



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA 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 COMPARATIVO DE LOS FRAMEWORKS LARAVEL Y
CODEIGNITER PARA LA IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE
CONCURSOS DE MÉRITOS Y OPOSICIÓN EN LA UNIVERSIDAD NACIONAL
DE CHIMBORAZO”.**

AUTORES: Jhonatan Santiago Arcos Chalán
Diego Eduardo Chicaiza Inguillay

DIRECTOR: Ing. Javier Haro

Riobamba - Ecuador

2016

PAGINA DE REVISIÓN

Los miembros del tribunal de graduación del proyecto de investigación de título: “Análisis comparativo de los frameworks laravel y codeIgniter para la implementación del sistema de gestión de concursos de méritos y oposición en la Universidad Nacional De Chimborazo” presentado por: Jhonatan Santiago Arcos Chalán y Diego Eduardo Chicaiza Inguillay, dirigido por: Ing., Javier Haro.

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

Para constancia de lo expuesto firman:

Ing. Javier Haro
Director de Tesis

FIRMA

Ing. Danny Velasco
Presidente de Tribunal

FIRMA

Ing. Alejandra Pozo
Miembro del Tribunal

FIRMA

AUTORIA DE LA INVESTIGACIÓN

"La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Jhonatan Santiago Arcos Chalán, Diego Eduardo Chicaiza Inguillay, a la Ing. Javier Haro Tutor; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo"



Jhonatan Santiago Arcos Chalán

CI. 060410767-2



Diego Eduardo Chicaiza Inguillay

CI. 060447087-2



Ing. Javier Haro

DIRECTOR DE TESIS

AGRADECIMIENTO

Agradezco en primer lugar a Dios, por haberme dado la vida, y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A mis padres por darme su apoyo incondicional, amor y fortaleza en todo momento para seguir adelante en el cumplimiento de una meta más en mi vida.

Al Ing. Javier Haro por su apoyo, dirección y guía para la realización de este proyecto de tesis.

A los docentes de la Facultad de Ingeniería en Sistemas y Computación por sus horas de dedicación para impartirnos sus conocimientos y experiencias profesionales.

A mis compañeros de estudio y amigos, quienes sin su ayuda nunca hubiera podido hacer esta tesis. A todos ellos se los agradezco desde el fondo de mi alma.

Diego Chicaiza Inguillay

AGRADECIMIENTO

Agradezco a mis padres por darme su apoyo incondicional, amor y fortaleza en todo momento para seguir adelante en el cumplimiento de una meta más en mi vida.

Al Ing. Javier Haro por su apoyo, dirección y guía para la realización de este proyecto de tesis.

A los docentes de la Facultad de Ingeniería en Sistemas y Computación por sus horas de dedicación para impartirnos sus conocimientos y experiencias profesionales.

Jhonatan Arcos Chalán

ÍNDICE GENERAL

CONTENIDO	
ÍNDICE GENERAL.....	VI
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XII
RESUMEN	XIII
INTRODUCCIÓN	15
CAPITULO I	17
MARCO REFERENCIAL	17
1.1 TÍTULO DEL PROYECTO.....	17
1.2 PROBLEMATIZACIÓN.....	17
1.2.1 IDENTIFICACION Y DESCRIPCION DEL PROBLEMA	17
1.2.2 PROGNOSIS	18
1.2.3 DELIMITACIÓN.....	18
1.2.4 FORMULACIÓN DEL PROBLEMA.....	19
1.2.5 HIPÓTESIS.....	19
1.2.6 IDENTIFICACIÓN DE VARIABLES	20
1.2.6.1 VARIABLE INDEPENDIENTE.....	20
1.2.6.2 VARIABLE DEPENDIENTE	20
1.3 OBJETIVOS.....	20
1.3.1 GENERAL.....	20
1.3.2 ESPECÍFICOS	20
1.4 JUSTIFICACIÓN.....	20
CAPITULO II	22
FUNDAMENTACIÓN TEÓRICA	22
2.1 FRAMEWORK.....	22
2.1.1 CONCEPTO DE FRAMEWORK	22
2.1.2 CARACTERÍSTICAS DE UN FRAMEWORK	22
2.2 LENGUAJES DE PROGRAMACIÓN PARA DESARROLLO WEB.	22
2.3 PHP	23
2.3.1 DEFINICIÓN DE PHP.....	23
2.3.2 EVOLUCIÓN DE PHP	24
2.3.3 VENTAJAS DE PHP	24
2.3.4 CARACTERÍSTICAS DE LEGUAJE DE PROGRAMACIÓN PHP.....	25

2.3.5 SINTAXIS	26
2.3.6 VARIABLES	26
2.3.7 COMENTARIOS	26
2.3.8 TIPOS DE DATOS.....	27
2.3.9 CONSTANTES	27
2.4 HTML	28
2.4.1 DEFINICIÓN DE HTML.....	28
2.4.2 CARACTERÍSTICAS DE HTML.....	28
2.4.3 EVOLUCIÓN DEL HTML.....	29
2.4.4 VENTAJAS DE HTML.....	29
2.5 MODELO, VISTA, CONTROLADOR.....	30
2.5.1 CONCEPTO DE MODELO, VISTA, CONTROLADOR	30
2.5.2 VISTAS.....	30
2.5.3 CONTROLADOR	31
2.6 JAVASCRIPT	31
2.7 CSS.....	31
2.7.1 DEFINICIÓN DE CSS.....	31
2.7.2 EVOLUCIÓN DE CSS	31
2.8 BOOTSTRAP	32
2.8.1 CONCEPTO DE BOOTSTRAP.....	32
2.8.2 EVOLUCIÓN DE BOOTSTRAP	32
2.8.3 VENTAJAS Y DESVENTAJAS DE BOOTSTRAP	33
2.9 POSTGRESQL	34
2.9.1 INTRODUCCIÓN.....	34
2.9.2 CARACTERÍSTICAS.....	36
2.10 APLICACIÓN WEB	38
2.10.1 DEFINICIÓN DE APLICACIÓN WEB.....	38
2.10.2 VENTAJAS	39
2.11 MÉTRICAS DE SOFTWARE.....	39
2.11.1 DEFINICIÓN.....	39
2.11.2 IMPORTANCIA DE MEDIR SOFTWARE.....	40
2.11.3 CARACTERÍSTICAS.....	40
2.11.4 MÉTRICAS DEL PROCESO Y DEL PROYECTO.	41
2.11.4.1 MÉTRICAS DEL PROCESO.....	41

