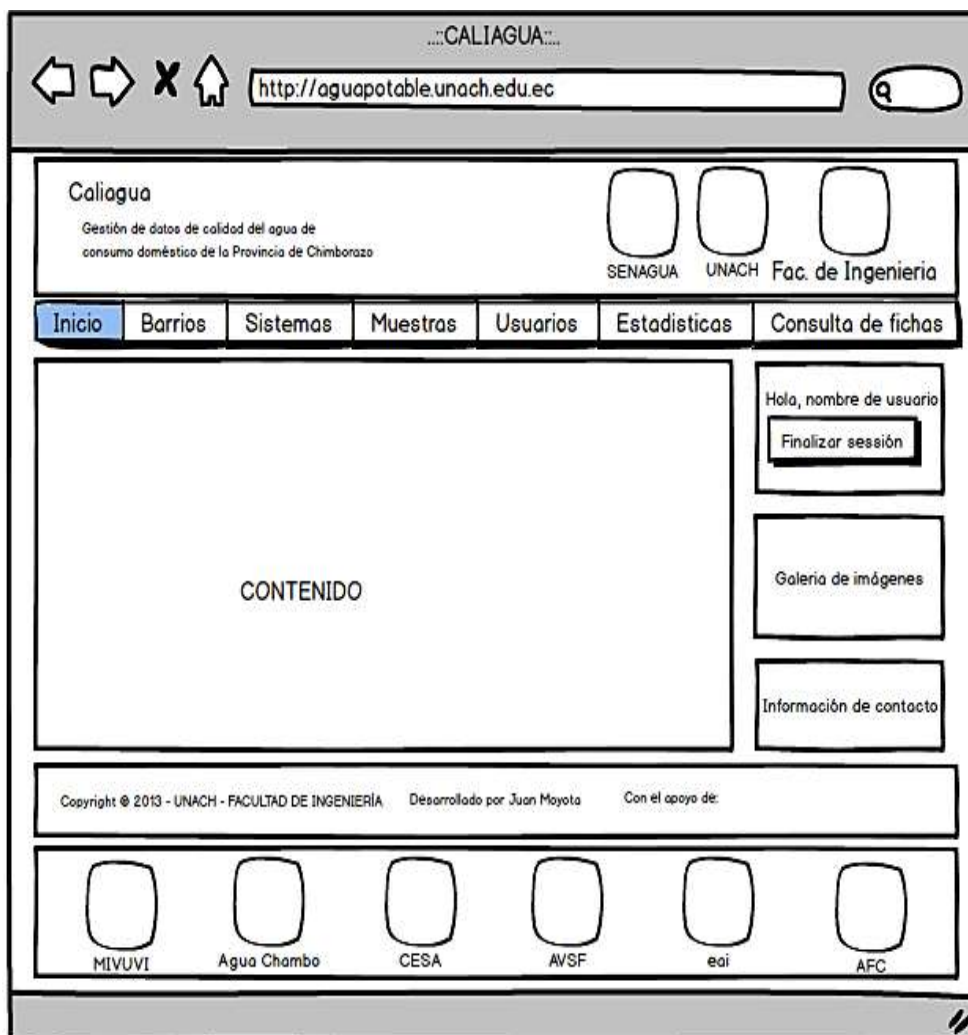


Figura 53. Interfaz de usuario administración de datos – Administrador  
Elaborado por: Juan Carlos Moyota

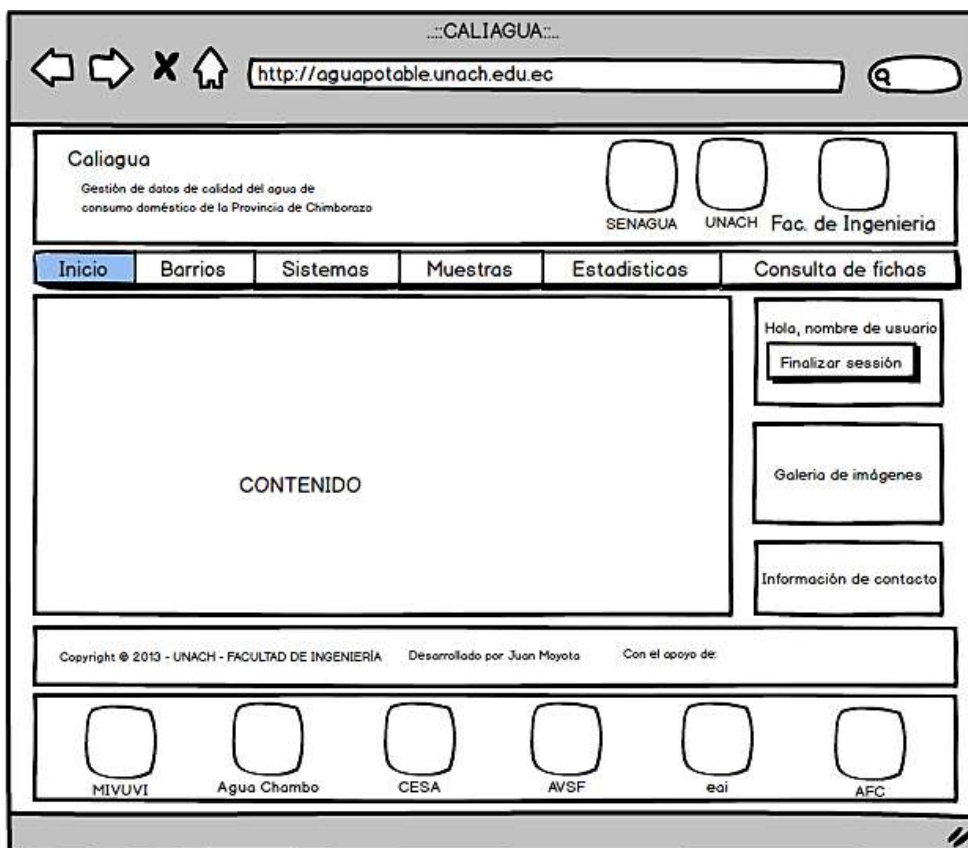


Figura 54. Interfaz de usuario administración de datos – Registrado  
Elaborado por: Juan Carlos Moyota

#### 4.5.3.3. INTERFAZ DE USUARIO ADMINISTRACIÓN DE BARRIOS O COMUNIDADES

Las interfaces para realizar operaciones CRUD con los datos de los barrios o comunidades tienen un diseño de interfaz parecido a la página de administración principal **Figura 54** pero con la diferencia que no se muestran los elementos de la parte lateral derecha. Existirán interfaces de usuario para:

- Registrar nuevo barrio o comunidad
- Actualizar datos de barrios o comunidades

#### 4.5.3.4. INTERFAZ DE USUARIO ADMINISTRAR DATOS DE SISTEMAS DE AGUA POTABLE

Las interfaces para realizar operaciones CRUD con los datos de los sistemas de agua potable tienen un diseño de interfaz como se presenta en la **Figura 54** sin tomar en cuenta los elementos de la parte lateral derecha. Se podrán realizar las siguientes operaciones con los datos de los sistemas de agua potable:



- Agregar nuevo sistema de agua (datos básicos)
- Registrar datos técnicos de los sistemas de agua.
- Actualizar datos de los sistemas de agua potable tanto técnicos como información básica.

#### 4.5.3.5. INTERFAZ DE USUARIO ADMINISTRAR MUESTRAS DE ANÁLISIS DE AGUA

A continuación se presenta la interfaz de usuario del módulo de administrar datos de las fichas de análisis de muestras de agua registrados por los técnicos o laboratoristas de los municipios de la provincia:

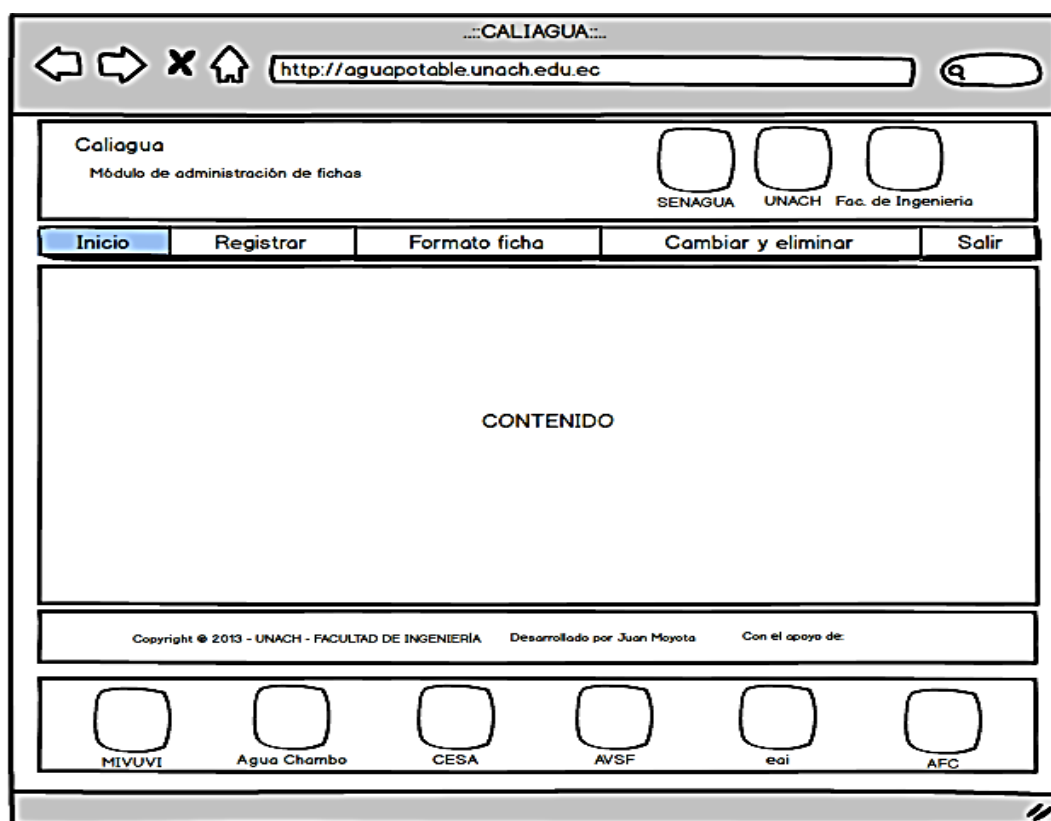


Figura 55. Interfaz de usuario - administración de muestras de agua  
Elaborado por: Juan Carlos Moyota

Las operaciones que el usuario puede realizar dentro del módulo de administración de datos de las muestras están:

- Registrar un nuevo reporte de análisis de muestra de agua
- Actualizar datos de las muestras.
- Eliminar reporte de análisis de muestras de agua potable.

#### 4.5.3.6. INTERFAZ DE USUARIO PARA ADMINISTRACIÓN DE USUARIOS

Al igual que los módulos de administración de datos de barrios y sistemas de agua potable la interfaz de usuario para la administración de datos es la misma de la Figura 57 donde se realizan tareas administrativas como: registrar un nuevo usuario, eliminar y actualizar datos de los usuarios.

#### 4.5.3.7. INTERFAZ DE USUARIO PARA EL MÓDULO DE CONSULTA DE FICHAS

El sistema ayudará a consultar las fichas por sistema y cantón, la siguiente interfaz de usuario representa la página principal del módulo de consulta de fichas:

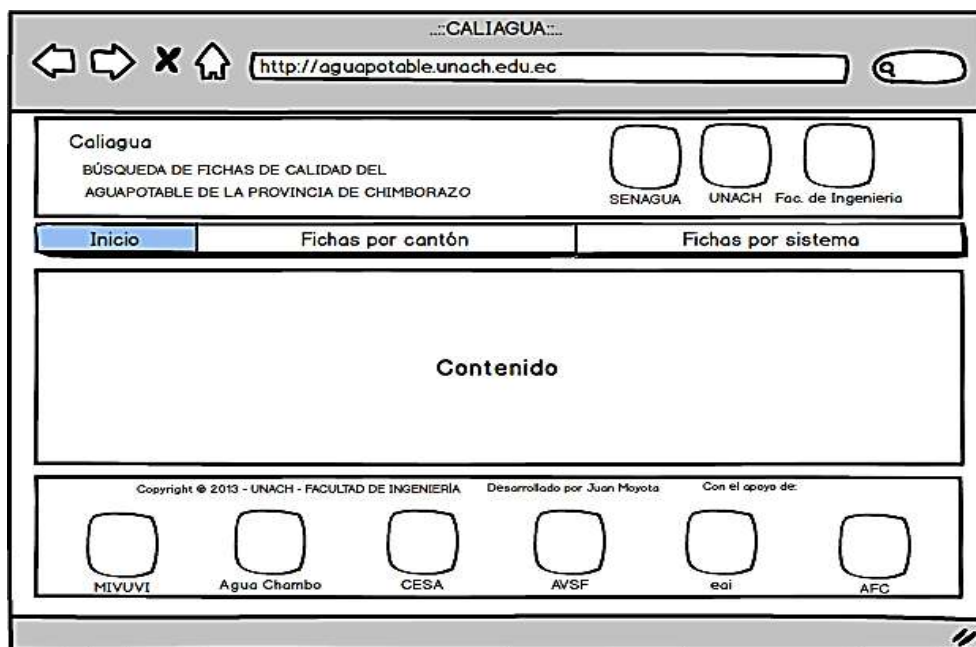
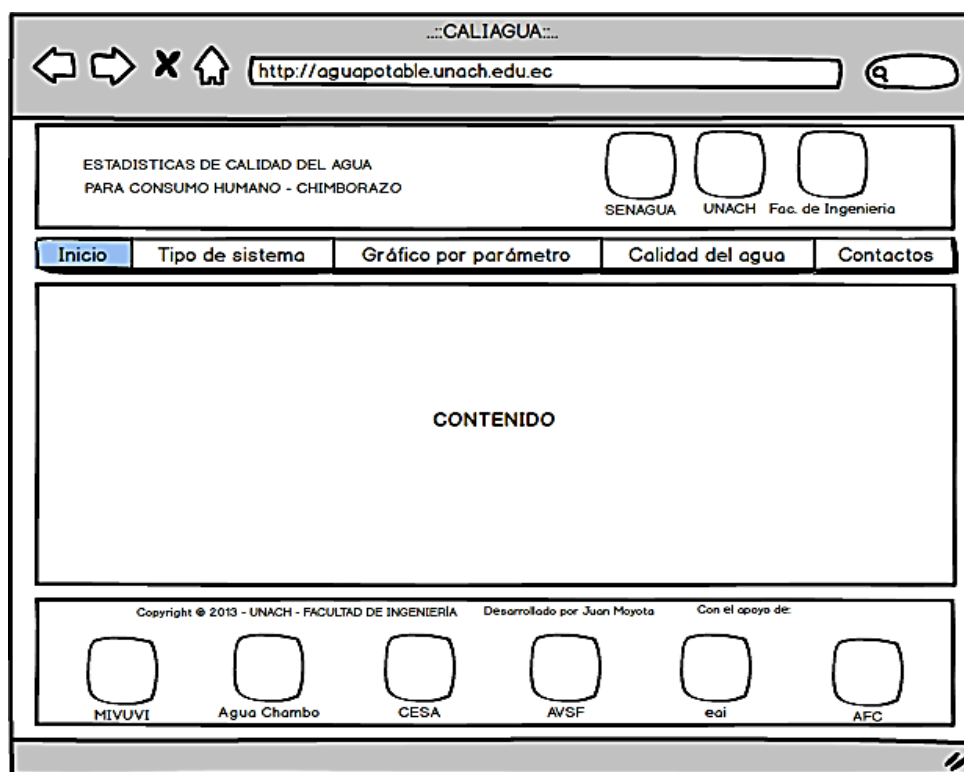


Figura 56. Interfaz de usuario - Modulo de consulta de fichas  
Elaborado por: Juan Carlos Moyota

Las fichas pueden ser consultadas por cantón y por sistema, con la posibilidad de exportar cualquier reporte de análisis de muestra de agua a PDF.