2.11.4.2 MÉTRICAS DEL PRODUCTO.....	41
2.11.5 CLASIFICACIÓN DE LAS MEDIDAS DE SOFTWARE.	42
2.11.6 PRODUCTIVIDAD	42
2.11.6.1 DEFINICIÓN.....	42
2.11.6.2 FACTORES QUE AFECTAN A LA PRODUCTIVIDAD.	42
2.11.7 MÉTRICAS DE PRODUCTIVIDAD	43
2.11.7.1 MÉTRICAS ORIENTADAS AL TAMAÑO.	43
2.11.7.2 LÍNEAS DE CÓDIGO (LDC)	44
CAPITULO III	45
ANÁLISIS COMPARATIVO ENTRE LOS FRAMEWORK LARAVEL Y CODEIGNITER.....	45
3.1 FRAMEWORKS DE OPEN SOURCE	45
3.1.1 ANÁLISIS COMPARATIVO ENTRE LOS FRAMEWORK PHP: LARAVEL Y CODEIGNITER	46
3.2 LARAVEL	47
3.2.1 DEFINICIÓN.....	47
3.1.2 HISTORIA DEL FRAMEWORK LARAVEL	47
3.1.3 CARACTERÍSTICAS DEL FRAMEWORK LARAVEL	48
3.1.4 FILOSOFÍA.	49
3.1.5 ARQUITECTURA	49
3.1.6 CAPAS MVC EN LARAVEL.....	49
3.1.6.1 CAPA DEL MODELO	49
3.1.6.2 CAPA DE VISTA	50
3.1.6.3 CAPA DEL CONTROLADOR.....	50
3.1.6.4 CICLO DE PETICIÓN MVC EN LARAVEL	50
3.1.7 ELOQUENT ORM.....	50
3.1.8 REQUERIMIENTOS	51
3.1.8.1 LENGUAJE DE PROGRAMACIÓN.	51
3.1.8.2 EXTENSIÓN DE MCRYPT PHP.....	51
3.1.8.3 COMPOSER.....	51
3.1.8.4 SERVIDOR WEB.....	51
3.1.8.5 SERVIDOR DE BASES DE DATOS.....	51
3.2 CODEIGNITER	52
3.2.1 DEFINICIÓN FRAMEWORK CODEIGNITER.	52
3.2.2 HISTORIA DEL FRAMEWORK CODELGNITER.....	52
3.2.3 CARACTERÍSTICAS GENERALES DE CODEIGNITER	53

3.2.4 ARQUITECTURA DEL FRAMEWORK CODEIGNITER.....	54
3.3.5 MODELO - VISTA - CONTROLADOR EN CODEIGNITER.....	54
3.3 DETERMINACIÓN DE PARÁMETROS DE COMPARACIÓN	56
3.3.1 NÚMERO DE LÍNEAS DE CÓDIGO	56
3.3.2 TIEMPO DE RESPUESTA DEL SISTEMA.....	56
3.3.3 MANEJO DE SEGURIDAD	57
3.4 PONDERACIÓN DE LOS INDICADORES DE LOS PARÁMETROS DE COMPARACIÓN	58
3.5 INSTRUMENTOS DE MEDICIÓN	59
3.6 DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS	61
3.7 SISTEMA UTILIZANDO EL FRAMEWORK LARAVEL.....	61
3.8 SISTEMA UTILIZANDO EL FRAMEWORK CODEIGNITER.....	62
3.9 DESARROLLO DE LAS PRUEBAS CON LOS PARÁMETROS DE COMPARACIÓN.....	63
3.10 PARÁMETRO 1: NÚMERO DE LÍNEAS DE CÓDIGO.....	63
3.10.1 RESULTADOS	64
3.10.2 CÁLCULO DE PESOS	64
3.10.3 GRÁFICO ESTADÍSTICO	65
3.10.4 GRÁFICO ESTADÍSTICO DE LOS RESULTADOS FINALES DEL PARÁMETRO 1.....	65
3.11 PARÁMETRO 2: TIEMPO DE RESPUESTA DEL SISTEMA.....	66
3.11.1 TIEMPO DE RESPUESTA DEL SISTEMA DE TRANSACION DE INSERCIÓN DE DATOS DE LOS POSTULANTES.	66
3.11.2 TIEMPO DE RESPUESTA DEL SISTEMA EN TRANSACCIONES DE MODIFICACIÓN DE DATOS DE LOS POSTULANTE.....	67
3.11.3 TIEMPO DE RESPUESTA DEL SISTEMA EN TRANSACCIONES DE CONSULTA DE DATOS DE LOS POSTULANTES.	67
3.11.4 TIEMPO DE RESPUESTA DEL SISTEMA EN TRANSACCIONES DE ELIMINACIÓN DE DATOS DE LOS POSTULANTES.....	68
3.11.5 GRÁFICO ESTADÍSTICO	69
3.11.6 GRÁFICO ESTADÍSTICO	70
3.11.7 DESCRIPCIÓN DE RESULTADOS	70
3.11 PARÁMETRO 3: MANEJO DE SEGURIDAD.....	71
3.11.1 MANEJO DE SEGURIDAD.....	71
3.11.2 CALIFICACIÓN	71
3.11.3 CÁLCULO DE PESOS	71
3.11.4 GRÁFICO ESTADÍSTICO	72
3.11.5 GRÁFICO ESTADÍSTICO	73

3.12 ANÁLISIS DE RESULTADOS.....	73
3.12.1 GRÁFICO ESTADÍSTICO	74
3.12.2 GRÁFICO ESTADÍSTICO	74
CAPITULO IV	76
IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE PROCESOS DE CONCURSOS DE MÉRITOS Y OPOSICIONES EN LA UNIVERSIDAD NACIONAL DE CHIMBORAZO CON EL FRAMEWORKS SELECCIONADO.	76
4.1 METODOLOGÍA XP.....	76
4.1.1 PROPÓSITO.....	77
4.1.2 ÁMBITO DEL SISTEMA	77
4.1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS	78
4.2 DESARROLLO DEL SISTEMA	78
4.1.1 HERRAMIENTAS DE DESARROLLO	78
4.3 GESTIÓN DEL PROYECTO	80
4.3.1 PLANIFICACIÓN DEL PROYECTO	80
4.3.2 INTEGRANTES Y ROLES	80
4.3.3 DESCRIPCIÓN DEL SISTEMA.....	80
4.3.4 ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA	81
4.3.5 REQUERIMIENTOS FUNCIONALES.....	81
4.2 FASE II. DISEÑO DE SOFTWARE	83
4.2.1 DISEÑO DE LA BASE DE DATOS.....	83
4.2.2 DICCIONARIO DE DATOS	84
4.2.3 DISEÑO DE ARQUITECTURA	85
4.2.4 DIAGRAMA DE CASOS DE USOS	87
4.3 PROGRAMACIÓN.....	91
4.4 IMPLEMENTACIÓN	91
CAPITULO V	92
METODOLOGÍA.....	92
5.1 TIPO DE ESTUDIO.....	92
5.1.1 SEGÚN EL OBJETO DE ESTUDIO:.....	92
5.1.2 SEGÚN LA FUENTE DE INVESTIGACIÓN:.....	92
5.1.3 SEGÚN LAS VARIABLES:.....	92
5.2 HIPÓTESIS.....	92
5.2.1 FORMULACIÓN DE LA HIPÓTESIS.....	92
5.2.2 TIPO DE HIPÓTESIS	92

5.2.3 DETERMINACIÓN DE LAS VARIABLES	93
5.2.4 OPERACIONALIZACIÓN CONCEPTUAL.....	93
5.2.5 COMPROBACIÓN DE LA HIPÓTESIS	95
CAPITULO VI	99
6.1 CONCLUSIONES	99
6.2 RECOMENDACIONES	100
7 BIBLIOGRAFÍA	101
8 APENDICES O ANEXOS.....	105
8.1 GLOSARIO	105
8.2 ANEXO	107