#### 4.5.3.8. INTERFAZ DE USUARIO PARA GENERACIÓN DE ESTADÍSTICAS

La generación de estadísticas será programada con un Framework PHP, con una interfaz de usuario que se presenta en la **Figura 57** como pantalla principal desde donde se podrán acceder a las opciones de menú.



*Figura 57. Interfaz de usuario - módulo de estadísticas  
Elaborado por: Juan Carlos Moyota*

Existen tres tipos de reportes que se pueden generar con este módulo de generación de estadísticas:

- Gráficas de los tipos de sistemas de un determinado cantón y/o parroquia.
- Gráfica de parámetros de calidad del agua potable según las Normas INEN 1108 2011 y Tulas.
- Consultar la calidad del agua de un determinado cantón, donde se muestra el porcentaje de sistemas que cumplen la norma, los que no cumplen y los sistemas que no tengan fichas de muestras registradas.

#### **4.6. GENERACIÓN DE PÁGINAS**

La generación de las páginas del sistema se realizó con herramientas de desarrollo y diseño tanto libres como propietarios, en la ERS (literal 4.4.) se detallaron todas las herramientas software y el hardware informático usado en el desarrollo del sistema. En esta parte del proceso de ingeniería web se presentan las pantallas finales del sistema, cabe mencionar que no se mostrarán todas ya que

en *Manual técnico* se encuentra detallado cómo está estructurado el sistema de archivos, los nombres de los archivos, pantallas, casos de uso detallados, etc.

#### 4.6.1. PÁGINA DE INICIO

En la **Figura 58** se presenta la pantalla principal del sistema web Caliagua:



*Figura 58. Pantalla principal del sistema web Caliagua  
Elaborado por: Juan Carlos Moyota*

#### 4.6.2. PANTALLA DE ADMINISTRACIÓN DE DATOS PARA USUARIO DEL SISTEMA

Las **Figuras 59 y 60** presentan las pantallas de administración de datos para los dos tipos de usuarios que pueden gestionar información del sistema.



Figura 59. Pantalla de administración – usuario administrador  
Elaborado por: Juan Carlos Moyota



Figura 60. Pantalla de administración - usuario registrado  
Elaborado por: Juan Carlos Moyota



### 4.6.3. PANTALLAS DE ADMINISTRAR BARRIOS O COMUNIDADES

En las siguientes gráficas se presentan las pantallas de usuario para registrar y actualizar datos de los barrios:

The screenshot shows the 'Registrar nuevo barrio' (Register new neighborhood) form. At the top, there is a navigation bar with the following menu items: INICIO, BARRIOS, SISTEMAS, MUESTRAS, USUARIOS, ESTADISTICAS, CONSULTA DE FICHAS, and SALIR. The main header area contains the CALIAGUA logo and the text: "Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo". Below the header, there are three logos: the logo of the Secretaría del Agua, the logo of the Universidad Nacional de Chimborazo, and the logo of the Facultad de Ingeniería. The form itself consists of the following fields: 'Cantón (\*)' with a dropdown menu showing '--Seleccionar--'; 'Parroquia (\*)' with a dropdown menu showing '--Seleccionar--'; and 'Nombre del barrio (\*)' with a text input field. A note next to the name field says 'Por favor ingresar sólo letras mayúsculas'. Below the form fields, there is a note: '(\*)Campo necesario'. At the bottom of the form, there are two buttons: 'Borrar' (Delete) and 'Guardar' (Save).

Figura 61. Pantalla de registrar barrios o comunidades  
Elaborado por: Juan Carlos Moyota

The screenshot shows the 'Cambiar nombre del barrio' (Change neighborhood name) form. At the top, there is a navigation bar with the following menu items: SECRETARIA DEL AGUA, DIRECCIÓN PROVINCIAL, RIOBAMBA, and UNIVERSIDAD NACIONAL DE CHIMBORAZO. The main header area contains the CALIAGUA logo and the text: "Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo". Below the header, there are three logos: the logo of the Secretaría del Agua, the logo of the Universidad Nacional de Chimborazo, and the logo of the Facultad de Ingeniería. The form itself consists of the following fields: 'Cantón' with a text input field containing 'Alausi'; 'Parroquia' with a text input field containing 'Pistishi'; and 'Nombre del barrio' with a text input field containing 'Achalsi' and a note: '(\*)'. Below the form fields, there is a note: '(\*) Campo editable'. At the bottom of the form, there is a button: 'Cambiar' (Change).

Figura 62. Pantalla de actualizar datos de un barrio o comunidad  
Elaborado por: Juan Carlos Moyota

#### 4.6.4. PANTALLAS DE ADMINISTRAR SISTEMAS DE AGUA POTABLE

Las siguientes figuras presentan las pantallas para registrar y actualizar datos de los sistemas de agua potable de la provincia de Chimborazo, los datos que se administran son datos básicos y datos técnicos.

The screenshot shows the 'Registro de datos básicos del sistema' form. At the top, there is a blue header with the text 'Secretaría del Agua' and 'Dirección Provincial'. Below the header is a banner for 'CALIAGUA' with the subtitle 'Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo'. The banner includes logos for the 'Secretaría del Agua', 'MUNICIPALIDAD NACIONAL DE CHIMBORAZO', and 'EJECUTIVO PROVINCIAL DE CHIMBORAZO'. A navigation menu below the banner contains links for 'INICIO', 'BARRIOS', 'SISTEMAS', 'MUESTRAS', 'USUARIOS', 'ESTADÍSTICAS', 'CONSULTA DE FICHAS', and 'SALIR'. The form itself has the title 'Registro de datos básicos del sistema' and contains the following fields: 'Cantón (\*)' with a dropdown menu showing '--Seleccionar--'; 'Parroquia (\*)' with a dropdown menu showing '--Seleccionar--'; 'Barrio (\*)' with a dropdown menu showing '--Seleccionar--'; 'Nombre del sistema (\*)' with a text input field; and 'Tipo de sistema (\*)' with a dropdown menu showing 'Tipo'. At the bottom of the form are two buttons: a red 'Borrar' button and a green 'Guardar' button.

Figura 63. Pantalla de registrar sistemas de agua potable (datos básicos)  
Elaborado por: Juan Carlos Moyota

The screenshot shows the 'Cambiar nombre del barrio' form. It features the same header and banner as Figure 63. The form title is 'Cambiar nombre del barrio'. The fields are: 'Cantón' with a text input field containing 'Alausí'; 'Parroquia' with a text input field containing 'Tixán'; and 'Nombre del barrio' with a text input field containing 'Aña Moyocancha' and a red asterisk (\*) indicating it is a required field. At the bottom of the form is a green 'Cambiar' button. Below the form, there is a legend: '(\*) Campo editable'.

Figura 64. Pantalla de actualizar sistemas de agua potable (datos básicos)  
Elaborado por: Juan Carlos Moyota

# CALIAGUA

Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo

Secretaría del Agua

INICIO
BARRIOS
SISTEMAS
MUESTRAS
ESTADÍSTICAS
CONSULTA DE FICHAS
SALIR

Nombre del sistema: Red de distribución Los Pungales

Datos técnicos:

1. Tipo de captación:
 

--Seleccionar--
2. Distancia de conducción:
 

0

 metros
3. Diámetro de tubería:
 

0

Magnitud
4. Tanques rompersiones en conducción:
 

0

 unidades
5. Válvulas de aire:
 

0

 unidades
6. Tanque de reserva:
  - 6.1. Capacidad del tanque:  m<sup>3</sup>
  - 6.2. Material de construcción: 

--Seleccionar--
  - 6.3. Forma del tanque: 

--Seleccionar--
  - 6.4. Año de construcción: 

--Año--
7. Redistribución:
  - 7.1. Longitud de redistribución:  metros.
  - 7.2. Diámetro de la tubería: 

Magnitud
  - 7.3. Rompe presiones:
- 7.4. Número de conexiones domiciliarias:
  - 7.4.1. Con medidor:
  - 7.4.2. Sin medidor:

← Atrás

||

Guardar →

(\*) Campo necesario

Figura 65. Pantalla de registrar datos técnicos de los sistemas de agua  
Elaborado por: Juan Carlos Moyota



Dirección Provincial — Riobamba — Universidad Nacional de Chimborazo — Facultad de Ingeniería —

# CALIAGUA

"Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo"



INICIO BARRIOS SISTEMAS MUESTRAS ESTADÍSTICAS CONSULTA DE FICHAS SALIR

Nombre del sistema: Sistema de agua de consumo Panicocho

Datos técnicos:

1. Tipo de captación:

Campo de filtración

2. Distancia de conducción:

metros

3. Diámetro de tubería:

mm.\*

4. Tanques rompiones en conducción:

unidades

5. Válvulas de aire:

unidades

6. Tanque de reserva:

6.1. Capacidad del tanque:  m<sup>3</sup>

6.2. Material de construcción: **Hormigón armado**

6.3. Forma del tanque: **Cuadrado**

6.4. Año de construcción: **1972**

7. Redistribución:

7.1. Longitud de redistribución:  metros.

7.2. Diámetro de la tubería:  pul\*

7.3. Rompe presiones:

7.4. Número de conexiones domiciliarias:

7.4.1. Con medidor:

7.4.2. Sin medidor:

7.4.3 Población aproximada: 600 Habitantes

||

(\*) Campo editable y necesario

Figura 66. Pantalla de actualizar datos técnicos de sistemas de agua potable  
Elaborado por: Juan Carlos Moyota

#### 4.6.5. PANTALLAS DE ADMINISTRAR DATOS DE LAS MUESTRAS DE AGUA

La pantalla de administración de fichas de datos de muestras de análisis de agua se ejecuta en una página deferente a la página de administración pero con el mismo diseño que se presenta a continuación:



Figura 67. Pantalla de administración de fichas de muestras de agua potable  
Elaborado por: Juan Carlos Moyota

Secretaría del Agua — Dirección Provincial — Riobamba — Unive

## CALIAGUA

MÓDULO DE ADMINISTRACIÓN DE FICHAS

Inicio REGISTRAR FORMATO FICHA CAMBIAR Y ELIMINAR SALIR

Registrar fichas

### Reporte de análisis de agua

**1. DATOS DE LA MUESTRA**

Número de Muestra:

Provincia: Chimborazo      Cantón:

Parroquia:       Barrio/Comunidad:

Nombre del sistema:        Tipo:

Departamento:

Responsable de la recolección:

Fecha de recolección:            Fecha de análisis:

Hora de recolección: hh  min       Hora de análisis: hh  min

Sitio de toma de la muestra:

**2. Características Físicas:**

Parámetro	Unidad	Límite	Resultados
pH	Unidades	6 - 9 *	<input type="text"/>
Color	Color real	15	<input type="text"/>
Olor	--	No objetable	<input type="text"/>
Turbiedad	NTU	5	<input type="text"/>
Temperatura	°C	Condición natural+/-3 °C *	<input type="text"/>
Sólidos totales disueltos	mg/l	500 *	<input type="text"/>
Conductividad	µS/cm	- *	<input type="text"/>

**3. Características químicas:**

Parámetro	Unidad	Límite	Resultados
Dureza, CaCO <sub>3</sub>	mg/l	500 *	<input type="text"/>
Cloro libre residual	mg/l	0,2 a 1,5	<input type="text"/>
Hierro total, Fe <sup>+</sup>	mg/l	0,3 *	<input type="text"/>
Nitratos, NO <sub>3</sub> <sup>-</sup>	mg/l	50	<input type="text"/>
Nitritos, NO <sub>2</sub> <sup>-</sup>	mg/l	3,0	<input type="text"/>
Sulfatos, SO <sub>4</sub> <sup>-</sup>	mg/l	250 *	<input type="text"/>
Fosfatos, PO <sub>4</sub> <sup>-</sup>	mg/l	0,3 *	<input type="text"/>
Manganeso, Mn	mg/l	0,1 *	<input type="text"/>
Fluoruros (Flúor), F	mg/l	1,5	<input type="text"/>
Nitrógeno amoniacal(Amoníaco), NH <sup>+</sup>	mg/l	1,0 *	<input type="text"/>

**4. Requisitos microbiológicos:**

Parámetro	Unidad	Límite	Resultados
Coliformes totales	NMP/100 ml	3000 *	<input type="text"/>
Coliformes fecales	NMP/100ml	50 *	<input type="text"/>

(\*) Límites permisibles para aguas de consumo humano y uso doméstico, que únicamente requieren tratamiento convencional basados en la norma TULAS Libro VI, Anexo I, Tablas 1 y 2.

Observaciones:

Límite permisible: NORMA TÉCNICA ECUATORIANA(NTE) INEN 1108 QUINTA REVISIÓN 2014 01

||

Figura 68. Pantalla para registrar muestras de agua potable  
Elaborado por: Juan Carlos Moyota

Inicio | Iñabamba | Universidad Nacional del Chimborazo | Facultad de Ingeniería

# CALIAGUA

MÓDULO DE ADMINISTRACIÓN DE FICHAS

Secretaría del Agua

INICIO REGISTRAR FORMATO FICHA CAMBIAR Y ELIMINAR SALIR

## Reporte de análisis de agua

**1. DATOS DE LA MUESTRA**

Número de Muestra: 92

Provincia: Chimborazo      Cantón: Iñabamba

Parroquia: Pungará      Barrio/Comunidad: TIC-TIC Alto San Antonio

Nombre del sistema: Sistema de agua de consumo TIC      Tipo: A gravedad

Institución: Gobierno Autónomo Descentralizado Municipal del Cantón Iñabamba

Departamento: LABORATORIO BROMATOLÓGICO MUNICIPAL

Recolector: DRA. ANA MARIA LÓPEZ, RESPONSABLE TÉCNICA

Fecha de recolección: 2011-07-26      Fecha de análisis: 2011-07-26

Hora de recolección: 10:30:00      Hora de análisis: 11:30:00

Sito de toma de la muestra: RCD DE DISTRIBUCION

**2. Características Físicas:**

Parámetro	Unidad	Límite	Resultados
pH	Unidades	6 - 9 *	7
Color	Color real	15	10
Olor	--	No objetable	
Turbiedad	N TU	5	1.3
Temperatura	°C	Condición natural+/- 3 °C *	11
Sólidos totales disueltos	mg/l	500 *	100
Conductividad	µS/cm	--	210

**3. Características químicas:**

Parámetro	Unidad	Límite	Resultados
Dureza, CaCO <sub>3</sub>	mg/l	500 *	
Cloro libre residual	mg/l	0,3 a 1,5	
Hierro total, Fe <sup>+</sup>	mg/l	0,3 *	0.01
Nitratos, NO <sub>3</sub> <sup>-</sup>	mg/l	50	0.4
Nitritos, NO <sub>2</sub> <sup>-</sup>	mg/l	3.0	0.03
Sulfatos, SO <sub>4</sub>	mg/l	250 *	0
Fosfatos, PO <sub>4</sub> <sup>-</sup>	mg/l	0,3 *	0.21
Manganeso, Mn	mg/l	0,1 *	0.011
Fluoruros (Flúor), F	mg/l	1.0	0.25
Nitrógeno amoniacal(Amoníaco), NH <sup>+</sup>	mg/l	1,0 *	0.01

**4. Requisitos microbiológicos:**

Parámetro	Unidad	Límite	Resultados
Coliformes totales	NMP/100 ml	3000 *	
Coliformes fecales	NMP/100ml	50 *	0

(\*) Límites permisibles para aguas de consumo humano y uso doméstico, que únicamente requieren tratamiento convencional basado en la norma TULAS Libro VI, Anexo I, Tablas 1 y 2.