ÍNDICE DE FIGURAS

Figura 1: Tabla comparativa entre los frameworks PHP: Laravel y CodeIgniter	19
Figura 2 Etiquetas de PHP	26
Figura 3: Diseño de MVC	30
Figura 4: Componentes en un sistema PostgreSQL.....	35
Figura 5: logo del framework Laravel.....	47
Figura 6: logo del framework CodeIgniter	52
Figura 7: Interfaz Principal del prototipo con el Framework Laravel.....	62
Figura 8: Interfaz Principal del sistema con el Framework CodeIgniter	63
Figura 9: resultado de las líneas de código de los dos Frameworks PHP: Laravel y CodeIgniter	65
Figura 10: Resultado Final del parámetro 1.	65
Figura 11: Resultado por indicador del Parámetro 2: Tiempo de Respuesta del Sistema.....	69
Figura 12: Resultado Final del Parámetro 2: Tiempo de Respuesta del Sistema.....	70
Figura 13: Resultado por indicador del Parámetro 3: Manejo de Seguridad.....	72
Figura 14: Resultado Final del Parámetro 3: Manejo de Seguridad.....	73
Figura 15: Resultados porcentuales de cada parámetro de comparación.	74
Figura 16: Resultado Final del Análisis	74
Figura 17: Fases de la Metodología XP.....	77
Figura 18: Diseño de la Base de Datos.	84
Figura 19: Proceso de planificación y envió de cronogramas	87
Figura 20: Gestión de categorías.....	87
Figura 21: Gestión de concurso.....	88
Figura 22: Gestión de matriculas	88
Figura 23: Gestión de usuarios.....	89
Figura 24: Procedimiento para el reporte de resultados	89

Figura 25: Proceso de calificación de requisitos	90
Figura 26: Procesos de calificación de mérito y oposición.....	90
Figura 27: Distribución t de Student	97
Figura 28: Grafica de T de Student.....	98

ÍNDICE DE TABLAS

Tabla 1: Tipos de datos primitivos en PHP	27
Tabla 2: Tipos de datos especiales en PHP.....	27
Tabla 3: Tipos de datos compuestos en PHP	27
Tabla 4: Límites de PostgreSQL	38
Tabla 5: Descripción de los indicadores del Parámetro: Número de líneas de código.....	56
Tabla 6: Descripción de los indicadores del Parámetro: Tiempo de respuesta del sistema.....	57
Tabla 7: Descripción de los indicadores del Parámetro: Manejo de seguridad.....	58
Tabla 8: Ponderación de los Indicadores de los Parámetros de comparación	59
Tabla 9: Herramientas de Hardware Como Software para realizar las pruebas.....	61
Tabla 10: Resultados del Parámetro 1: Número de líneas de código	64
Tabla 11: Peso porcentual de los indicadores del Parámetro 1: Número de líneas de código... 64	64
Tabla 12: Resultados del indicador Tiempo de respuesta del sistema en transacciones de inserción	66
Tabla 13: Resultados del indicador Tiempo de respuesta del sistema en transacciones de modificación de datos de los postulantes.....	67
Tabla 14: Resultados del indicador Tiempo de respuesta del sistema en transacciones de consulta	68
Tabla 15: Resultados del indicador Tiempo de respuesta del sistema en transacciones de eliminación de datos de los postulantes.....	68
Tabla 16: Peso porcentual de los indicadores del Parámetro 6: Tiempo de Respuesta del Sistema	69
Tabla 17: Escala de Evaluación Parámetro 3: Manejo de seguridad.....	71
Tabla 18: Calificación del parámetro 5: Manejo de Seguridad	71
Tabla 19: Peso porcentual de los indicadores del Parámetro 3: Manejo de Seguridad	72
Tabla 20: Resultados Finales de los Parámetros de Comparación.....	73
Tabla 21: Definiciones	78
Tabla 22: Abreviaturas	78
Tabla 23: Herramientas de Desarrollo para SIGCOM.....	79
Tabla 24: Integrantes y Roles	80
Tabla 25: Descripción tabla Usuarios Base de datos SIGCOM.	85
Tabla 26: Operacionalización Conceptual.....	94
Tabla 27: Cálculos.....	96

RESUMEN

Se realizó el análisis comparativo de los Frameworks PHP: Laravel y CodeIgniter para mejorar la productividad en el desarrollo del Sistema de gestión de concursos y mérito y oposición en la Universidad Nacional de Chimborazo. Se registró y analizó los resultados de las pruebas realizadas a los dos sistemas desarrollados en base a los parámetros e indicadores establecidos, para llegar a determinar el framework con mejor productividad. Las herramientas utilizadas en hardware: una computadora, y en software: los frameworks PHP: Laravel y CodeIgniter, HTML 5, JavaScript, Bootstrap 3.3.6, Microsoft Visio 2010, JMeter y Quick Line Counter QLC. De acuerdo a los resultados obtenidos se infiere que el framework Laravel mejora la productividad en el desarrollo del Sistema de gestión de concursos y mérito y oposición para la Universidad Nacional de Chimborazo, cumpliendo con el 100% de los parámetros establecidos en el análisis el mismo que es mayor, en comparación al 68.86% de cumplimiento del Framework CodeIgniter. Se desarrolló con el Framework Laravel el Sistema de gestión de concursos y mérito y oposición en la Universidad Nacional de Chimborazo, ya que brinda mejor productividad en el desarrollo de aplicaciones web. Se recomienda a la institución hacer uso del Sistema debido a que esta aplicación web automatiza todos los procesos de manera eficiente.

Palabras Claves: <PRODUCTIVIDAD> <FRAMEWORKS PHP> <LARAVEL FRAMEWORK>
<CODEIGNITER FRAMEWORK> <SISTEMAS>.



UNIVERSIDAD NACIONAL DE CHIMBORAZO

CENTRO DE IDIOMAS INSTITUCIONAL

Master. Janneth Caisaguano

1 de agosto del 2016

ABSTRACT

The analysis was made of the comparative frameworks PHP: Laravel and CodeIgniter to improve productivity in the development of the system of management of competitions and merit and opposition in the National University of Chimborazo. It is recorded and analyzed the results of the tests performed on the two systems developed on the basis of the parameters and indicators established, in order to determine the framework with better productivity. The tools used in hardware: a computer, and in software - the frameworks PHP: Laravel and CodeIgniter, HTML 5, JavaScript, Bootstrap 3.3.6, Microsoft Visio 2010, JMeter and Quick Line Counter QLC. According to the obtained results it is inferred that the framework Laravel improves productivity in the development of the system of management of competitions and merit and opposition to the National University of Chimborazo, complying with the 100% of the parameters established in the analysis the same that is greater, compared to 68.86% of compliance with the Framework CodeIgniter. Was developed with the Framework Laravel the management system of competitions and merit and opposition in the National University of Chimborazo, because it provides better productivity in the development of web applications. It is recommended that the institution to make use of the system due to this web application automate all processes in an efficient manner.

Keywords: < PRODUCTIVITY > <FRAMEWORKS PHP> < LARAVEL FRAMEWORK> < CODEIGNITER FRAMEWORK > < SYSTEM >.



INTRODUCCIÓN

En el desarrollo de sistemas o software se aprecia mediante una necesaria mejora en la puesta en práctica de metodologías de desarrollo, así también como la flexibilización de estas para potenciar la productividad de las mismas sin renunciar a la calidad en el desarrollo del software.

Por esta razón se hace necesario disponer de herramientas efectivas para aumentar la productividad. La mejora de la productividad en el desarrollo de software está ligada a la utilización de buenas prácticas de Ingeniería de Software.

Mediante los Frameworks de desarrollo PHP se puede obtener mayor productividad en el desarrollo de una aplicación web, ya que estas son herramientas que brindan una estructura definida las mismas que nos permiten crear aplicaciones con mayor facilidad y rapidez, con orientación a dar solución en menor tiempo posible a problemas. (Mehdi Achour & Jakub Vrana, 2016)

En la presente de la investigación de Vilcaguano Zumba María Ibeth y Tierra Llamuca Juan Manuel, con experiencia en Análisis Comparativo de frameworks PHP (Vilcaguano Zumba, 2015) y en base las normas métricas de calidad 1045 (Vargas, 2016). Además, en el criterio de del investigador de Software Libre de la UOC Mario Gómez Molina.a través de indicadores de comparación aplicados a cada uno de los dos sistemas realizados.

El presente proyecto de investigación, contiene los siguientes capítulos presentados a continuación:

En el **Capítulo I**, contiene todo lo relacionado al Marco Referencial como: los antecedentes, la justificación del proyecto de tesis, los objetivos planteados y la hipótesis, la misma que se demostrará en las pruebas finales realizadas.

En el **Capítulo II**, comprende todo lo referente al Marco Teórico, las definiciones conceptuales aplicaciones Web, Framework PHP Laravel, y del Framework PHP CodeIgniter, dicha información será de muy importancia para la ejecución de este proyecto de investigación.

En el **Capítulo III**, directamente se enfoca principalmente en el análisis comparativo entre los Frameworks PHP Laravel y CodeIgniter, para lo cual en forma detallada se define las herramientas utilizadas, creación de los dos sistemas, e indicadores de productividad establecidos, para determinar el framework de mayor productividad en el desarrollo de aplicaciones web para la Universidad Nacional de Chimborazo.

En el **Capítulo IV**, se detalla el desarrollo del Sistema de gestión de concursos y mérito y oposición en la Universidad Nacional de Chimborazo, aplicando la metodología de desarrollo XP (Programación Extrema), con el framework que proporciona mayor productividad, que se seleccionó en el Capítulo III, en base a los resultados obtenidos de las pruebas.

En el **Capítulo V**, se demuestra la hipótesis planteada en el capítulo I utilizando el método T de Student.

El presente proyecto de investigación, servirá de soporte para la toma de decisiones, al elegir entre los dos frameworks PHP: Laravel y CodeIgniter, y así optar por el que brinda mayor productividad al desarrollo del sistema.

CAPITULO I

MARCO REFERENCIAL

1.1 TÍTULO DEL PROYECTO

ANÁLISIS COMPARATIVO DE LOS FRAMEWORKS LARAVEL Y CODEIGNITER PARA LA IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE CONCURSOS DE MÉRITOS Y OPOSICIÓN EN LA “UNIVERSIDAD NACIONAL DE CHIMBORAZO”.

1.2 PROBLEMATIZACIÓN

1.2.1 IDENTIFICACION Y DESCRIPCION DEL PROBLEMA

Hoy en día, con el avance de las tecnologías de información, los desarrolladores de aplicaciones informáticas ponen interés en dos aspectos fundamentales: El primer aspecto a considerar es la construcción de aplicaciones en un tiempo determinado. El segundo aspecto a valorar es el desarrollo de aplicaciones que contengan estándares o patrones de diseño que permitan mayor reutilización de código y fácil mantenimiento de los sistemas.

Los ingenieros en sistemas o programadores deben ser eficientes al instante de dar solución a un problema, para esto buscan nuevas tecnologías que simplifiquen la tarea de diseño y desarrollo de sistemas informáticos.

Diseñar la arquitectura de un software es la fase primordial para la elección de los mejores instrumentos de desarrollo que permitan crear sistemas estandarizados, asegurando el rendimiento sin importar futuras alteraciones que se produzcan.

Los DreamWorks son desarrollados con el objetivo de brindarles a los programadores y diseñadores una mejor organización y estructura a sus proyectos. Se utiliza la Programación Orientada a Objetos (POO), permitiendo la reutilización de nuestro código, aparte de que imponen patrones de diseño como MVC (Modelo-Vista-Controlador).

A la hora de seleccionar una herramienta o framework de PHP para el desarrollo de aplicaciones Web se deben considerar dos puntos sustanciales; el primero es el alcance del proyecto y segundo es la experiencia que tiene el programador o equipo de trabajo.

Uno de los procesos administrativos de la Carrera de Sistemas y Computación son los concursos de méritos y oposición para llenar las vacantes de docentes del sector público, se realizará de conformidad con lo prescrito en los Arts. 93 al 107 de la Ley Orgánica de Educación Intercultural, así como los Arts. 262 al 300 del Reglamento General de la indicada Ley y la normativa que regulan los actuales concursos: Acuerdo Ministerial 408-12 expedido el 12 de septiembre del 2012.

Es de suma importancia para la carrera realizar la calificación de postulantes a docentes de forma automática, rápida y eficiente mediante la implementación de un sistema informático que gestione dicho proceso, y que utilice el framework de PHP con el patrón de diseño MVC (Modelo-vista-Controlador) que más se adapte a los requerimientos funcionales.

1.2.2 PROGNOSIS

El estudio comparativo de frameworks de PHP permitirá seleccionar la mejor herramienta para el aumento de la productividad de cualquier aplicativo web, en este caso para la implementación del Sistema de gestión de procesos de merecimientos y oposición, agilizando la entrega de los resultados de manera rápida y segura, ya que cada candidato recibirá en su correo electrónico información acerca de las fases del concurso en las que participe y los puntajes parciales y final obtenido.

1.2.3 DELIMITACIÓN

La investigación se realizará en la Universidad Nacional de Chimborazo, en la Facultad de Ingeniería, en la Carrera de Sistemas y Computación ubicado en el Km 1 ½ vía a Guano Campus Norte.

El estudio comparativo será de los framework de PHP; Laravel y CodeIgniter, escogidos basándose en la medición del interés de búsqueda general de Google Trends por parte de los usuarios que día a día realizan consultas de diferentes frameworks en este buscador.

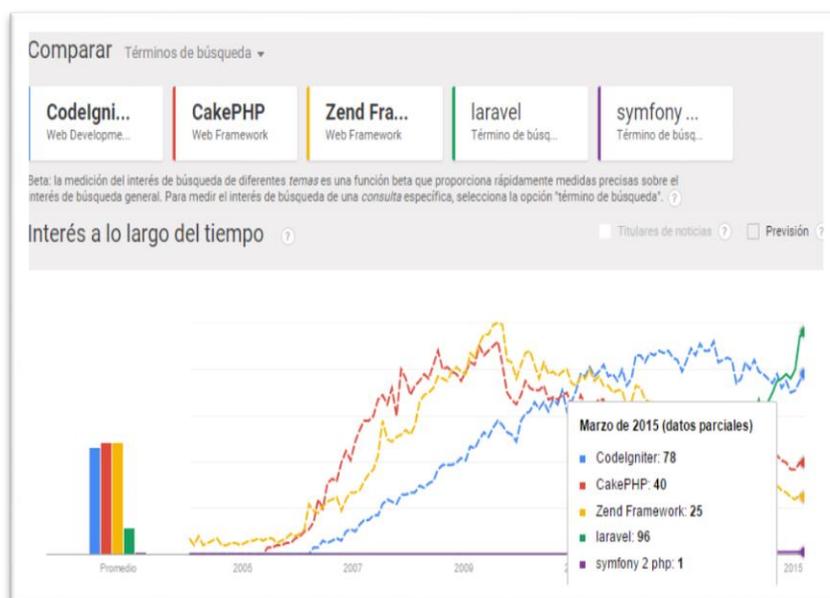


Figura 1: Tabla comparativa entre los frameworks PHP: Laravel y CodeIgniter

Elaborado por: Diego Chicaiza & Jhonatan Arcos

La implementación del sistema de gestión de procesos de Méritos (Contratos que se realizan Ocasionalmente). Méritos y oposiciones (Para nombramientos indefinidos) se la realizará para todas las carreras de pregrado de las cuatro facultades de la UNACH. El tiempo que tomará el desarrollo de la investigación es aproximadamente 14 meses, de tal manera que el sistema será implementado en los servidores de Data center en el edificio CTE (Centro de Tecnologías Educativas).