Observaciones:

Límite permisible: NORMA TÉCNICA ECUATORIANA(NTE) INEN 1108 QUINTA REVISIÓN 2014-01

Guardar

Figura 69. Pantalla para actualizar datos de muestras de agua potable  
Elaborado por: Juan Carlos Moyota

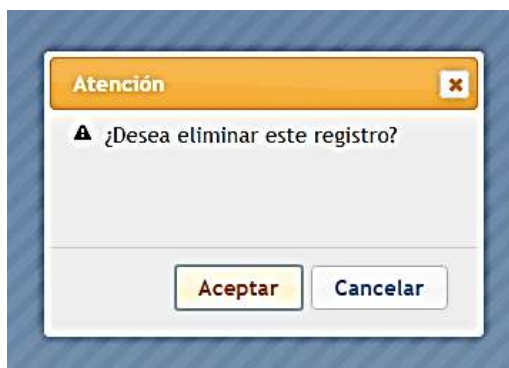


Figura 70. Pantalla de confirmación para eliminar muestra  
Elaborado por: Juan Carlos Moyota

#### 4.6.6. PANTALLAS DE ADMINISTRACIÓN DE USUARIOS

Las operaciones de registrar, eliminar y actualizar datos de los usuarios se presenta en los siguientes gráficos:

A screenshot of a web application interface for user registration. The header features the logo for "CALIAGUA" with the tagline "Gestión de datos de calidad del agua de consumo domestico de La Provincia de Chimborazo". To the right of the header are logos for the "Secretaría de Agua" and other regional entities. A blue navigation bar contains the following menu items: INICIO, BARRIOS, SISTEMAS, MUESTRAS, USUARIOS, ESTADÍSTICAS, CONSULTA DE FICHAS, and SALIR. Below the navigation bar, the page title is "Registrar nuevo usuario". The form includes the following fields: "Nombre de usuario:" (text input), "Contraseña:" (password input), "Repetir contraseña:" (password input), "Cantón:" (dropdown menu with "Seleccione" selected), "Perfil:" (radio buttons for "Registrado" and "Administrador"), and "E-mail:" (text input). At the bottom of the form, there are two buttons: "Registrar" (red) and "Cancelar" (green).

Figura 71. Pantalla de registrar nuevo usuario  
Elaborado por: Juan Carlos Moyota



**CALIAGUA**  
"Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo"

INICIO BARRIOS SISTEMAS MUESTRAS ESTADÍSTICAS CONSULTA DE FICHAS SALIR

**DATOS ACTUALES:**

Nombre de usuario: Administrador2  
 Contraseña: admin2013  
 E-mail: cobajose@hotmail.es

**NUEVOS DATOS:**

Nombre de usuario:

Contraseña:

Repetir contraseña:

Figura 72. Pantalla de registro de nuevo usuario  
 Elaborado por: Juan Carlos Moyota

Secretaría del Agua Dirección Provincial Facultad de Ingeniería

**CALIAGUA**  
"Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo"

INICIO BARRIOS SISTEMAS MUESTRAS USUARIOS ESTADÍSTICAS CONSULTA DE FICHAS SALIR

**LISTA DE USUARIOS DEL SISTEMA**

USUARIO	PERFIL	CANTÓN	EDITAR	ELIMINAR
Administrador2	administrador	Riobamba		
SUPER_ADMIN	administrador	Riobamba		
juancarlos	administrador	Riobamba		
Administrador3	administrador	Riobamba		
GAD_Riobamba	registrado	Riobamba		
GAD_PALLATANGA	registrado	Pallatanga		
juanka2106	registrado	Riobamba		

Figura 73. Pantalla de mensaje de confirmación al eliminar un usuario  
 Elaborado por: Juan Carlos Moyota

#### 4.6.7. PANTALLAS DE CONSULTA DE FICHAS POR CANTÓN Y PARROQUIA

La consulta de fichas de muestras de análisis de agua se presenta en los gráficos siguientes:

The screenshot shows a web application interface for searching water quality records by canton. At the top, it features the text 'CALIAGUA BÚSQUEDA DE FICHAS DE CALIDAD DEL AGUA POTABLE DE LA PROVINCIA DE CHIMBORAZO' and logos for the 'Secretaría del Agua', 'UNACH', and 'FACULTAD DE INGENIERIA'. Below this is a navigation bar with 'Fichas por cantón' and 'Fichas por sistema'. The main search area is titled 'Seleccione un cantón:' and contains a dropdown menu for 'Cantón (\*)' with a '---Seleccionar---' option. There are 'Borrar' and 'Buscar' buttons, and a note: '(\*) Campo necesario para la búsqueda'. At the bottom, there is a copyright notice: 'Copyright © 2013 - UNACH - FACULTAD DE INGENIERIA. | Desarrollado por: Juan C. Moyota | Con el apoyo de:' followed by logos for 'MINISTERIO DE DESARROLLO URBANO Y RURALIDAD (MIDURI)', 'AGUA CHIMBO', 'cesa', 'agronomía veterinaria', 'eau SEINE NOROINDE', and 'afca'.

Figura 74. Pantalla de consulta de fichas por cantón  
Elaborado por: Juan Carlos Moyota

The screenshot shows a web application interface for searching water quality records by system. It has the same header and navigation bar as Figure 74. The main search area is titled 'Seleccione los datos del sistema:' and contains four dropdown menus for 'Cantón (\*)', 'Parroquia (\*)', 'Barrio (\*)', and 'Sistema (\*)', each with a '---Seleccionar---' option. There are 'Borrar' and 'Buscar' buttons, and a note: '(\*) Campo necesario para la búsqueda'. At the bottom, there is a copyright notice: 'Copyright © 2013 - UNACH - FACULTAD DE INGENIERIA. | Desarrollado por: Juan C. Moyota | Con el apoyo de:' followed by logos for 'MINISTERIO DE DESARROLLO URBANO Y RURALIDAD (MIDURI)', 'AGUA CHIMBO', 'cesa', 'agronomía veterinaria', 'eau SEINE NOROINDE', and 'afca'.

Figura 75. Pantalla de consulta de fichas por sistema de agua potable  
Elaborado por: Juan Carlos Moyota

#### 4.6.8. PANTALLA DE GENERACIÓN DE ESTADÍSTICAS

La generación de estadísticas de calidad del agua se presenta en los gráficos siguientes y es necesario que existan registradas muestras de análisis de agua potable que registran los técnicos de los municipios de la provincia de Chimborazo.



Figura 76. Pantalla principal de generación de estadísticas  
Elaborado por: Juan Carlos Moyota



Figura 77. Pantalla de generar estadísticas de tipos de sistemas  
Elaborado por: Juan Carlos Moyota





Figura 78. Pantalla para generar gráfico de parámetros de calidad de agua  
Elaborado por: Juan Carlos Moyota



Figura 79. Pantalla para consultar la calidad del agua según su cantón  
Elaborado por: Juan Carlos Moyota



Figura 80. Pantalla de gráfico de calidad del agua según su cantón  
Elaborado por: Juan Carlos Moyota



Figura 81. Pantalla de gráfico estadístico de parámetro de calidad del agua  
Elaborado por: Juan Carlos Moyota



Figura 82. Pantalla de gráfico estadístico de los tipos de sistemas de agua  
Elaborado por: Juan Carlos Moyota

#### 4.7. PRUEBAS

Una vez realizados los prototipos de interfaz iniciales y todo el código para realizar las operaciones del sistema se realizaron los siguientes cambios:

- Cambio de interfaz de usuario de la página principal del sistema, página de administración de datos.
- Implementar operaciones de administración de datos técnicos de los sistemas de agua potable lo cual no estaba contemplado al inicio del sistema.
- Diseñar e implementar nuevas interfaces para consultar fichas de análisis de muestras de agua.
- Diseñar e implementar una nueva interfaz para el módulo de administración de fichas de análisis de agua.
- Agregar la operación de eliminar fichas de muestras de análisis de agua.

## 4.8. IMPLEMENTACIÓN DEL SISTEMA WEB CALIAGUA

El sistema de gestión de datos de calidad del agua de consumo doméstico de la provincia de Chimborazo, CALIAGUA, está alojado en un servidor web de propiedad de *Universidad Nacional de Chimborazo* así como la infraestructura de red utilizada.

- La base de datos MySQL lleva el nombre de “**caliagua**” que es el archivo que contiene toda la estructura de las tablas y los procedimientos almacenados que se utilizan para generar las estadísticas. Esta base de datos se necesita restaurar en el SGBD que se aloja en el mismo servidor web.
- La carpeta de archivos lleva el nombre de “**caliagua**” la cual fué alojada en la carpeta raíz del servidor `/var/www/html/`

### 4.8.1. ARCHIVO DE CONEXIÓN CON LA BASE DE DATOS

El código del archivo de conexión “*Conexion.php*” con la base de datos y la aplicación es el siguiente:

```
<?php
class Conexion
{
    public static function con()
    {
        $conexion = @mysql_connect("localhost","root","2321") or
die ("EXISTEN PROBLEMAS EN EL SERVIDOR.".mysql_error());

        mysql_query("SET NAMES 'utf8'");

        mysql_select_db("caliagua") or die ("EXISTEN PROBLEMAS CON
LA BASE DE DATOS.".mysql_error());

        return $conexion;
    } ?>
```

#### 4.8.2. CONEXIÓN DEL MÓDULO DE ESTADÍSTICAS CON LA BASE DE DATOS

El módulo de generación de estadísticas del sistema web CALIAGUA está desarrollado en **Yii framework PHP 1.1.14** el cual tiene un archivo de configuración principal “*main.php*”, el código fuente que se presenta a continuación configura toda la aplicación de generación de estadísticas.

En las líneas del siguiente código fuente se realizan las siguientes configuraciones para hacer que funcione la aplicación web:

- Incluye un archivo de funciones para utilizarlas en cualquier vista o controlador.
- Importa la carpeta de modelos con todos los modelos creados para cada una de las tablas de base de datos.
- Habilita la navegación de urls limpias.
- Conecta la aplicación con la base de datos.

```
<?php

// uncomment the following to define a path alias

// Yii::setPathOfAlias('local','path/to/local-folder');

// This is the main Web application configuration. Any writable
// CWebApplication properties can be configured here.

//Yii::setPathOfAlias('chartjs', dirname(__FILE__).'/../extensions/yii-
chartjs');

require_once (dirname(__FILE__).'/../components/helpers.php');

return array(

    'basePath'=>dirname(__FILE__).DIRECTORY_SEPARATOR.'..',

    'name'=>'Estadísticas de calidad del agua para consumo humano',

    // preloading 'log' component

    'preload'=>array('log'),

    // autoloading model and component classes
```

```

'import'=>array(
    'application.models.*',
    'application.components.*',
),
'modules'=>array(
    // uncomment the following to enable the Gii tool
    'gii'=>array(
        'class'=>'system.gii.GiiModule',
        'password'=>'12345',
        'ipFilters'=>array('127.0.0.1','::1'),
    ),
),
// application components
'components'=>array(
    'user'=>array(
        // enable cookie-based authentication
        'allowAutoLogin'=>true,
    ),
    // uncomment the following to enable URLs in path-format
    'urlManager'=>array(
        'urlFormat'=>'path',
        'showScriptName'=>false,
        'urlSuffix'=>'.php',
        'rules'=>array(
            '<controller:\w+>/<id:\d+>'=>'<controller>/view',

```

```

        '<controller:\w+>/<action:\w+>/<id:\d+>'=>'<controller>/<action>'
    ,
    '<controller:\w+>/<action:\w+>/<id:\d+>/<id2:\d+>'=>'<controller>/<action>',
    '<controller:\w+>/<action:\w+>/<id:\d+>/<id2:\d+>/<id3:\d+>'=>'<controller>/<action>',
        '<controller:\w+>/<action:\w+>'=>'<controller>/<action>',
    ),
),
/*'db'=>array(
    'connectionString'=>
'sqlite:'.dirname(__FILE__).'/../data/testdrive.db',
),*/
// uncomment the following to use a MySQL database
'db'=>array(
    'connectionString'=>
'mysql:host=localhost;dbname=caliagua;charset=utf8',
    'emulatePrepare' => true,
    'username' => 'root',
    'password' => '2321',
    'charset' => 'utf8',
),
'errorHandler'=>array(
    // use 'site/error' action to display errors
    'errorAction'=>'site/error',
),
'log'=>array(
    'class'=>'CLogRouter',
    'routes'=>array(
        array(

```

```

        'class'=>'CFileLogRoute',
        'levels'=>'error, warning',
    ),
    // uncomment the following to show log messages on web pages
    /*
    array(
        'class'=>'CWebLogRoute',
    ),
    */
),
),
    //'chartjs'=>array('class' => 'chartjs.components.ChartJs'),
),
// application-level parameters that can be accessed
// using Yii::app()->params['paramName']
'params'=>array(
    // this is used in contact page
    'adminEmail'=>'juankamoyota21@gmail.com',
),
);

```

#### 4.9. DOMINIO DE LA APLICACIÓN

El dominio o dirección web fué establecido como: *<http://aguapotable.unach.edu.ec>* ya que al existir un convenio entre la institución contratante CESA y la UNACH hubo la facilidad de poder alojar el sistema web y la base de datos MySQL.



## CAPÍTULO V

### METODOLOGÍA

#### 5.1. TIPO DE ESTUDIO

A continuación se detalla los tipos de investigación que se utilizan para este proyecto de investigación:

*Según el objeto de estudio:* investigación aplicada, por cuanto involucra el desarrollo de una aplicación para la gestión de reportes estadísticos con el uso de una herramienta framework PHP.

*Según la fuente de información:* es una investigación bibliográfica, se obtuvo información teórica de los frameworks PHP en libros, revistas, manuales, etc. para poder realizar una evaluación, calificación y selección del framework PHP.

*Según las variables:* es una investigación experimental, con el uso de criterios de medición y evaluación de las WebApp se evalúa el desempeño de una aplicación desarrollada con framework y otra con programación tradicional, con esto estableceremos que la aplicación realizada con el framework mejora o no el desarrollo de la aplicación web para generación de reportes estadísticos.

#### 5.2. POBLACIÓN Y MUESTRA

##### 5.2.1. POBLACIÓN

La población es considerada los frameworks escogidos para ser analizados los cuales se mencionan los siguientes:

- Yii
- CodeIgniter
- CakePHP

### 5.2.2. TAMAÑO DE LA MUESTRA

Al considerar la población muy pequeña se trabaja con toda la población.

### 5.3. OPERACIONALIZACIÓN DE VARIABLES

Variable	Tipo	Definición Conceptual	Dimensión	Indicadores
Framework PHP opensource	Independiente	Framework PHP opensource es una herramienta de soporte que permite que otras aplicaciones se implementado y desarrollado.	Framework PHP opensource según su funcionalidad: Acceso a datos  Interfaz de usuario  Conectividad entre sistemas  Modularidad	Número de base de datos compatibles.  Soporte de la comunidad.  Número de proyectos desarrollados.  Tipo de licencia
Desarrollo de una aplicación web para la gestión de reportes estadísticos.	Dependiente	Técnicas de una mejor comprensión de los datos mediante la representación de los números con el uso de gráficas.	Gráficas estadísticas.  Tablas.  Reportes.	Líneas de código  Esfuerzo aplicado  Tiempo de desarrollo  Costo del desarrollo del software

Tabla 35. Operacionalización de variables  
Elaborado por: Juan C. Moyota

### 5.4. PROCEDIMIENTOS

#### 5.4.1. TÉCNICAS DE INVESTIGACIÓN

Las técnicas de recolección de datos aplicadas para este proyecto son:

- Entrevistas
- Observación

**Entrevista:** se realizó entrevistas con los técnicos encargados del departamento de agua potable y saneamiento de la SENAGUA con los cuales se establecieron los requerimientos de software.