1.2.4 FORMULACIÓN DEL PROBLEMA

¿El Análisis comparativo de los frameworks Laravel y CodeIgniter, permitirá seleccionar la mejor herramienta en la implementación del Sistema de Gestión de concursos de méritos y oposición en la “Universidad Nacional de Chimborazo”?

1.2.5 HIPÓTESIS

El Framework Laravel brinda las mejores prestaciones de productividad en el desarrollo del sistema de Gestión de concursos de méritos y oposición en la “Universidad Nacional de Chimborazo.

1.2.6 IDENTIFICACIÓN DE VARIABLES

1.2.6.1 VARIABLE INDEPENDIENTE

Análisis comparativo de los frameworks Laravel y CodeIgniter

1.2.6.2 VARIABLE DEPENDIENTE

Sistemas de gestión de concursos de mérito y oposición en la “Universidad Nacional de Chimborazo”.

1.3 OBJETIVOS

1.3.1 GENERAL

Analizar y comparar los frameworks Laravel y CodeIgniter para la implementación de un Sistema de Gestión de procesos de concursos de méritos y oposiciones en la Universidad Nacional de Chimborazo.

1.3.2 ESPECÍFICOS

- Realizar un análisis comparativo los framework PHP: Laravel y CodeIgniter.
- Implementar escenarios de pruebas para medir la productividad y seleccionar el framework adecuado.
- Implementar Sistema de Gestión de concursos de méritos y oposición basado en la plataforma PHP utilizando el framework seleccionado en la Universidad Nacional de Chimborazo.

1.4 JUSTIFICACIÓN

Las Instituciones de educación a nivel superior dentro del país requieren automatizar cada uno de sus procesos administrativos, adaptándose convenientemente a políticas gubernamentales. En el departamento de procesos con más transparencia y a la vez eficiente, siendo oportuno al momento de realizar los concursos de méritos y oposiciones, para la contratación del personal no titular (Honorarios, Invitados y Ocasional).

Los concursos de méritos se realiza cada inicio del periodo semestral expuestos según la ley de la institución estipula en artículo 14 del literal e hasta el literal g deberán acatar a cada uno de estos literales según el concurso de mérito que se va a realizarse. Además, dentro de la institución se realizan concursos de méritos y oposición para la titularidad (Principales, Auxiliares y Agregados) según la estipulación del artículo 15 del literal a

hasta el literal **d**, donde menciona que para el ingreso del concurso cada concursante deberá cumplir con los requisitos expuestos en los literales antes mencionados según en la titularidad que se vaya a concursar.

PHP es uno de los lenguajes con mayor popularidad entre los lenguajes de programación, debido a que está enfocado exclusivamente a la programación web, y está claro que hoy en día la programación web está en auge. Otra de las razones es la cantidad de CMS (Sistema Gestor de Contenidos) y frameworks que utilizan este lenguaje, lo cual facilita el desarrollo de páginas web, portales, blogs, redes sociales, etcétera, apoyada además a en su enorme comunidad que hay detrás.

PHP está muy presente en internet, entre las grandes webs que utilizan PHP tenemos a: Yahoo INC, Wikipedia.org, Friendster.com, Facebook.com, Digg.com, Sourceforge.org, Flickr.com

La aplicación será alojada en el Data Center del Centro de Tecnologías Educativas de la UNACH. Este Data Center es de nivel 3 que ofrece una disponibilidad del 99,98%.

Con esta configuración, existe la posibilidad de programar periodos de mantenimiento sin que afecten a la continuidad del servicio en los servidores. Además, tiene niveles importantes de tolerancia a fallos al contar con todos los equipamientos básicos redundados incluido el suministro eléctrico, permitiéndose una configuración Activo / Pasivo.

CAPITULO II

FUNDAMENTACIÓN TEÓRICA

2.1 FRAMEWORK

2.1.1 CONCEPTO DE FRAMEWORK

Un framework es considerado como un esqueleto o patrón que no tiene un lenguaje concreto, aunque sea así en muchas ocasiones, en otras palabras, un framework se considera una aplicación genérica incompleta y configurable que podemos añadir lo que nosotros necesitemos para realizar una aplicación. (Gutiérrez., 2012)

2.1.2 CARACTERÍSTICAS DE UN FRAMEWORK

Entre las principales características de un framework son:

- **Abstracción de URLs y sesiones.** Los framework son los encargados de manipulación.
- **Acceso a Datos.** Los framework cuentan con todas las herramientas e interfaces necesarias.
- Los framework son los encargados de crear sus propios controladores estos suelen ser fáciles, adaptables a los proyectos.
- **Autenticación y control.** Mantiene la seguridad tanto para la autenticación a través de un login y un password y para el acceso a determinados usuarios.
- **Separación entre diseño y contenido.** Tienen herramientas e interfaces necesarias para su diseño y contenido. (Gutiérrez., 2012)

2.2 LENGUAJES DE PROGRAMACIÓN PARA DESARROLLO WEB.

Al mismo tiempo de que surgió la necesidad de mejorar el contenido de las aplicaciones web de estáticas a dinámicas, los lenguajes de programación han evolucionado a la par con la interacción de los datos con el usuario a través de la aplicación web. A continuación, se describen los más destacados:

- **PHP:** Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. (Mehdi Achour & Jakub Vrana, 2016). Este lenguaje esta publicado bajo la licencia MIT.
Además de tener las características de ser multiplataforma, libre por lo cual es de fácil acceso para todos, cuenta con una curva de aprendizaje muy fácil. Siendo este un lenguaje de programación centrado en el lado del servidor por medio de scripts, también realiza algo innovador al lado del cliente, ejecutando el código en el servidor, generando HTML y enviándolo al cliente. (Mehdi Achour & Jakub Vrana, 2016)
- **Java Server Pages(JSP):** Es una tecnología para generar páginas web en el servidor, basado en scripts de igual manera que PHP y ASP, el cual puede incrustar código HTML o tenerlo contenido en su interior, introduciendo a lo que se conoce como eventos ejecutados por los usuarios. Publicado bajo la Licencia Pública General de GNU, siendo así libre para desarrollar. Entre sus características es que es multiplataforma, permite a los programadores generar dinámicamente HTML, XML o algún otro tipo de página web, permite la utilización de servlets.

Entre estos 2 lenguajes de programación son las más usadas para el desarrollo web, y además se puede concluir que a nivel de licencias JSP y PHP son libres.

2.3 PHP

PHP, es un lenguaje de programación de código abierto se utiliza para el desarrollo web, y puede ser embebido en páginas HTML y ejecutados en un servidor. Su sintaxis es similar a C, Java y Perl y es fácil de aprender. (Mehdi Achour & Jakub Vrana, 2016)

2.3.1 DEFINICIÓN DE PHP

A PHP se define como a un lenguaje de programación interpretado. Este lenguaje es al que le debemos la visualización de contenido dinámico en las páginas web. Todo el código PHP es invisible para el usuario, porque todas las interacciones que se desarrollan en este lenguaje son por completo transformadas para que se puedan ver imágenes, variedad de multimedia y los formatos con los que somos capaces de interactuar añadiendo o descargando información de ellos. (García, 2014)

2.3.2 EVOLUCIÓN DE PHP

PHP ha recorrido un largo camino desde su nacimiento a mediados de los 90. Desde sus humildes comienzos hasta ser uno de los más importantes lenguajes de programación web. (Mehdi Achour & Jakub Vrana, 2016)

En junio de 1995, Rasmus publicó el código fuente de PHP, lo que permitió a los desarrolladores usarlo de manera apropiada.

PHP 3.0 fue la primera versión una de las mejores características de PHP 3.0 era proveer una interfaz madura para múltiples bases de datos, la sencillez de ampliar el lenguaje atrajo a docenas de desarrolladores que presentaron variedad de módulos. Podría decirse que esta fue la clave para el éxito de PHP 3.0 (Mehdi Achour & Jakub Vrana, 2016)

PHP 4.0 en esta versión estuvo asociado con gran rango de nuevas características adicionales, entre ellas como el soporte para la mayoría de los servidores Web, sesiones HTTP, buffers de salida, formas más seguras de controlar las entradas de usuario y muchas nuevas construcciones de lenguaje. (Mehdi Achour & Jakub Vrana, 2016)

PHP 5 fue lanzado en Julio del 2004 después de un largo desarrollo está básicamente impulsado por su núcleo, Zend Engine 2.0 que contiene un nuevo modelo de objetos y docenas de nuevas opciones, entre ellas cuentan con documentación, y una infraestructura en red subyacente de más de cien servidores web individuales en seis de los siete continentes del mundo. Aunque es solo una estimación basada en estadísticas de años anteriores, es seguro suponer que PHP ahora está instalado en diez o quizá cien millones de dominios en todo el mundo. (Mehdi Achour & Jakub Vrana, 2016)

La versión 6 de PHP ha estado en preparación durante largo tiempo en formato de borrador, sin llegar a publicarse. Por ello se propuso que la nueva versión de PHP se llamara PHP 7, dejando PHP 6 como una versión no publicada.

2.3.3 VENTAJAS DE PHP

- **Multiplataforma.** A diferencia de otros lenguajes (especialmente de ASP), se trata de un lenguaje que se puede lanzar en casi todas las plataformas de trabajo (Windows, Linux, Mac) (McDade, 2016)

- **Abierto y gratuito.** Pertenece al software licenciado como GNU, la licencia del sistema Linux; lo que permite su distribución gratuita y que la comunidad mejore el código. (Otwell, 2016)
- **Gran comunidad de usuarios.** La popularidad de PHP, permite tener una comunidad amplia y muy dinámica a la que se puede acudir en caso de necesidad.
- **Apache, MySQL.** Apache es el servidor web y de aplicaciones más utilizado en la actualidad. MySQL es el servidor de bases de datos relacionales más popular en Internet para crear aplicaciones web. (Hazaël-Massieux, 2016)
- **Extensiones.** Dispone de un enorme número de extensiones que permiten ampliar las capacidades del lenguaje, facilitando la creación de aplicaciones web complejas. (McDade, 2016)

2.3.4 CARACTERÍSTICAS DE LENGUAJE DE PROGRAMACIÓN PHP

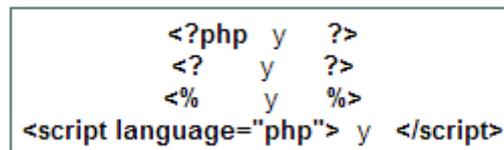
A continuación, se listan las características más importantes:

- Multiplataforma.
- Completamente orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una Base de Datos.
- Seguro y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos o extensiones.
- Buena Documentación.
- Es libre.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5). (Otwell, 2016)

2.3.5 SINTAXIS

La sintaxis del lenguaje PHP es similar a la de C y Perl. PHP nos permite embeber sus fragmentos de código dentro de la página HTML, es decir, es posible incluir lenguaje PHP en un código HTML, delimitando nuestro código por etiquetas.

Existen 4 tipos de etiquetas, que indican a PHP dónde empezar y finalizar la interpretación del código ya que todo lo que esté fuera de las etiquetas de PHP será ignorado por el intérprete. En la Figura. 2 se muestran los tipos de etiquetas, de las cuales solo la primera y la cuarta están siempre disponibles y las otras dos pueden ser configuradas en el archivo php.in (Mehdi Achour & Jakub Vrana, 2016)



```
<?php y ?>
<? y ?>
<% y %>
<script language="php"> y </script>
```

Figura 2 Etiquetas de PHP

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Como podemos observar el tipo de etiqueta número tres, es un método creado para que tenga compatibilidad con el lenguaje ASP. (Otwell, 2016)

2.3.6 VARIABLES

Las variables aquí son representadas con un signo de dólar “\$”, seguido por el nombre de la variable, las cuales ya quedan declaradas en su primera aparición del código. Estas pueden ser de tipo entero, números en punto flotante, cadenas, array y de tipo objetos. Sin embargo, hay muchas formas de declarar variables, pero la forma correcta es que tienen que empezar con una letra (o si no empieza con una letra, tendrá que hacerlo con un carácter de subrayado), seguido de cualquier carácter. (Larrea, 2015)

2.3.7 COMENTARIOS

El uso de comentarios en PHP y en otro lenguaje es bastante recomendado para documentar el código que se esté haciendo, e indicando que hace cada línea y que hace cada función. PHP soporta comentarios C y C++ existiendo tres tipos los cuales son (Mehdi Achour & Jakub Vrana, 2016):

- /* Comentarios de varias líneas */
- // Comentarios de una sola línea
- # Comentarios estilo Bash / Perl (Hasta el final de línea)

2.3.8 TIPOS DE DATOS

El lenguaje PHP posee cuatro tipos de datos primitivos mostrados en la siguiente tabla:

Tabla 1: Tipos de datos primitivos en PHP

Boolean	Un valor que puede ser verdadero(true) o falso (false)
Int	Un valor numérico con signo
Float	Un valor numérico de punto flotante con signo
String	Una colección binaria de datos que puede ser un texto

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Además, cuenta con dos tipos de datos especiales:

Tabla 2: Tipos de datos especiales en PHP

Null	Representando la ausencia de un valor para una variable
resource	Recursos representados de forma no nativa por PHP, por ejemplo, conexiones a bases de datos y manejadores de archivo.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

PHP también posee tipos de datos compuestos:

Tabla 3: Tipos de datos compuestos en PHP

Array	Matriz o vector que puede contener diversos valores a la vez.
Object	Estructura con atributos y métodos.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

2.3.9 CONSTANTES

Las constantes son identificadoras o nombres para expresar un valor simple que no puede variar en la ejecución de un script. En PHP las constantes se declaran con la siguiente instrucción: (Larrea, 2015)

Define ('**NOMBRE_DE_CONSTANTE**'), valor_de_constante);

El nombre de la constante va entre comillas simples o dobles y el valor puede ser cualquier tipo primitivo. Además, se recomienda asignar nombres en mayúsculas a las constantes y separar las palabras con un guion bajo, omitiendo el símbolo \$(Tipos de datos, variables y constantes en PHP).

Los tipos de datos que soporta PHP son: Integer, float o double, String y boolean, tomando en cuenta que al asignar las variables no es necesario definir el tipo de dato al que pertenece. (Tipos de variables en PHP. Declaración y asignación. La función echo: mostrar texto en pantalla. (Mehdi Achour & Jakub Vrana, 2016)

2.4 HTML

HTML es el lenguaje que se emplea para el desarrollo de páginas de internet. Está compuesto por una serie de etiquetas que el navegador interpreta y da forma en la pantalla. HTML dispone de etiquetas para imágenes, hipervínculos que nos permiten dirigirnos a otras páginas, saltos de línea, listas, tablas, etc. (González, 2015)

2.4.1 DEFINICIÓN DE HTML

A HTML se define como a una implementación de estándar SGML (Standard Generalized Markup Language), estándar internacional para la definición de texto electrónico independiente de dispositivos, sistemas y aplicaciones. Metalenguaje para definir lenguajes de diseño descriptivos ya que proporciona un medio para codificar documentos hipertexto cuyo destino sea el intercambio directo entre sistemas o la aplicación. (Martin, 2016).