**Observación:** se cuenta el número de las líneas de código fuente de un módulo pequeño de la generación de reportes estadísticos el mismo que estará programado con un framework y este mismo módulo sin usar el framework.

## **5.5. HIPÓTESIS**

### **5.5.1. HIPÓTESIS DE INVESTIGACIÓN**

El uso de un Framework PHP opensource mejorará el desarrollo de una aplicación web para la gestión de reportes estadísticos.

### **5.5.2. DETERMINACIÓN DE LAS VARIABLES**

**Variable independiente:**

Framework PHP opensource

**Variable dependiente:**

Desarrollo de una aplicación web para la gestión de reportes estadísticos.

## **5.6. PROCESAMIENTO Y ANÁLISIS**

### **5.6.1. COMPROBACIÓN DE LA HIPÓTESIS**

Para comprobar la hipótesis planteada en esta investigación se realiza una estimación del software con el uso del modelo de estimación de software COCOMO en base a líneas de código fuente LDC que se utilizaron para realizar la generación de una gráfica de parámetros de calidad del agua, en dos aplicaciones web:

- Una aplicación web programada con el framework Yii 1.1.14 (WebApp 1).
- Una aplicación web desarrollada sin framework (WebApp 2).

Se plantean las siguientes hipótesis:

**H<sub>1</sub>:** Un Framework PHP opensource mejorará el desarrollo de una aplicación web para la gestión de reportes estadísticos.

**Ho:** Un Framework PHP opensource no mejorará el desarrollo de una aplicación web para la gestión de reportes estadísticos.

#### 5.6.1.1. CÁLCULOS DE DATOS INICIALES

Para establecer el número de líneas de código se toma en cuenta las reglas de conteo de líneas de código establecidas en el literal 2.8.2. El código fuente utilizado en la programación de los módulos de prueba se muestra en el **Anexo 4**.

La aplicación web desarrollada con el framework **WebApp 1** utiliza la arquitectura MVC se realiza el conteo de las líneas de código existente en los archivos de modelo, vista y el controlador además de las funciones adicionales.

<i>Líneas de código por componente</i>	<i>Líneas de código formulario de datos de consulta</i>	<i>Líneas de código del reporte estadístico</i>
<b>Modelo</b>	17	17
<b>Controlador</b>	78	78
<b>Vista</b>	157	111
<b>Funciones</b>	122	122
<b>Total:</b>	485	

Tabla 36. Líneas de código fuente de la aplicación WebApp 1  
Elaborado por: Juan Carlos Moyota

La **WebApp 2** es desarrollada sin framework y se contarán las líneas de código de los archivos HTML y PHP, clases y funciones de esta manera se tienen los datos de la siguiente tabla:

<i>Líneas de código</i>	
<i>Líneas de código formulario de datos de consulta</i>	144
<i>Líneas de código del reporte estadístico</i>	172
<i>Líneas de código de las funciones</i>	299
<b>Total:</b>	<b>615</b>

Tabla 37. Líneas de código fuente de la aplicación WebApp 2  
Elaborado por: Juan Carlos Moyota

En la **Tabla 38** se muestran los resultados de LDC para las dos aplicaciones web de prueba.

<i>Aplicación</i>	<i>Líneas de código</i>
<i>WebApp 1 con framework</i>	<i>485</i>
<i>WebApp 2 sin framework</i>	<i>615</i>

*Tabla 38. Líneas totales de código fuente de las aplicaciones de prueba  
Elaborado por: Juan Carlos Moyota*

Para poder estimar el software se calculan los valores del esfuerzo (E), tiempo de desarrollo (D) del software, con estos valores se podrá estimar el software e identificar la aplicación con el menor número de personas en un menor tiempo de desarrollo y por consecuencia con el menor costo del personal utilizado para la programación del software.

En la siguiente tabla se expresan las líneas de código en miles de LDC (KLDC) que es un valor importante para el cálculo de la estimación del software en base a LDC.

	<i>WebApp 1</i>	<i>WebApp 2</i>
<i>Líneas de código LDC</i>	<i>485</i>	<i>615</i>
<i>Líneas de código en KLDC</i>	<i>0.485</i>	<i>0.615</i>

*Tabla 39. KLDC para las aplicaciones de prueba  
Elaborado por: Juan Carlos Moyota*

Para realizar los cálculos del esfuerzo **E** y el tiempo de desarrollo **D** se utilizan las fórmulas:

$$E = a_b KLDC^{b_b} \quad F1$$

$$D = C_b E^{d_b} \quad F2$$

***El tipo de proyecto corresponde a un proyecto orgánico ya que es un software relativamente pequeño y sencillo.***

Los coeficientes  $a_b$  y  $C_b$  y los exponentes  $d_b$  y  $b_b$  son valores constantes que se detallan en la siguiente tabla:

<i>Proyecto</i>	$a_b$	$b_b$	$C_b$	$d_b$
<b>Orgánico</b>	2.4	1.05	2.5	0.38
<b>Semiacoplado</b>	3.0	1.12	2.5	0.35
<b>Empotrado</b>	3.6	1.20	2.5	0.32

Tabla 40. Valores coeficientes y exponentes  
Elaborado por: Juan Carlos Moyota

Se puede observar que a medida que aumenta la complejidad del proyecto, las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno.

#### ***Cálculos de esfuerzo (E) y tiempo (D) para la aplicación WebApp 1***

$$E = (2.4) (0.485)^{1.05} = 1.12 \text{ personas-mes}$$

$$D = (2.5) (1.12)^{0.38} = 2.61 \text{ meses}$$

#### ***Cálculos de E y D para la aplicación WebApp 2***

$$E = (2.4) (0.615)^{1.05} = 1.44 \text{ personas-mes}$$

$$D = (2.5) (1.44)^{0.38} = 2.87 \text{ meses}$$

Con los resultados obtenidos de los cálculos y el conteo de LDC realizados se obtienen los siguientes resultados de tiempo y esfuerzo.

#### ***Cálculo de horas totales para el software:***

$$Nh_{\text{WebApp 1}} = 2.61 \text{ meses} * \frac{20 \text{ dias}}{1 \text{ mes}} * \frac{8 \text{ horas}}{1 \text{ dia}} = 417.6 \text{ horas}$$

$$Nh_{\text{WebApp 2}} = 2.87 \text{ meses} * \frac{20 \text{ dias}}{1 \text{ mes}} * \frac{8 \text{ horas}}{1 \text{ dia}} = 459.2 \text{ horas}$$

Donde  $Nh_{\text{WebApp 1}}$  es el número de horas utilizadas para el desarrollo de la aplicación con framework y  $Nh_{\text{WebApp 2}}$  la aplicación sin framework.

***Cálculo del costo de la hora de trabajo:*** para establecer el costo que involucra la programación del software se necesita conocer el valor de la hora de trabajo para

un programador o técnico afín. Razón por la cual se necesita conocer el sueldo promedio de un programador o técnico afín que en la tabla 43 se presenta esta información.

Técnico	Sueldo
Programador	\$ 800
Diseñador	\$ 800
Desarrollador de base de datos	\$ 800

*Tabla 41. Sueldo promedio de un desarrollor de sistemas.*

*Fuente: Calderón & Tierra, (2011). Trabajo de graduación. Estudio de tecnologías Open Source y su incidencia en el Costo del desarrollo de aplicaciones web. Caso aplicativo: Sistema de facturación de la empresa Nachos Sport.*

Como se aprecia en la tabla 42 tenemos un sueldo promedio de \$ 800 dólares que se considera para 160 horas en 20 días de trabajo. Entonces se divide el sueldo para las 160 horas se obtiene el valor por hora de trabajo.

$$H_{trabajo} = \frac{\$ 800}{160 \text{ horas}} = 5 \text{ dólares/hora}$$

**Cálculo del costo del desarrollo del software:** Para estimar el costo del personal involucrado en el desarrollo del software el tiempo de desarrollo se debe estimar en horas de trabajo, se considera que se trabaja 8 horas diarias en un lapso de 4 semanas. Para establecer el costo del personal para el desarrollo del software se aplica la siguiente fórmula:

$$Costo_{personal} = H_{trabajo} * Nh_{WebApp}$$

Donde ***Htrabajo*** es el costo por hora de trabajo, ***NHWebApp*** es el número de horas de desarrollo de la aplicación y ***Costo\_{personal}*** es el costo del personal para el desarrollo del software.

$$Costo_{personal} - WebApp 1 = 5 \text{ dólares/hora} * 417.6 \text{ horas} = \$ 2088 \text{ dólares}$$

$$Costo_{personal} - WebApp 2 = 5 \text{ dólares/hora} * 459.2 \text{ horas} = \$ 2296 \text{ dólares}$$

### 5.6.1.2. RESULTADOS DE CÁLCULOS

<i>Parámetro</i>	<i>WebApp 1 con framework</i>	<i>WebApp 2 sin framework</i>
<i>Líneas de código</i>	485	615
<i>Esfuerzo aplicado</i>	1.12 personas-mes	1.44 personas-mes
<i>Tiempo de desarrollo</i>	417.6 horas	459.2 horas
<i>Costo del desarrollo del software</i>	\$ 2088	\$ 2296

*Tabla 42. Resumen de resultados de la estimación del software  
Elaborado por: Juan Carlos Moyota*

### 5.6.1.3. DATOS PARA CÁLCULO DE PRUEBA DE HIPÓTESIS

Con los resultados de la **Tabla 42** se determina un grupo de datos que representan las líneas de código que permite conocer el esfuerzo de 1.12 personas y que dentro de este intervalo de datos está el número de líneas de código que se utilizaron para desarrollar la aplicación prototipo y de la misma manera un grupo de datos de líneas de código que permite conocer el esfuerzo de 1.44 personas para desarrollar la aplicación prototipo.

En las **Tablas 43 y 44** podemos observar 10 datos para cada una de las aplicaciones prototipo.



LDC	KLDC	Esfuerzo
481	0.481	1.11
482	0.482	1.12
483	0.483	1.12
484	0.484	1.12
485	0.485	1.12
486	0.486	1.13
487	0.487	1.13
488	0.488	1.13
489	0.489	1.13
490	0.49	1.13

*Tabla 43. Intervalo de datos de la WebApp 1  
Elaborado por: Juan C. Moyota*

LDC	KLDC	Esfuerzo
611	0.611	1.43
612	0.612	1.43
613	0.613	1.44
614	0.614	1.44
615	0.615	1.44
616	0.616	1.44
617	0.617	1.45
618	0.618	1.45
619	0.619	1.45
620	0.62	1.45

*Tabla 44. Intervalo de datos de la WebApp 2  
Elaborado por: Juan C. Moyota*

En la **Tabla 45** observamos el grupo de datos que representan las líneas de código que son programadas por 1.12 personas-mes para la WebApp 1 y de la misma manera por 1.44 personas-mes para la WebApp 2.

N°	LDC con Framework	LDC sin framework
1	482	613
2	483	614
3	484	615
4	485	616

Tabla 45. Líneas de código con y sin framework

Elaborado por: Juan C. Moyota

#### 5.6.1.4. PRUEBA DE HIPÓTESIS

El método de comprobación de hipótesis aplicado en la investigación es el método de comparación de medias independientes como los datos de la Tabla 45 son menores que 30 se utiliza la distribución de probabilidad T-Student.

Los pasos para la prueba de hipótesis son:

- 1) Determinar las hipótesis  $H_0$  y  $H_1$

**Hipótesis nula:** el promedio de líneas de código generadas con el framework es mayor que el promedio de líneas de código sin usar el framework.

$$H_0 = \mu_{cf} > \mu_{sf}$$

**Hipótesis alternativa:** el promedio de líneas de código generadas con el framework es menor o igual que el promedio de líneas de código sin usar el framework.

$$H_1 = \mu_{cf} \leq \mu_{sf}$$

- 2) El nivel de significancia es de 0.05
- 3) Determinar la región crítica

gl = grados de libertad

$$gl = (n1 - 1) + (n2 - 1) = 6$$

n1 = número de datos con framework

n2 = número de datos sin framework

T – Tabulación

$$t_t(0.05 ; 6) = 1.94$$

$$t_t(0.05 ; 6) = > 1.94$$

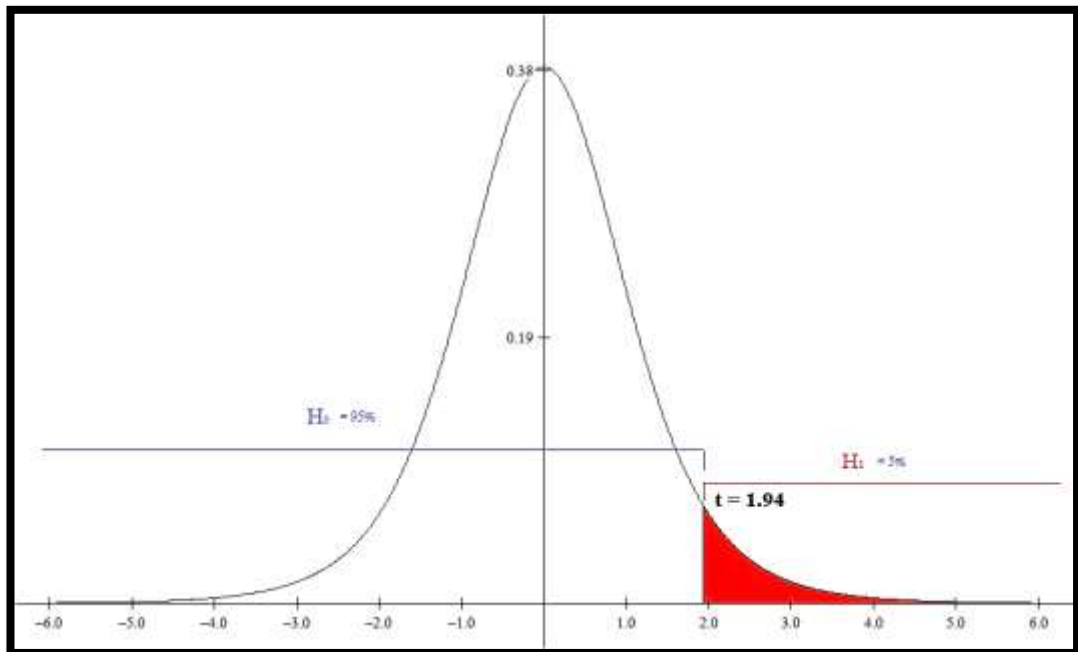


Figura 83. Gráfica de T – Student  
Elaborado por: Juan C. Moyota

#### 4) Cálculos

En la **Tabla 46** se observa que la varianza en ambos grupos de datos es igual por lo tanto se aplicará una **Prueba t para dos muestras suponiendo varianzas iguales**

	<b>Con framework</b>	<b>Sin framework</b>
Media	483.5	614.5
Varianza	1.66666667	1.66666667
Observaciones	4	4
Varianza agrupada	1.66666667	
Diferencia hipotética de las medias	0	
Grados de libertad	6	
Estadístico t	143.50331	
P(T<=t) una cola	3.8616E-12	
Valor crítico de t (una cola)	1.94318028	

*Tabla 46. Resultados de la tabla T-Student  
Elaborado por: Juan C. Moyota*

### **5.6.2. DECISIÓN**

Se rechaza  $H_0$  por cuanto el valor de  $P(T \leq t) = 3.8616E-12$  es menor que 0.05 es decir el promedio de líneas de código generadas con el framework es menor que el promedio de líneas de código sin usar el framework por lo mismo se acepta la  $H_1$ .