2.4.2 CARACTERÍSTICAS DE HTML

- Permite crear lenguajes de codificación descriptivos.
- Define una estructura de documentos jerárquica, con elementos y componentes interconectados.
- Proporciona una especificación formal completa del documento.
- No tiene un conjunto implícito de convenciones de señalización.
- Soporta, por tanto, un conjunto flexible de juegos de etiquetas.
- Los documentos generados por él son legibles.

2.4.3 EVOLUCIÓN DEL HTML

La plataforma de HTML 1 fue el primero en salir al mercado, sin embargo, no fue considerado un estándar a la hora de implementar esta versión, por lo que muchas veces se considera la versión de HTML 2 como la primera versión del HTML. (Castrejón, 2012).

HTML 2: Presentado en el año de 1995, y es considerado la primera versión del HTML, el cual no soportaba tablas y donde la declaración explícita de los elementos body, HTML y head que es opcional.

HTML 3.2: Esta versión se lanzó en el año de 1997 gracias al organismo de estandarización llamado W3C (World Wide Web Consortium). El cual incorporó los últimos avances de las páginas web.

HTML 4.0: Lanzada en 1998. Entre sus novedades más destacadas se encuentran las hojas de estilos CSS, la posibilidad de incluir pequeños programas o scripts en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios.

HTML 4.01: salió a la luz en 1999 como una revisión y actualización de la versión HTML 4.0, por lo que no incluye novedades significativas.

HTML 5: quinta revisión importante del lenguaje básico de la World Wide Web, el cual especifica dos variantes de sintaxis para HTML: el clásico HTML (text/HTML), y la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5. Este nuevo estándar aun no es tan utilizado pero su potencial es mucho mayor a las versiones anteriores, el cual incluye interesantes mejoras como nuevos elementos y etiquetas.

2.4.4 VENTAJAS DE HTML

- Es el lenguaje de formateo para los Navegadores Web.
- Es Fácil de entender y Como utilizar
- Su USO Es Muy Extendido (CHAIN, 2014)

2.5 MODELO, VISTA, CONTROLADOR

2.5.1 CONCEPTO DE MODELO, VISTA, CONTROLADOR

El Modelo Vista Controlador es un patrón de diseño de aplicaciones que se ha puesto muy de moda en los últimos años. Tiene varias razones, pero la más importante es separar el código en tres partes bien definidas: el modelo, la vista y el controlador. (Gaitán, 2016).

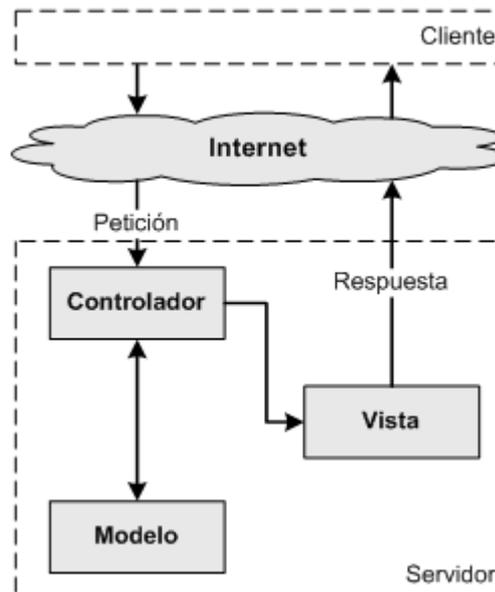


Figura 3: Diseño de MVC

Fuente: http://librosweb.es/libro/symfony_1_2/capitulo_2/el_patron_mvc.html2.4.2 Modelo

El modelo, son las clases encargadas de trabajar con las consultas de la base de datos, es decir que por cada tabla tendrá una clase, cada registro será un objeto y las consultas se llamarán a través de métodos de esas clases. A su vez Laravel trabaja con Eloquent que facilita el trabajo de las consultas a través de métodos ya establecidos. (Gaitán, 2016).

2.5.2 VISTAS

La vista es el producto final de una petición, el código HTML que se le devuelve al cliente, aquí no debería haber ninguna lógica, sin embargo, puede contener impresiones de variables, condicionales; pero no más que eso. La vista tiene un fin y es ése, entregar el código HTML de respuesta.

2.5.3 CONTROLADOR

El controlador son clases con métodos, también llamados acciones, estas acciones se comunican con los modelos para hacer consultas a la base de datos, y con las vistas para devolver una respuesta al cliente.

2.6 JAVASCRIPT

Existen procesos dentro de la aplicación del distributivo, como por ejemplo validaciones al momento de ingresar o actualizar datos, el lenguaje JavaScript se emplea para realizar estas validaciones de información y se podría describir como el complemento de HTML, debido a que interactúa directamente con éste, tomando la responsabilidad de los procesos que se desarrollarán en la aplicación a través del navegador. (CHAIN, 2014)

La función principal de este lenguaje es presentar una página Web dinámica, en la cual, los cambios o modificaciones realizadas en la aplicación web, se vean reflejadas de inmediato en el navegador.

JavaScript, tiene muchas posibilidades, es orientado a objetos, maneja funciones y estructuras de datos complejas, existen dos características básicas de los lenguajes orientados a objetos que JavaScript no implementa herencia y polimorfismo, aunque permite la creación y manipulación de objetos sencillos, además de la definición de métodos y propiedades para dichos objetos. La mayoría de los navegadores, en sus últimas versiones interpretan el código JavaScript integrado dentro de las páginas Web (Hazaël-Massieux, 2016)

2.7 CSS

2.7.1 DEFINICIÓN DE CSS

CSS se define como un lenguaje que permite realizar el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. (Alvarez, 2016).

2.7.2 EVOLUCIÓN DE CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML (Standard Generalized Markup Language), alrededor del año 1970. Desde la creación de

SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos, desde entonces se fue creando varias versiones de CSS que a continuación se dará a conocer:

CSS 1: Esta versión ya significó un avance considerable a la hora de diseñar páginas web, aportando mucho mayor control de los elementos de la página. Pero como todavía quedaron muchas otras cosas que los diseñadores deseaban hacer, pero que CSS no permitía especificar, éstos debían hacer uso de trucos para el diseño. Lo peor de esos trucos es que muchas veces implica alterar el contenido de la página para incorporar nuevas etiquetas HTML que permitan aplicar estilos de una manera más elaborada.

CSS 2: En esta versión incorporó algunas novedades interesantes, que hoy ya utilizan habitualmente, además en la versión CSS 3 todavía avanza un poco más en la dirección, de aportar más control sobre los elementos de la página.

Así pues, la novedad más importante que aporta CSS 3, a los desarrolladores webs, consiste en la incorporación de nuevos mecanismos para mantener un mayor control sobre el estilo con el que se muestran los elementos de las páginas, sin tener que recurrir a trucos, que a menudo complicaban el código del web.

2.8 BOOTSTRAP

2.8.1 CONCEPTO DE BOOTSTRAP

Bootstrap, es un framework originalmente creado por la empresa Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “**responsive design**” o diseño adaptativo.