## **CAPÍTULO VI**

### **RESULTADOS**

Existen muchos framework PHP opensource pero podemos señalar una lista de tres de los más utilizados según sitios web como PHP Frameworks y la empresa Zfort Group consultora con sede en Ucrania líder en soluciones de ciclo completo de desarrollo web y especialista en montaje de TI, emplea tecnologías para la web como PHP, ASP.NET, AJAX, HTML5, ANDROID, etc.

Para el análisis se distinguieron los siguientes framework:

- Yii
- CakePHP
- CodeIgniter

Estos frameworks fueron seleccionados por razones que se mencionan a continuación:

- Existencia de documentación en línea, libros, y capacitación en canales como YouTube facilitan el aprendizaje y dominio del framework.
- Soporte para conexión con SGBD como MySQL, PostgreSQL, Oracle, Microsoft SQL, etc.
- Posibilidad de incorporar módulos, plugins y extensiones para aumentar el alcance de las funcionalidades del framework.

Al aplicar la metodología de evaluación, calificación y selección de herramientas de software libre llamada QSOS a los tres frameworks seleccionados se obtuvo los siguientes resultados en la calificación de los indicadores:

Framework	Porcentaje
<i>Yii</i>	92.50%
<i>CakePHP</i>	88.13%
<i>CodeIgniter</i>	89.38%

*Tabla 47. Porcentaje de frameworks PHP  
Elaborado por: Juan Carlos Moyota*

Con los resultados obtenidos se eligió Yii para desarrollar el módulo de gestión de reportes estadísticos por cuanto cumple en un 92.5% los requisitos técnicos, funcionales y estratégicos, mediante la clasificación y comparación de las tres herramientas framework.

La métrica de estimación del software con el método COCOMO se basa en las líneas de código fuente LDC para calcular el tiempo expresado en meses y el esfuerzo expresado en personas-mes que se necesita para desarrollar el software.

Se aplicó el método COCOMO con el uso de líneas de código necesarias en un pequeño módulo de generación de reportes gráficos desarrollados uno con el Framework Yii y otro desarrollado sin la utilización del framework. Al obtener los resultados de los cálculos realizados con las fórmulas establecidas por el método COCOMO básico se obtuvieron los siguientes resultados:

- Se necesita 1.12 personas para programar 485 LDC en 459.2 horas (2.61 meses) para la aplicación desarrollada con Yii a un costo aproximado de \$ 2088 dólares.
- La aplicación desarrollada sin el framework con 615 LDC se puede programar en 459.2 horas (2.87 meses) con 1.44 personas-mes y un costo aproximado de \$ 2296 dólares.

Las pruebas de aceptación del análisis de la distribución T-Student dieron como resultado que el promedio de líneas de código de la aplicación utilizando el framework es menor que el promedio de líneas de código de la aplicación sin usar el framework por lo tanto se aprueba la hipótesis de investigación.

## **CAPÍTULO VII**

### **DISCUSIÓN**

Esta investigación enfoca la utilización de Framework (Marco de Trabajo) para el lenguaje de programación web PHP. Entre los lenguajes de programación dinámicos para la web están: JSP, PHP, ASP, Ruby, etc. Para esta investigación se escogió utilizar el lenguaje del lado del servidor PHP por la facilidad de aprendizaje de su sintaxis de codificación, con la posibilidad de embeberse con código HTML y utilizar tecnologías como JavaScript y JQuery para desarrollar aplicaciones web interactivas.

Al aplicar la metodología de selección de productos de software libre QSOS a tres de los framework PHP opensource: CodeIgniter, CakePHP y Yii, Yii con un porcentaje del 92.5% de cumplimiento en los requisitos técnicos, funcionales y estratégicos, resaltó como apta para utilizar en el desarrollo del módulo de gestión de reportes estadísticos aplicado al sistema de agua potable de la provincia de Chimborazo.

El uso el modelo de estimación de software COCOMO, se afirma que al utilizar un framework PHP opensource se mejora el desarrollo en el ahorro de las líneas de código necesarias para realizar funciones específicas, con el uso de Yii se necesitaron 485 líneas de código que se generaron con el uso una herramienta generadora de código que incorpora el framework adaptándose a las necesidades del programador. Al contar con un menor número de líneas de código se necesita un menor tiempo de desarrollo a un menor costo de desarrollo del software y una mayor rentabilidad para la empresa desarrolladora del presupuesto general del proyecto.

## CAPÍTULO VIII

### CONCLUSIONES Y RECOMENDACIONES

#### 8.1. CONCLUSIONES

- Al analizar los tres frameworks PHP con el uso de los indicadores de la metodología QSOS (calificación y selección de herramientas de software libre) se establece que Yii en un 92.5% de los requisitos técnicos, funcionales y estratégicos es apto para desarrollar una aplicación web para gestión de reportes estadísticos.
- Se desarrolló la aplicación web para generación de reportes estadísticos utilizando Yii Framework al cumplir en un 92.5% de los requisitos técnicos, funcionales y estratégicos.
- Con Yii framework se utilizan 485 líneas de código para generar un reporte que a diferencia de una aplicación sin el uso del framework utiliza 615 líneas de código con un ahorro de 130 líneas de código con un ahorro de recursos del servidor como la memoria ocupada por el uso de variables.
- Los frameworks PHP cuentan con componentes adicionales que extienden las funcionalidades para los cuales fueron creados, esta característica podría ser un aspecto importante al momento de elegir el framework adecuado para desarrollar aplicaciones web.



## 8.2. RECOMENDACIONES

- Antes de utilizar un framework PHP para desarrollo de aplicaciones web identificar las características que hacen que este marco de trabajo sea apto para el desarrollo y cumplimiento de los objetivos del sistema.
- En base a la facilidad de uso y aprendizaje que brinda Yii utilizar este potente framework para el desarrollo de proyectos web por contar con mucha información de ayuda, soporte y documentación existiendo un ahorro de recursos del servidor al interpretar una menor cantidad de líneas de código fuente.
- Para el desarrollo de aplicaciones web utilizar herramientas que ayuden a disminuir las líneas de código que procesa el servidor de esta forma se reducen los recursos que se utilizan al procesar el código fuente.
- Utilizar de preferencia extensiones o componentes propios de los frameworks para desarrollar la aplicación por cuanto los componentes de terceros pueden no funcionar correctamente y emitir errores o resultados inesperados.
- Utilizar herramientas de software libre en proyectos de desarrollo web ya que cuentan con mucha información de ayuda para su utilización además que al estar internet no existirían problemas legales con el uso de licencias de uso reduciendo el costo de desarrollo del proyecto de esta manera existiría una mayor rentabilidad para la empresa de desarrollo.

## IX. BIBLIOGRAFÍA

- Alonso, S., Tejedor, J., & Gil, Y. (2006). *Creación de sitios web con PHP5*. Madrid: McGraw-Hill.
- Cabezas Granado, L. M. (2005). *Manual Imprescindible de PHP 5*. Madrid: EDICIONES ANAYA MULTIMEDIA (GRUPO ANAYA, S.A.).
- Calderón, R., & Tierra, C. TRABAJO DE GRADUACIÓN. *ESTUDIO DE TECNOLOGÍAS OPEN SOURCE Y SU INCIDENCIA EN EL COSTO DEL DESARROLLO DE APLICACIONES WEB. CASO APLICATIVO: SISTEMA DEFACTURACIÓN EN LA EMPRESA NACHOS SPORT*. Universidad Nacional de Chimborazo, Riobamba. Obtenido de [dspace.unach.edu.ec](http://dspace.unach.edu.ec)
- CodeIgniter. (2012). *Guía de usuario de CodeIgniter*. Recuperado el 25 de Abril de 2014, de EllisLab, Inc.: [http://escodeigniter.com/guia\\_usuario/](http://escodeigniter.com/guia_usuario/)
- Digiovannini, M. (Febrero de 2007). Comparativa de frameworks web. Buenos Aires, Argentina. Obtenido de JavaHispano.org: [http://www.javahispano.org/storage/contenidos/frameworks\\_web.pdf](http://www.javahispano.org/storage/contenidos/frameworks_web.pdf)
- Ferrer, J. (2012). *Implementación de aplicaciones web - CFGS*. Madrid, España: RA-MA Editorial.
- Gutierrez, A., Tejada, D., Silveira, E., & Caballero, C. (03 de Enero de 2013). *Libro de Kumbia: Porque Programar debería ser más fácil*. Obtenido de <http://www.detodoprogramacion.com/>
- Luján, S. (2005). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario.
- Manuel, D. (2006). *Programación en PHP*. Sevilla, Sevilla, España.
- Martinez, G., Camacho, G., & Daniel, B. (2010). Diseño de framework web para el desarrollo dinámico de aplicaciones web. *Scientia et Technica*, 44, 181. Recuperado el 21 de Enero de 2014, de <http://revistas.utp.edu.co/index.php/revistaciencia/article/viewFile/1817/1127>
- McArthur, K. (2008). *PRO PHP: Patterns, Frameworks, Testing and More*. Manhattan, Nueva York, USA: Editorial Board.
- Merkel, D. (2010). *Expert PHP 5 Tools*. Olton, Reino Unido: Packt Publishing Ltd.
- Niño, J. (2010). *Aplicaciones web*. Madrid, España: Editex.

- PHP Frameworks. (s.f.). *PHP Frameworks*. Recuperado el 22 de Abril de 2014, de PHP Frameworks: <http://www.phpframeworks.com/>
- Pressman, R. (2010). *Ingeniería del software un enfoque práctico*. Madrid: McGraw-Hill.
- Welling , L., & Thomson , L. (2005). *Desarrollo web con PHP y MySQL*. Madrid : Ediciones Anaya Multimedia .
- Winesett, J. (2010). *Agile Web Application Development with Yii 1.1* (Primera ed.). Olton, Ucrania: Packt Publishing Ltd. Recuperado el 14 de Mayo de 2014
- Yii Software LLC. (21 de 11 de 2008). *Yii framework*. Recuperado el 12 de Mayo de 2014, de Yii framework: <http://yiiframework.com>
- Zabala, X., & Ochoa, C. Estudio de Frameworks PHP e integración a una IDE aplicado al portal web de la comunidad Linux de la ESPOCH. *Tesis de grado*. Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.

## X. APÉNDICES O ANEXOS

### 10.1. GLOSARIO DE TÉRMINOS Y ABREVIATURAS

**PHP:** PHP Hypertext Preprocessor, lenguaje de programación para la web basado en lenguajes como C o C++.

**JSP:** (Java Server Pages), páginas de servidor compiladas en Java.

**ASP:** (Active Server Pages), tecnología de Microsoft para la creación de páginas web dinámicas.

**SGBD:** siglas de sistema gestor de base de datos.

**Ajax:** acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), tecnología web del lado del cliente para interactuar con el servidor asincrónicamente.

**EDP:** siglas de programación dirigida por eventos.

**ORM:** siglas de mapeo objeto-relacional, es un modelo de programación que consiste en la transformación de las tablas de una base de datos, en una serie de entidades que simplifiquen las tareas básicas de acceso a los datos para el programador.

**GAD:** Gobierno Autónomo Descentralizado

**ERS:** Especificación de requisitos de software, documento donde se especifican concretamente los requisitos de un sistema en fase de análisis.

**POO:** siglas programación orientada a objetos.

**LDC:** líneas de código

**UNACH:** Universidad Nacional de Chimborazo.



### 10.3. CÓDIGO FUENTE DE LAS APLICACIONES PROTOTIPO

#### Código fuente de la aplicación WebApp 1

La WebApp 1 está programada con el framework Yii a continuación se muestra el código fuente de la vista, controlador y modelo que se utilizaron para la programación del módulo de prueba:

#### Controlador

```
<?php
```

```
/**controlador para graficos de parametros de calidad del agua. Autor: JUAN CARLOS MOYOTA*/
```

```
class ParametroController extends Controller {
    public function actionIndex() {
        $model = new Muestras();
        if(isset($_POST["Muestras"])){
            $idcanton = (int)$_POST["Muestras"]["codcanton"];
            $idsistema = (int)$_POST["Muestras"]["codsistema"];
            $parametro = (string)$_POST["Muestras"]["parametro"];
            $titulo = nombre_parametro($parametro);
            $tabla_nombre = tabla_muestras($idcanton);
            $nombre_sistema = nombresistema($idsistema);
            $lim = getLimitePermisible($parametro);
            $unidad = getUnidadesParametro($parametro);
            $regi = array();$fecha = array();$fuente = array();$lim_per
= array();
            $reg = $model->
>getDatosParamBySistema($tabla_nombre,$idsistema,$parametro);
            for($i=0;$i<count($reg);$i++)
            {
                $regi[]=(float)$reg[$i]["$parametro"];
                $fecha[]=$reg[$i]["fecha"]."<br>".$reg[$i]["hora"];
                $fuente[]=$reg[$i]["fuente"];
                $lim_per[] = $lim;
            }
            $datos =
array('tabla'=>$tabla_nombre,'idsistema'=>$idsistema,
        'parametro'=>$titulo,'regi'=>$regi,
        'mes'=>$fecha,'nombre_sistema'=>$nombre_sistema,
        'fuente'=>$fuente,'lim'=>$lim_per,'unidad'=>$unidad
        );
            $this->render("graficaParametro", $datos);
        }else{ $this->render("index",compact("model")); }
    }
}
```

```

public function actionGraficaParametro(){
    $this->render("graficaParametro");
}

public function actionGraficaFecha() {
    $this->render("graficaFecha");
}

public function actionListaParroquias()
{
    $id_uno = $_POST['Muestras']['codcanton'];
    //$criteria = new CDbCriteria();
    //$criteria->condition
    $lista = Parroquias::model()->findAll('codcanton = :id order
by nombreparroquia ASC',array(':id'=>$id_uno));
    $lista =
CHtml::listData($lista,'codparroquia','nombreparroquia');

    echo CHtml::tag('option', array('value' => ''), '--
Seleccionar parroquia--', true);

    foreach ($lista as $valor => $descripcion){
        echo
CHtml::tag('option',array('value'=>$valor),CHtml::encode($descripcion),
true );
    }
}

public function actionListaBarrios()
{
    $id_dos = $_POST['Muestras']['codparroquia'];
    $lista = Barrios::model()->findAll('codparroquia = :id order
by nombrebarrio ASC',array(':id'=>$id_dos));
    $lista = CHtml::listData($lista,'codbarrio','nombrebarrio');
    echo CHtml::tag('option', array('value' => ''), '--
Seleccionar barrio--', true);
    foreach ($lista as $valor => $descripcion){
        echo
CHtml::tag('option',array('value'=>$valor),CHtml::encode($descripcion),
true );
    }
}

public function actionListaSistemas()
{
    $id_tres = $_POST['Muestras']['codbarrio'];
    $lista = Sistemas::model()->findAll('codbarrio = :id order
by nombresistema ASC',array(':id'=>$id_tres));
    $lista =
CHtml::listData($lista,'codsistema','nombresistema');
    echo CHtml::tag('option', array('value' => ''), '--
Seleccionar sistema--', true);
    foreach ($lista as $valor => $descripcion){

```

```

                echo
CHtml::tag('option',array('value'=>$valor),CHtml::encode($descripcion),
true );
        }
    }
}

```

## Modelo

```

<?php
class Muestras extends CActiveRecord{

    public function __construct() {
        $this->con = new CDbConnection(Yii::app()->db-
>connectionString,Yii::app()->db->username,Yii::app()->db->password);
        $this->con->active=true;
    }
    public static function model($className = __CLASS__) {
        parent::model($className);
    }

    public function
getDatosParamBySistema($nombre_tabla,$idsistema,$parametro){
        $sql = "select $parametro,fecha,hora,fuente from $nombre_tabla
            where (codsistema='$idsistema') and $parametro!='' ORDER
BY fecha ASC,hora ASC;";
        $datos = $this->con->createCommand($sql)->queryAll();
        $this->con->active=false;
        return $datos;
    }
}

```

## Vista formulario de consulta

```

<?php
$this->widget('application.extensions.jui.EJqueryUiInclude',
            array('theme'=>'ui-lightness'));
$this->pageTitle=Yii::app()->name . ' - Gráfica por parámetro';
$this->breadcrumbs=array(
    'Gráfica por parámetro',
);
?>
<h3>Gráfica de parámetros</h3>
<?php if (Yii::app()->user->hasFlash('mensaje')) { ?>
<div class="alert alert-info"><?php echo Yii::app()->user-
>getFlash('mensaje') ?></div>
<?php } ?>
<?php $this-
>beginWidget('application.extensions.jui.ETabs',array('name'=>'tabs'));
?>
<?php $this->beginWidget('application.extensions.jui.ETab',
    array('name'=>'tab1','title'=>'Datos de consulta')) ?>

```



```

<?php echo
CHtml::beginForm('', 'post', array("name"=>"muestras", "id"=>"muestras"));
?>
<div class="row-fluid">
    <div class="alert alert-success span4 text-center">
        <?php
            echo CHtml::activeDropDownList($model, 'codcanton',
                CHtml::listData(Cantones::model()->findAll(),
'codcanton', 'nombrecanton'),
            array('ajax'=>array(
                'type'=>'POST',
                'url'=>
CController::createUrl('parametro/ListaParroquias'),
'update'=>'#.CHtml::activeId($model, 'codparroquia'),
                'beforeSend' => 'function(){
$("#Muestras_codparroquia").find("option").remove();
$("#Muestras_codbarrio").find("option").remove();
$("#Muestras_codsistema").find("option").remove();
                }'
            ), "prompt"=>"--Seleccionar cantón--",
            "class"=>"input-xlarge"));
        ?>
        <?php echo CHtml::error($model, 'codcanton'); ?>
        <br>
        <?php
            $lista_dos = array();
            if(isset($model->codparroquia)){
                $id_uno = (int)($model->codcanton);
                $lista_dos = CHtml::listData(Parroquias::model()-
>findAll("codcanton = '$id_uno'"), 'codparroquia', 'nombreparroquia');
            }
            echo CHtml::activeDropDownList($model, 'codparroquia',
                $lista_dos,
                array(
                    'ajax'=>array(
                        'type'=>'POST',
                        'url'=>
CController::createUrl('parametro/ListaBarrios'),
                        'update'=>'#.CHtml::activeId($model, 'codbarrio'),
                        'beforeSend' => 'function(){
$("#Muestras_codbarrio").find("option").remove();
$("#Muestras_codsistema").find("option").remove();
                    }'
                ),
                "prompt"=>"--Seleccionar parroquia--",
                "class"=>"input-xlarge"
            ));
            echo CHtml::error($model, 'codparroquia');
        ?>

```

```

        <br>
        <?php
        $lista_tres = array();
        if(isset($model->codbarrio)){
        $id_dos = (int)($model->codparroquia);
        $lista_tres = CHtml::listData(Barrios::model()-
>findAll("codparroquia = '$id_dos'"), 'codbarrio', 'nombrebarrio');
        }
        echo
        CHtml::activeDropDownList($model, 'codbarrio', $lista_tres,
        array(
            'ajax'=>array(
                'type'=>'POST',
                'url'=>
        CController::createUrl('parametro/ListaSistemas'),
        'update'=>'#.CHtml::activeId($model, 'codsistema'),
            'beforeSend'=>'function(){
        $("#Muestras_codsistema").find("option").remove();
            }'
        ),
            "prompt"=>"--Seleccionar barrio--",
            "class"=>"input-xlarge"
        ));
        echo CHtml::error($model, 'codbarrio');
        ?>
        <br>
        <?php
        $lista_cuatro = array();
        if(isset($model->codsistema)){
            $id_tres=(int)($model->codbarrio);
            $lista_cuatro= CHtml::listBox(Sistemas::model()-
>findAll("codbarrio = '$id_tres'"), 'codsistema', 'nombresistema');
        }
        echo CHtml::activeDropDownList($model, 'codsistema',
        $lista_cuatro,
        array(
            "prompt"=>"--Seleccionar sistema--",
            "class"=>"input-xlarge"
        )
        );
        echo CHtml::error($model, 'codsistema');
        ?>
    </div>
    <div class="alert alert-info span4 text-center">
    <?php echo CHtml::activeDropDownList($model, 'parametro',
        array(
            'Características físicas'=>array(
                'ph'=>'pH',
                'color'=>'Color',
                'turbiedad'=>'Turbiedad',
                'temperatura'=>'Temperatura',
                'solidos_totales'=>'Sólidos totales disueltos',
                'conductividad'=>'Conductividad'
            )
        )
    );
    </div>

```

```

    ),
    'Características químicas'=>array(
        'dureza'=>'Dureza',
        'cloro_libre'=>'Cloro Libre residual',
        'hierro'=>'Hierro',
        'nitratos'=>'Nitratos',
        'nitritos'=>'Nitritos',
        'sulfatos'=>'Sulfatos',
        'fosfatos'=>'Fosfatos',
        'manganeso'=>'Manganeso',
        'fluoruros'=>'Fluoruros',
        'amoniac'=>'Amoniac'
    ),
    'Requisitos microbiológicos'=>array(
        'coliformes_fecales'=>'Coliformes fecales',
        'coliformes_totales'=>'Coliformes totales'
    )
),
array("prompt"=>"Seleccione un parámetro","class"=>"input-
xlarge"))
?>
</div>
<div class="alert alert-info span4 text-center">
<?php echo
CHtml::activeTextField($model,'fecha',array("placeholder"=>"Fecha
inicial","readonly"=>"readonly"));
echo CHtml::error($model,'fecha');
?><img src = "<?php echo Yii::app()->request->baseUr1
?>/images/calendar.gif" />
<br>
<?php echo
CHtml::activeTextField($model,'fecha2',array("placeholder"=>"Fecha
final","readonly"=>"readonly"));
echo CHtml::error($model,'fecha2');
?><img src = "<?php echo Yii::app()->request->baseUr1
?>/images/calendar.gif" />
</div>
</div>
<div class="container text-center">
<?php
echo CHtml::submitButton("submit",array("value"=>"Generar
gráfica","title"=>"Generar gráfica","class"=>"btn btn-success"));
?>
<script type="text/javascript">
$(document).ready(function(){
    $("#muestras").validate({
        rules:{
            'Muestras[codcanton]': 'required',
            'Muestras[codbarrio]': 'required',
            'Muestras[codparroquia]': 'required',
            'Muestras[codsistema]': 'required',
            'Muestras[parametro]': 'required'
        }
    });
});
});

```

```

        </script>
    </div>
<?php echo CHtml::endForm(); ?>
<?php $this->endWidget('application.extensions.jui.ETab'); ?>
<?php $this->endWidget('application.extensions.jui.ETabs');

```

## Vista del reporte gráfico

```

<?php
$this-
>widget('application.extensions.jui.EjQueryUiInclude',array('theme'=>'ui-
-lightness'));
$this->pageTitle=Yii::app()->name . ' - Gráfica del parámetro';$this-
>breadcrumbs=array('Gráfica del parámetro',);
$titulo = "Gráfico de niveles de $parametro de $nombre_sistema";
$titulo = strtoupper($titulo);
?><br>
<?php $this-
>beginWidget('application.extensions.jui.ETabs',array('name'=>'tabs'));
?>
<?php $this-
>beginWidget('application.extensions.jui.ETab',array('name'=>'tab1','tit
le'=>'Gráfico')) ?>
<?php if (count($regi)>0){ ?>
    <?php
$this->Widget('ext.highcharts.HighchartsWidget', array(
    'scripts' => array(
        'highcharts-more',
        'modules/exporting',
        'themes/sand'
    ),
    'options'=>array(
        'chart' => array(
            'plotBackgroundColor' => '#cccccc',
            'type'=>'line',
        ),
        'title' => array('text' => $titulo),
        'subtitle' => array('text' => "Fuente: <strong>SENAGUA</strong>
"),
        'xAxis' => array(
            'categories' => $mes
        ),
        'yAxis' => array(
            'title' => array('text' =>"NIVELES DE ".$parametro."
($unidad)"),
        ),
        'plotOptions'=>array(
            'line'=>array(
                'enableMouseTracking'=>true,
                'dataLabels'=>array(
                    'enabled'=>true,
                ),
            ),
        ),
    ),
    'colors'=>array("#0099ff", "#ff0033"),

```

```

        'series' => array(
            array('name' => $parametro, 'data' =>
                $regi, 'marker' => array('symbol' => 'diamond', 'radius' => 4)),
            array('name' => "Limite permisible $lim[0]", 'data' => $lim,
                'dashStyle' => 'shortdot', 'marker' => array('enabled' => false),
                'dataLabels' => array('enabled' => false)
            )
        )
    )
));
?>
<p class="text-right"><small>Limite permisible para <?php echo
strtolower($parametro) ?> es <strong><?php echo $lim[0]. " ".$unidad;
?></strong></small></p>
<?php
$n = count($regi);
if($regi[$n-1]>$lim[0])
{
    $indicador = strtolower($parametro);
    echo "<div class='alert alert-error'>Según el último resultado, el
parámetro <strong>$indicador</strong> no está dentro de los límite
permisible <strong>$lim[0]</strong></div>";
}
?>
<?php
}else{
    echo "<h3>NO SE HAN ENCONTRADO REGISTROS</h3>";
}
?>
<?php $this->endWidget('application.extensions.jui.ETab'); ?>
<?php $this-
>beginWidget('application.extensions.jui.ETab', array('name' => 'tab2', 'tit
le' => 'Datos')) ?>
<?php if (count($regi)>0) { ?>
<div class="">
    <script language="javascript" >
        $(document).ready(function(){
            $("#tb_resultados").dataTable({
                "sPaginationType": "full_numbers",
                "bJQueryUI": true,
                "bProcessing": true,
                "oLanguage": {
                    "sLengthMenu": "Ver _MENU_ registros",
                    "sInfo": "Resultados _START_ - _END_ de
TOTAL_ registros",
                    "sSearch": "Buscar: ",
                    "sInfoFiltered": "(Filtrado de _MAX_
registros)",
                    "oPaginate": {
                        "sFirst": "Primero",
                        "sLast": "Último",
                        "sNext": "Siguiente",
                        "sPrevious": "Anterior"
                    }
                }
            }
        }
    )
}
}

```

```

        });
    });
</script>
<div class="container text-center"><h3><?php echo
$nombre_sistema ?></h3></div>
<table class="table table-striped table-bordered"
id="tb_resultados">
    <thead>
        <tr>
            <th>Sitio de toma de muestra</th>
            <th>Fecha y hora de recolecci&oacute;n</th>
            <th>Resultado</th>
        </tr>
    </thead>
    <tbody>
        <?php for($i=0;$i<count($regi);$i++) { ?>
            <tr>
                <td><?php echo $fuente[$i] ?></td>
                <td><?php echo $mes[$i] ?></td>
                <td><?php echo $regi[$i] ?></td>
            </tr>
        <?php } ?>
    </tbody>
</table>
</div>
<?php }else{ echo "<h3>NO SE HAN ENCONTRADO REGISTROS</h3>"; }?>
<?php $this->endWidget('application.extensions.jui.ETab'); ?>
<?php $this->endWidget('application.extensions.jui.ETabs'); ?><br>
<?php echo CHtml::link("Atras", Yii::app()->request-
>baseUrl."/parametro/index",array("class"=>"label label-success")); ?>

```

## Funciones

```

function nombre_parametro($param){
    $parametro = strtoupper($param);
    $parametro = strstr($parametro,"_", " ");
    return $parametro;
}
function nombre_parametro($param){
    $parametro = strtoupper($param);
    $parametro = strstr($parametro,"_", " ");
    return $parametro;
}
function nombresistema($param)
{
    $criteria = new CDbCriteria();
    $criteria->select='nombresistema';
    $criteria->condition='codsistema=:id';
    $criteria->params=array(':id'=>$param);
    $nombre = Sistemas::model()->find($criteria);
    return $nombre["nombresistema"];
}
function getLimitePermisible($param)
{

```

```

$lim_sup = array();
if($param == "ph")
{
    $lim_sup = 9;
}
if($param == "color")
{
    $lim_sup = 15;
}
if($param == "turbiedad")
{
    $lim_sup = 5;
}
if($param == "temperatura")
{
    $lim_sup = "";
}
if($param == "solidos_totales")
{
    $lim_sup = 500;
}
if($param == "conductividad")
{
    $lim_sup = "";
}
if($param == "dureza")
{
    $lim_sup = 500;
}
if($param == "cloro_libre")
{
    $lim_sup = 1.5;
}
if($param == "hierro")
{
    $lim_sup = 0.3;
}
if($param == "nitritos")
{
    $lim_sup = 3;
}
if($param == "nitratos")
{
    $lim_sup = 50;
}
if($param == "sulfatos")
{
    $lim_sup = 250;
}
if($param == "fosfatos")
{
    $lim_sup = 0.3;
}
if($param == "manganeso")
{

```

```

        $lim_sup = 0.1;
    }
    if($param == "fluoruros")
    {
        $lim_sup = 1.5;
    }
    if($param == "amoniaco")
    {
        $lim_sup = 1;
    }
    if($param == "coliformes_fecales")
    {
        $lim_sup = 50;
    }
    if($param == "coliformes_totales")
    {
        $lim_sup = 3000;
    }
    return $lim_sup;
}
function getUnidadesParametro($param)
{
    unset($unidad);
    $unidad[] = array();
    if($param == "ph")
    {
        $unidad = "unidades";
    }
    if($param == "color")
    {
        $unidad = "color real";
    }
    if($param == "turbiedad")
    {
        $unidad = "NTU";
    }
    if($param == "temperatura")
    {
        $unidad = "°C";
    }
    if($param == "solidos_totales")
    {
        $unidad = "mg/l";
    }
    if($param == "conductividad")
    {
        $unidad = "µS/cm";
    }
    if($param == "dureza")
    {
        $unidad = "mg/l";
    }
    if($param == "cloro_libre")
    {
        $unidad = "mg/l";
    }
}

```



```

    }
    if($param == "hierro")
    {
        $unidad = "mg/l";
    }
    if($param == "nitritos")
    {
        $unidad = "mg/l";
    }
    if($param == "nitratos")
    {
        $unidad = "mg/l";
    }
    if($param == "sulfatos")
    {
        $unidad = "mg/l";
    }
    if($param == "fosfatos")
    {
        $unidad = "mg/l";
    }
    if($param == "manganeso")
    {
        $unidad = "mg/l";
    }
    if($param == "fluoruros")
    {
        $unidad = "mg/l";
    }
    if($param == "amoniaco")
    {
        $unidad = "mg/l";
    }
    if($param == "coliformes_fecales")
    {
        $unidad = "NMP/100 ml";
    }
    if($param == "coliformes_totales")
    {
        $unidad = "NMP/100 ml";
    }
    return $unidad;
}

```

### **Código fuente de la aplicación WebApp 2**

La aplicación WebApp 2 está desarrollada sin la utilización del framework, a continuación se presenta el código fuente del formulario de consulta, reporte gráfico y las funciones utilizadas para la construcción del módulo de prueba.

## Código fuente del formulario de consulta

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Genera graficas con los parámetros</title>
<link href="css/default.css" rel="stylesheet" type="text/css"
media="all" />
<link href="../themes/ui-lightness/jquery-ui-1.10.3.custom.css"
rel="stylesheet" type="text/css" media="all" />
<link href="../css/bootstrap.css" rel="stylesheet" type="text/css"
media="all" />
<link href="../css/calendario.css" type="text/css" rel="stylesheet" />
<script type="text/javascript" src="../js/jquery-
1.10.2.min.js"></script>
<script type="text/javascript"
src="../js/jquery.validate.min.js"></script>
<script type="text/javascript" src="../js/messages_es.js"></script>
<script type="text/javascript" src="../js/jquery-ui.js"></script>
<script type="text/javascript" src="js/combos.js"></script>
<script type="text/javascript">
    $(document).ready(function(){
        $("#grafico,#procesar,#recargar").hide();
        $("#three-columns").attr("width","auto");
        $("#tabb").tabs({
            collapsible: false
        });
        $("#pestanias").tabs({
            collapsible: false
        });
        $("#datos").validate({
            rules:{
                'canton':'required',
                'parroquia':'required',
                'barrio':'required',
                'sistema':'required',
                'parametro':'required',
                'fecha_inicial':'required',
                'fecha_final':'required'
            },
            submitHandler:function(){
                $.ajaxStart(function(){
                    $('#procesar').show();
                }).ajaxStop(function(){
                    $('#procesar').hide();
                    $('#grafico').fadeIn('slow');
                    $('#canton option:selected').removeAttr("selected");
                    $('#parroquia option:selected').removeAttr("selected");
                });
                $.ajax({
                    beforeSend:function(){
                        $('#procesar').show();
                    },

```

```

        type: 'POST',
        url: "graf_parametro.php",
        data: $("#datos").serialize()+"&id=" + Math.random(),
        timeout:3000000,
        success: function(data) {
            $('#grafico,#recargar').fadeIn('slow');
            $('#procesar').hide();
            $('#grafico').html(data);
            $('#principal').hide();
        }
    });
    return false;
}
});
});
</script>
</head>
<body onload="document.form.reset()">
    <?php include("partes/header.php"); ?>
    <div class="content" id="principal">
        <div class="breadcrumbs">
            <a href="index.php">Home</a>>
            <span>Gráfica por parámetro</span>
        </div>
        <h3>Gráfica de parámetros</h3>
        <form name="datos" id="datos"
action="graf_parametro.php" method="POST">
            <div id="pestanias">
                <ul>
                    <li><a href="#tab1">Datos de
consulta</a></li>
                </ul>
                <div id="tab1">
                    <div class="row-fluid">
                        <div class="span4">
                            <h4>Ubicaci&oacute;n:</h4>
                            <div class="alert alert-success">
                                <label
for="canton">Cant&oacute;n:&nbsp;<select name="canton"
id="canton"></select></label>
                                <label
for="parroquia">Parroquia:&nbsp;<select name="parroquia" class="span8"
id="parroquia"></select></label>
                                <label for="barrio">Barrio:&nbsp;<select
name="barrio" id="barrio" ></select></label>
                                <label
for="sistema">Sistema:&nbsp;<select name="sistema"
id="sistema"></select></label>
                            </div>
                        </div>
                        <div class="span4">
                            <h4>Par&aacute;metro:</h4>
                            <div class="alert alert-success">
                                <h4>Seleccione un par&aacute;metro:</h4>
                                <select name="parametro" id="parametro">

```

```

                                <option value="">---Seleccionar---
</option>
                                <optgroup
label="Características físicas" ></optgroup>
                                <option value="ph">pH</option>
                                <option value="color">Color</option>
                                <option
value="turbiedad">Turbiedad</option>
                                <option
value="temperatura">Temperatura</option>
                                <option
value="solidos_totales">Sólidos totales disueltos</option>
                                <option
value="conductividad">Conductividad</option>
                                <optgroup
label="Características químicas" ></optgroup>
                                <option
value="dureza">Dureza</option>
                                <option value="cloro_libre">Cloro
libre residual</option>
                                <option
value="nitratos">Nitratos</option>
                                <option
value="nitritos">Nitritos</option>
                                <option
value="sulfatos">Sulfatos</option>
                                <option
value="fosfatos">Fosfatos</option>
                                <option
value="manganeso">Manganeso</option>
                                <option
value="fluoruros">Fluoruros</option>
                                <option value="amoniaco">Nitrógeno
amoniaco</option>
                                <optgroup label="Requisitos
microbiológicos" ></optgroup>
                                <option
value="coliformes_fecales">Coliformes fecales</option>
                                <option
value="coliformes_totales">Coliformes totales</option>
                                </select>
                                </div>
                                </div>
                                <div class="span4">
                                <h4>Fechas:</h4>
                                <div class="alert alert-success">
                                Fecha inicial:&nbsp;<input
name="fecha_inicial" type="text" disabled="disabled" class="input-small"
id="fecha_inicial" value="" /><i class="icon-calendar"
id="fecha1"></i><br />
                                <br>
                                Fecha final:&nbsp;<input name="fecha_final"
type="text" disabled="disabled" class="input-small" id="fecha_final"
value="" /><i class="icon-calendar" id="fecha2"></i>
                                </div>

```

```

        </div>
    </div>
    </div>
    <div class="container text-center">
        <button type="submit" value="Generar gráfica"
name="enviar" class="btn btn-success">Generar gráfica</button>
    </div>
    </div>
    </form>
</div>
    <div id="procesar" class="text-center">
        
    </div>
    <div id="grafico"></div><br>
    <div id="recargar" class="content text-center">
        <a class="label label-success"
onclick="window.location.reload();">Atras</a>
    </div>
</div>
    <?php include("partes/footer.php"); ?>
</body>
</html>

```

### Código fuente del reporte gráfico

```

<?php
error_reporting(E_ALL & ~E_NOTICE);
include_once '../clases/Sistema.php';
include_once '../conexion/Conexion.php';
include_once '../clases/nombre_canton.php';
include_once 'funciones/funciones_grafica1.php';
$id_canton = $_POST["canton"];
$id_sistema=$_POST["sistema"];
$indicador = $_POST["parametro"];
$sistema = new Sistema();
$dato_sistema = $sistema->nombre_by_id($id_sistema);
$nombre_sistema = $dato_sistema[0]["nombresistema"];
unset($sql,$muestra_canton);
$muestra_canton = canton($id_canton);
$sql = "select $indicador,fecha,hora,fuente from $muestra_canton
        where codsistema='$id_sistema' and $indicador!='' order by fecha
ASC;";
$resp = mysql_query($sql,Conexion::con());
$total = mysql_num_rows($resp);
$datos = array();
while ($reg = mysql_fetch_assoc($resp)) {
    $datos[] = $reg;
}
mysql_close();
$titulox = titulo_ejey($indicador);
$titulo = titulo_grafica($indicador);
$lim1 = getLimitePermisible($indicador);
?>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Gráfica</title>
<link type="text/css" href="../css/demo_table_jui.css" rel="stylesheet"
/>
<script type="text/javascript" src="../js/highcharts.js"></script>
<script type="text/javascript"
src="../js/modules/exporting.js"></script>
<script type="text/javascript"
src="../js/jquery.dataTables.js"></script>
<script type="text/javascript">
$(document).ready(function(){
    $("#datatables").dataTable({
        "bProcessing": true,
        "sPaginationType": "full_numbers",
        "bJQueryUI": true,
        "oLanguage": {
            "sLengthMenu": "Ver _MENU_ fichas",
            "sInfo": "Resultados _START_ - _END_ de _TOTAL_
fichas",
            "sSearch": "Buscar: ",
            "sInfoFiltered": "(Filtrado de _MAX_ fichas)",
            "sZeroRecords": "No se encontraron
resultados",
            "oPaginate": {
                "sFirst": "Primero",
                "sLast": "Último",
                "sNext": "Siguiente",
                "sPrevious": "Anterior"
            }
        }
    });
    $("#pestanias").tabs({
        collapsible: false
    });
    $('#grafica').highcharts({
        chart: {
            type: 'line'
        },
        title: {
            text: 'Grafico de niveles de <?php echo $titulo ?>'
        },
        subtitle: {
            text: 'Fuente: SENAGUA'
        },
        xAxis: {
            categories: [
                <?php for ($i=0;$i<$total;$i++) {
                    echo
                    "".$datos[$i][fecha]."/".$datos[$i][hora]."";
                    if ($i<$total-1){
                        echo ',';

```

```

        }
    }
    ?>
]
},
yAxis: {
    title: {
        text: '<?php echo $titulox ?>'
    }
},
plotOptions: {
    line: {
        dataLabels: {
            enabled: true
        },
        enableMouseTracking: true,
        'colors': ['#0099ff', '#ff0033', '#ff0033']
    }
},
series: [
{
    name: '<?php echo $titulo ?>',
    marker: {
        symbol: 'square'
    },
    data: [
        <?php for ($i=0;$i<$total;$i++) {
            echo $datos[$i]["$indicador"];
            if ($i<$total-1){
                echo ',';
            }
        } ?>
    ]
},
{
    name: '<?php echo "Limite permisible ".$lim1 ?>',
    marker: {
        enabled: false
    },
    dataLabels: {enabled: false},
    data: [
        <?php for ($i=0;$i<$total;$i++) {
            echo $lim1;
            if ($i<$total-1){
                echo ',';
            }
        } ?>
    ],
    dashStyle: 'shortdot',
    color: '#ff0033'
}
]
});
});

```

```

</script>
</head>
<body>
    <?php if (count($datos)>0) { ?>
        <div id="pestanian">
            <ul>
                <li><a href="#tab1">Gr&aacute;fico</a></li>
                <li><a href="#tab2">Datos</a></li>
            </ul>
            <div id="tab1">
                <div id="grafica" style="min-width: 900px; height: auto;
margin: 0 auto;">
            </div>
            </div>
            <div id="tab2">
                <h3><?php echo $nombre_sistema ?> <br> </h3>
                <table class="table table-bordered" id="datatables">
                    <thead>
                        <tr>
                            <th>Sitio de toma de muestra</th>
                            <th>Fecha/hora de recoleccion</th>
                            <th><?php echo $titulo ?></th>
                        </tr>
                    </thead>
                    <tbody>
                        <?php for ($i=0;$i<$total;$i++) { ?>
                            <tr>
                                <td><?php echo $datos[$i]["fuente"] ?></td>
                                <td><?php echo $datos[$i]["fecha"]." /
".$datos[$i]["hora"] ?></td>
                                <td><?php echo $datos[$i]["$indicador"]
?></td>
                            </tr>
                        <?php } ?>
                    </tbody>
                </table>
            </div>
        </div>
        <?php } else{ ?>
            <div class="container text-center alert alert-info">
                <h3>NO SE HAN ENCONTRADO DATOS</h3>
            </div>
        <?php } ?>
    </body>
</html>

```

### Código fuente de las funciones utilizadas

```

<?php
function titulo_ejey($parametro)
{
    $tituloy = "";
    if ($parametro == "ph"){
        $tituloy = "pH (unidades)";
    }
}

```



```

if ($parametro == "color"){
$tituloy = "Color (unidad de color)";
}
if ($parametro == "turbiedad"){
$tituloy = "Turbiedad (NTU)";
}
if ($parametro == "temperatura"){
$tituloy = "Temperatura (°C)";
}
if ($parametro == "solidos_totales"){
$tituloy = "Sólidos totales disueltos (mg/l)";
}
if ($parametro == "conductividad"){
$tituloy = "Conductividad (µS/cm)";
}
if ($parametro == "dureza"){
$tituloy = "Dureza (mg/l)";
}
if ($parametro == "cloro_libre"){
$tituloy = "Cloro libre residual (mg/l)";
}
if ($parametro == "nitratos"){
$tituloy = "Nitratos (mg/l)";
}
if ($parametro == "nitritos"){
$tituloy = "Nitritos (mg/l)";
}
if ($parametro == "sulfatos"){
$tituloy = "Sulfatos (mg/l)";
}
if ($parametro == "fosfatos"){
$tituloy = "Fosfatos (mg/l)";
}
if ($parametro == "manganeso"){
$tituloy = "Manganeso (mg/l)";
}
if ($parametro == "fluoruros"){
$tituloy = "Fluoruros (mg/l)";
}
if ($parametro == "amoniaco"){
$tituloy = "Nitrógeno amoniacal (mg/l)";
}
if ($parametro == "coliformes_fecales"){
$tituloy = "Coliformes fecales (NMP/100 ml)";
}
if ($parametro == "coliformes_totales"){
$tituloy = "Coliformes totales (NMP/100 ml)";
}
return $tituloy;
}
function titulo_grafica($parametro)
{
$tituloy = "";
if ($parametro == "ph"){
$tituloy = "pH";
}
}

```

```

}
if ($parametro == "color"){
$tituloy = "Color";
}
if ($parametro == "turbiedad"){
$tituloy = "Turbiedad";
}
if ($parametro == "temperatura"){
$tituloy = "Temperatura";
}
if ($parametro == "solidos_totales"){
$tituloy = "Sólidos totales disueltos";
}
if ($parametro == "conductividad"){
$tituloy = "Conductividad";
}
if ($parametro == "dureza"){
$tituloy = "Dureza";
}
if ($parametro == "cloro_libre"){
$tituloy = "Cloro libre residual";
}
if ($parametro == "nitratos"){
$tituloy = "Nitratos";
}
if ($parametro == "nitritos"){
$tituloy = "Nitritos";
}
if ($parametro == "sulfatos"){
$tituloy = "Sulfatos";
}
if ($parametro == "fosfatos"){
$tituloy = "Fosfatos";
}
if ($parametro == "manganeso"){
$tituloy = "Manganeso";
}
if ($parametro == "fluoruros"){
$tituloy = "Fluoruros";
}
if ($parametro == "amoniaco"){
$tituloy = "Nitrógeno amoniacal";
}
if ($parametro == "coliformes_fecales"){
$tituloy = "Coliformes fecales";
}
if ($parametro == "coliformes_totales"){
$tituloy = "Coliformes totales (NMP/100 ml)";
}
return $tituloy;
}

/* funcion para el limite permisible */
function getLimitePermisible($param)
{

```

```

if($param == "ph")
{
$lim_sup[0] = 6;

$lim_sup[1] = 9;
}
if($param == "color")
{
$lim_sup = 15;
}
if($param == "turbiedad")
{
$lim_sup = 5;
}
if($param == "temperatura")
{
$lim_sup = "";
}
if($param == "solidos_totales")
{
$lim_sup = 500;
}
if($param == "conductividad")
{
$lim_sup = "";
}
if($param == "dureza")
{
$lim_sup = 500;
}
if($param == "cloro_libre")
{
$lim_sup[0] = 0.3;
$lim_sup[1] = 1.5;
}
if($param == "hierro")
{
$lim_sup = 0.3;
}
if($param == "nitritos")
{
$lim_sup = 3;
}
if($param == "nitratos")
{
$lim_sup = 50;
}
if($param == "sulfatos")
{
$lim_sup = 250;
}
if($param == "fosfatos")
{
$lim_sup = 0.3;
}
}

```

```

if($param == "manganeso")
{
$lim_sup = 0.1;
}
if($param == "fluoruros")
{
$lim_sup = 1.5;
}
if($param == "amoniaco")
{
$lim_sup = 1;
}
if($param == "coliformes_fecales")
{
$lim_sup = 50;
}
if($param == "coliformes_totales")
{
$lim_sup = 3000;
}
return $lim_sup;
}

```

### **Código fuente de la conexión con la base de datos**

```

<?php
class Conexion
{
    public static function con()
    {
        $conexion = @mysql_connect("localhost","root","2321") or
die ("EXISTEN PROBLEMAS EN EL SERVIDOR.".mysql_error());
        mysql_query("SET NAMES 'utf8'");
        mysql_select_db("caliagua") or die ("EXISTEN PROBLEMAS CON
LA BASE DE DATOS.".mysql_error());
        return $conexion;
    }
}

```

### **Llenar los campos select**

```

<?php
include("../conexion/Conexion.php");
$Accion = $_REQUEST['Accion'];
if(is_callable($Accion)){
    $Accion();
}
/*170unción obtener cantones*/
function GetCantones() {
    header("Content-Type: application/json");
    $Cantones = array();
}

```

```

        $sql = "select codcanton,nombrecanton from cantones";
        $consulta = mysql_query($sql,€::con());
        mysql_close(Conexion::con());
        while ($Fila = mysql_fetch_assoc($consulta)) {
            $Cantones[]=$Fila;
        }
        echo json_encode($Cantones);
    }
    /*171unción obtener parroquias € el canton*/
    function GetParroquias() {
        header("Content-Type: application/json");
        $Parroquias=array();
        $query = "select codparroquia,nombreparroquia from parroquias where
codcanton="._REQUEST['codcanton'].” Order by nombreparroquia”;
        $consulta = mysql_query($query,€::con());
        mysql_close(Conexion::con());
        while ($Fila = mysql_fetch_assoc($consulta)){
            $Parroquias[]=$Fila;
        }
        echo json_encode($Parroquias);
    }
    /*171unción obtener barrios € el canton*/
    function GetBarrios() {
        header("Content-Type: application/json");
        $Barrios = array();
        $sql = "select codbarrio,nombrebarrio from barrios where
codparroquia="._REQUEST['codparroquia'].” Order by nombrebarrio”;
        $consulta = mysql_query($sql, €::con());
        mysql_close(Conexion::con());
        while ($Fila = mysql_fetch_assoc($consulta)){
            $Barrios[] = $Fila;
        }
        echo json_encode($Barrios);
    }
    /*171unción obtener sistemas € el barrio*/
    function GetSistemas() {
        header("Content-Type: application/json");
        $Sistemas = array();
        $sql = "select codsistema,nombresistema from sistemas where
codbarrio="._REQUEST['codbarrio'].” Order by nombresistema”;
        $consult = mysql_query($sql, Conexion::con());
        mysql_close(Conexion::con());
        while($Fila = mysql_fetch_assoc($consult))
        {
            $Sistemas[]=$Fila;
        }
        echo json_encode($Sistemas);
    }
}

```

## Clase Sistemas

```

<?php
include_once ("../conexion/Conexion.php");
class Sistema{

```

```

        private $listasistema;
public function nombre_by_id($id)
    {
        $sql[] = array();
        unset($sql);
        $sql = "select nombresistema from sistemas where
codsistema='$id'";
        $datos = mysql_query($sql,Conexion::con());
        mysql_close();
        while($dat = mysql_fetch_assoc($datos)){
            $this->listasistema[]=$dat;
        }
        mysql_free_result($datos);
        return $this->listasistema;
    }
}

```

### **Función para obtener el nombre de la tabla muestras**

```

<?php
function canton($idcanton) {
    $canton[] = array();
    switch ($idcanton){
        case 100:    $canton = "muestras_alausi";
                    break;
        case 101:    $canton = "muestras_chambo";
                    break;
        case 102:    $canton = "muestras_chunchi";
                    break;
        case 103:    $canton = "muestras_colta";
                    break;
        case 104:    $canton = "muestras_cumanda";
                    break;
        case 105:    $canton = "muestras_guamote";
                    break;
        case 106:    $canton = "muestras_guano";
                    break;
        case 107:    $canton = "muestras_pallatanga";
                    break;
        case 108:    $canton = "muestras_penipe";
                    break;
        case 109:    $canton = "muestras_riobamba";
                    break;
    }
    return $canton;
}

```

## Funciones adicionales

```
<?php
function titulo_eje($parametro)
{
    $tituloy = "";
    if ($parametro == "ph"){
        $tituloy = "pH (unidades)";
    }
    if ($parametro == "color"){
        $tituloy = "Color (unidad de color)";
    }
    if ($parametro == "turbiedad"){
        $tituloy = "Turbiedad (NTU)";
    }
    if ($parametro == "temperatura"){
        $tituloy = "Temperatura (°C)";
    }
    if ($parametro == "solidos_totales"){
        $tituloy = "Sólidos totales disueltos (mg/l)";
    }
    if ($parametro == "conductividad"){
        $tituloy = "Conductividad (µS/cm)";
    }
    if ($parametro == "dureza"){
        $tituloy = "Dureza (mg/l)";
    }
    if ($parametro == "cloro_libre"){
        $tituloy = "Cloro libre residual (mg/l)";
    }
    if ($parametro == "nitratos"){
        $tituloy = "Nitratos (mg/l)";
    }
    if ($parametro == "nitritos"){
        $tituloy = "Nitritos (mg/l)";
    }
    if ($parametro == "sulfatos"){
        $tituloy = "Sulfatos (mg/l)";
    }
    if ($parametro == "fosfatos"){
        $tituloy = "Fosfatos (mg/l)";
    }
    if ($parametro == "manganeso"){
        $tituloy = "Manganeso (mg/l)";
    }
    if ($parametro == "fluoruros"){
        $tituloy = "Fluoruros (mg/l)";
    }
    if ($parametro == "amoniaco"){
        $tituloy = "Nitrógeno amoniacal (mg/l)";
    }
    if ($parametro == "coliformes_fecales"){
        $tituloy = "Coliformes fecales (NMP/100 ml)";
    }
    if ($parametro == "coliformes_totales"){
```

```

        $tituloy = "Coliformes totales (NMP/100 ml)";
    }
    return $tituloy;
}
function titulo_grafica($parametro)
{
    $tituloy = "";
    if ($parametro == "ph"){
        $tituloy = "pH";
    }
    if ($parametro == "color"){
        $tituloy = "Color";
    }
    if ($parametro == "turbiedad"){
        $tituloy = "Turbiedad";
    }
    if ($parametro == "temperatura"){
        $tituloy = "Temperatura";
    }
    if ($parametro == "solidos_totales"){
        $tituloy = "Sólidos totales disueltos";
    }
    if ($parametro == "conductividad"){
        $tituloy = "Conductividad";
    }
    if ($parametro == "dureza"){
        $tituloy = "Dureza";
    }
    if ($parametro == "cloro_libre"){
        $tituloy = "Cloro libre residual";
    }
    if ($parametro == "nitratos"){
        $tituloy = "Nitratos";
    }
    if ($parametro == "nitritos"){
        $tituloy = "Nitritos";
    }
    if ($parametro == "sulfatos"){
        $tituloy = "Sulfatos";
    }
    if ($parametro == "fosfatos"){
        $tituloy = "Fosfatos";
    }
    if ($parametro == "manganeso"){
        $tituloy = "Manganeso";
    }
    if ($parametro == "fluoruros"){
        $tituloy = "Fluoruros";
    }
    if ($parametro == "amoniaco"){
        $tituloy = "Nitrógeno amoniacal";
    }
    if ($parametro == "coliformes_fecales"){
        $tituloy = "Coliformes fecales";
    }
}

```



```

        if ($parametro == "coliformes_totales"){
            $tituloy = "Coliformes totales (NMP/100 ml)";
        }
        return $tituloy;
    }
}
function getLimitePermisible($param)
{
    if($param == "ph")
    {
        $lim_sup[0] = 6;

        $lim_sup[1] = 9;
    }
    if($param == "color")
    {
        $lim_sup = 15;
    }
    if($param == "turbiedad")
    {
        $lim_sup = 5;
    }
    if($param == "temperatura")
    {
        $lim_sup = "";
    }
    if($param == "solidos_totales")
    {
        $lim_sup = 500;
    }
    if($param == "conductividad")
    {
        $lim_sup = "";
    }
    if($param == "dureza")
    {
        $lim_sup = 500;
    }
    if($param == "cloro_libre")
    {
        $lim_sup[0] = 0.3;
        $lim_sup[1] = 1.5;
    }
    if($param == "hierro")
    {
        $lim_sup = 0.3;
    }
    if($param == "nitritos")
    {
        $lim_sup = 3;
    }
    if($param == "nitratos")
    {
        $lim_sup = 50;
    }
    if($param == "sulfatos")

```

```
{
    $lim_sup = 250;
}
if($param == "fosfatos")
{
    $lim_sup = 0.3;
}
if($param == "manganeso")
{
    $lim_sup = 0.1;
}
if($param == "fluoruros")
{
    $lim_sup = 1.5;
}
if($param == "amoniaco")
{
    $lim_sup = 1;
}
if($param == "coliformes_fecales")
{
    $lim_sup = 50;
}
if($param == "coliformes_totales")
{
    $lim_sup = 3000;
}
return $lim_sup;}
```

#### 10.4. SOLICITUDES DE ENTREGA DE MANUALES DEL SISTEMA

Riobamba, 17 de noviembre de 2014

Ingeniero

Aurelio Silva

**DIRECTOR REGIONAL DE CESA CENTRO SUR**

Presente;

Por medio de la presente hago entrega del **manual de usuario corregido** del sistema web para la *"Gestión de datos de calidad del agua de consumo doméstico de la provincia de Chimborazo"*, ya que en ocasiones anteriores ya hice entrega de un manual de usuario pero por cuestiones de diseño se decidió darle al sistema un mejor aspecto, este manual conceptualiza aspectos del manejo del sistema ayudando a realizar correctamente las operaciones de ingreso, actualización y en algunos casos de eliminación de datos, este documento está dirigido a los técnicos que se encargan de administrar los datos y al administrador del sistema.

Esperando este material sirva de ayuda para el correcto manejo del sistema que está en operación bajo el dominio de la UNACH con la dirección <http://aguapotable.unach.edu.ec>, anticipo mi agradamiento y gratitud por su paciencia y colaboración.

Atentamente;

  
Juan Carlos Moyota  
CI: 060481111-7



Riobamba, 12 de noviembre de 2014

Ingeniero

Aurelio Silva

**DIRECTOR REGIONAL DE CESA CENTRO SUR**

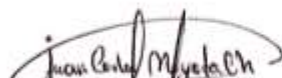
Presente;

Por medio de la presente hago entrega del manual técnico del sistema web para la "Gestión de datos de calidad del agua de consumo doméstico de la provincia de Chimborazo", donde se conceptualizan los aspectos técnicos para la instalación, corrección y mejoras a futuro en el sistema web, dicho manual es enfocado a técnicos y especialistas en desarrollo de sistemas o con conocimientos en desarrollo a aplicativos informáticos.

Además anexo un CD con el manual técnico y de usuario, archivos de código fuente y un respaldo de la base de datos.

Esperando este material sirva de ayuda para el correcto mantenimiento del sistema que está en operación bajo el dominio de la UNACH con la dirección <http://aguapotable.unach.edu.ec>, anticipo mi agradamiento y gratitud por su paciencia y colaboración.

Atentamente;

  
Juan Carlos Moyota Ch  
Juan Carlos Moyota  
CI: 060481111-7



Riobamba, 12 de Noviembre de 2014

Ingeniero

Anibal Armijos

**Técnico del departamento de agua potable y saneamiento de la SENAGUA**

Presente;

Por medio de la presente hago entrega del manual técnico del sistema web CALIAGUA para la "Gestión de datos de calidad del agua de consumo doméstico de la Provincia de Chimborazo", donde se muestra toda la información técnica para el correcto mantenimiento del sistema, correcciones y mejoras a futuro.

Además de un CD con los manuales técnico y de usuario, archivos de código fuente y un respaldo de la base de datos.

Esperando este material sirva de ayuda para el correcto manejo y operación del sistema web.

Atentamente

  
Juan Carlos Moyota PH

CI: 060481111-7

*Proceso Conformado  
12/11/2014  
09 Horas*  


 SECRETARIA DEL AGUA  
DE REGULACION INSTITUCIONAL DE TOLUANA  
CENTRO ZONAL  
RIOBAMBA

## 10.5. FOTOGRAFÍAS DE LAS CAPACITACIONES REALIZADAS



*Figura 85. Capacitacion en laboratorios de la UNACH  
Autor: Juan Carlos Moyota*



*Figura 86. Capacitacion en la SENAGUA subzonal Riobamba  
Autor: Juan Carlos Moyota*



*Figura 87. Segunda capacitacion en la SENAGUA subzonal Riobamba  
Autor: Juan Carlos Moyota*



*Figura 88. Capacitacion en laboratorios de la UNACH - foto 2  
Autor: Juan Carlos Moyota*





*Figura 89. Presentación del sistema en la SENAGUA Latacunga  
Autor: Juan Carlos Moyota*



*Figura 90. Presentación del sistema en la SENAGUA Latacunga - foto 2  
Autor: Juan Carlos Moyota*








*Figura 91. Firma de Acta de entrega del sistema, Quito  
Autor: Juan Carlos Moyota*



*Figura 92. Firma de Acta de entrega del sistema, Quito - foto 2  
Autor: Juan Carlos Moyota*

## 10.6. ACTA DE ENTREGA DEL SISTEMA EN LA SECRETARIA DEL AGUA

 <p>Secretaría del Agua</p>			
--	---	--	---

**ACTA DE ENTREGA Y RECEPCION DE CLAVE DE ACCESO A NIVEL DE ADMINISTRADOR, MANUAL TECNICO Y MANUAL DE USUARIO PARA EL ACCESO Y USO DE LA BASE DE DATOS DEL INVENTARIO PARTICIPATIVO DE RECURSOS HIDRICOS Y CALIDAD DE AGUA POTABLE CHIMBORAZO (CALIAGUA)**


Conste mediante el presente documento que a los 10 días del mes de Febrero del 2015, realiza la entrega La Universidad Nacional de Chimborazo - Centro de Tecnologías Educativas por medio del Ing. Jorge Delgado y CESA/AVSF por medio de la Ing. Marlene Barba Coordinadora del Proyecto Gestión Concertada de la Subcuenca del Chambo y recibe el Mgs. Alexis Sánchez Subsecretario de Agua Potable y Saneamiento - SENAGUA, los siguientes insumos:

CANTIDAD	DESCRIPCION
1	Manual de Usuario del Sistema de Información Hídrica de la Demarcación Hidrográfica del Pastaza.
1	Manual técnico del Sistema de Información Hídrica de la Demarcación Hidrográfica del Pastaza.
1	CD con respaldo magnético de manuales de Usuario y Técnicos
1	Sobre con USERNAME y PASSWORD para el acceso a nivel de administrador del modulo Inventario Participativo de Recursos Hídricos y Calidad de agua potable

La aplicación CALIAGUA – Chimborazo fue desarrollada como trabajo de tesis de la UNACH a cargo del Sr. Juan Carlos Moyota.


La subsecretaría de Agua Potable y Saneamiento - SENAGUA se compromete a manejar las claves con responsabilidad, así como las funciones que esta le permite realizar en el modulo de administración, y asumir la responsabilidad en caso de mal uso de estos accesos y del manual técnico de programación a su cargo.

ENTREGUE CONFORME




Ing. Marlene Barba  
Coordinadora Proyecto Gestión  
Subcuenca Chambo

ENTREGUE CONFORME



Ing. Jorge Delgado  
Director Centro de  
Tecnologías Educativas - UNACH

RECIBI CONFORME



Mgs. Alexis Sánchez  
Subsecretario de Agua Potable  
y Saneamiento - SENAGUA