2.8.2 EVOLUCIÓN DE BOOTSTRAP

En el año 2011, la herramienta Bootstrap creó como solución interna para solucionar las inconsistencias en el desarrollo dentro del equipo de ingeniería de Twitter. Básicamente, no se había establecido ninguna convención sobre las formas en las que los ingenieros de Twitter elegían cómo desarrollar la plataforma.

El desarrollo y la ingeniería web es un oficio, mucho incluso dicen que un arte, y cada ingeniero tiene su propio estilo de hacer las cosas. En algunos casos funciona, pero cuando hay multitud de ingenieros trabajando en el mismo proyecto con distintas formas de abordar los problemas, las inconsistencias son inevitables. Las inconsistencias en el desarrollo web pueden surgir problemas de codificación que generen incertidumbre y aumenten los costes de mantenimiento. Bootstrap fue una herramienta originalmente desarrollada por ingenieros Twitter. Mark Otto y Jacob Thorton. (Vinconti, 2015)

Aunque comenzó como una solución interna en Twitter, Mark y Jacob, pronto se dieron cuenta de que aquello tendría un gran potencial. En agosto de 2011, el framework Bootstrap fue lanzado al público como proyecto Open-Source en Github. En los siguientes meses, miles de desarrolladores de todo el mundo contribuyeron al proyecto y Bootstrap se convirtió en el proyecto Open-Source más activo del mundo. Desde entonces, Bootstrap solo ha ganado más notoriedad y se ha convertido en la herramienta de presentación más popular para desarrollar proyectos. (Tertre, 2014)

2.8.3 VENTAJAS Y DESVENTAJAS DE BOOTSTRAP

VENTAJAS:

- Utiliza componentes y servicios creados por la comunidad web, tales como: HTML, CSS, OOCSS, jQuery UI, LESS y GitHub.
- El uso de LESS, que es una ampliación a las famosas hojas de estilo CSS, pero a diferencia de éstas, funciona como un lenguaje de programación, permitiendo el uso de variables, funciones, operaciones aritméticas, entre otras, para acelerar y enriquecer los estilos en un sitio web.
- OOCSS, CSS orientado a objetos, que está organizado por módulos independientes y reutilizables en todo el proyecto.
- La herramienta es sencilla y ágil para construir sitios web e interfaces ya que minimiza el tiempo de desarrollo. (Lessin, 2014).

DESVENTAJAS:

- Es complicado, cambiar de versión si se realiza modificaciones profundas sobre el Core.
- Si necesita añadir componentes que no existen, debe hacerlo manualmente en CSS y cuidar de que mantenga coherencia con el diseño.
- Un problema que se presenta a menudo en muchos Themes, es que el responsive funciona bien, pero a nivel de diseño en ocasiones se le escapan ciertos detalles que le quitan estética al sitio. (Lessin, 2014).

2.9 POSTGRESQL

2.9.1 INTRODUCCIÓN

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales. (rafaelma, 2013).

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

A continuación, se representa en un gráfico que ilustra de manera general los componentes más importantes en un sistema PostgreSQL.

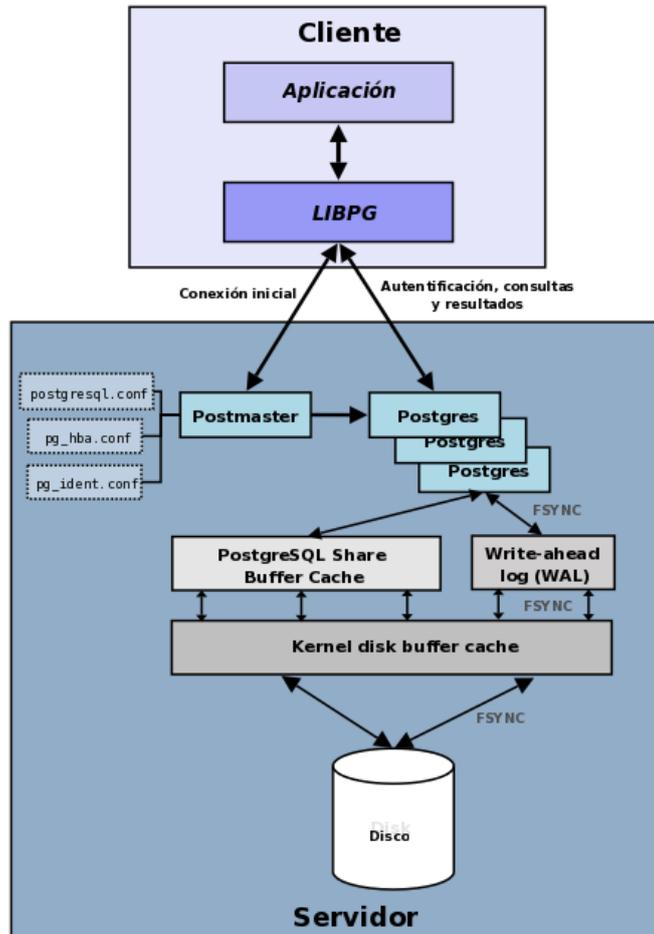


Figura 4: Componentes en un sistema PostgreSQL.

Fuente: http://www.postgresql.org.es/sobre_postgresql

- **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- **Demonio Postmaster:** Este es el proceso principal de PostgreSQL. Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autentificar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQLson: postgresql.conf, pg_hba.conf y pg_ident.conf.

- **Procesos hijos PostgreSQL:** Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes
- PostgreSQL share buffer cache: Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- Write-Ahead Log (WAL): Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)
- Kernel disk buffer cache: Caché de disco del sistema operativo
- Disco: Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione. (rafaelma, 2013).

2.9.2 CARACTERÍSTICAS

La última serie de producción es la 9.3. Sus características técnicas la hacen una de las bases de datos más potentes del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo.

Características Generales

- Es una base de datos 100% ACID(Reglas de Codd e Integridad de datos)
- Integridad referencial
- Tablespaces (es un almacén lógico de los ficheros de la base de datos)
- Nested transactions (savepoints)
- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- Two-phase commit
- PITR - point in time recovery
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado via SSL
- Actualización in-situ integrada (pg_upgrade)

- SE-postgres
- Completa documentación
- Licencia BSD
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

Programación / Desarrollo

- Funciones/procedimientos almacenados (stored procedures) en numerosos lenguajes de programación, entre otros PL/PostgreSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl
- Bloques anónimos de código de procedimientos (sentencias DO)
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido)
- APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.

SQL

- SQL92 SQL99, SQL2003, SQL2008
- Llaves primarias (primary keys) y foráneas (foreign keys)
- Check, Unique y Not null constraints
- Restricciones de unicidad postergables (deferrable constraints)
- Columnas auto-incrementales
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash o GiST
- Sub-selects
- Consultas recursivas
- Funciones 'Windows'
- Joins
- Vistas (views)

- Disparadores (triggers) comunes, por columna, condicionales.
- Reglas (Rules)
- Herencia de tablas (Inheritance)
- Eventos LISTEN/NOTIFY

Tabla 4: límites de PostgreSQL

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

Fuente: http://www.postgresql.org.es/sobre_postgresql

2.10 APLICACIÓN WEB

2.10.1 DEFINICIÓN DE APLICACIÓN WEB

Una aplicación web se define como a cualquier aplicación que es accedida vía web por una red como internet o una intranet. En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador (por ejemplo, un applet de Java) o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca la aplicación. (ALEGSA, 2015).

2.10.2 VENTAJAS

- Una de las ventajas de las aplicaciones web es que son cargadas desde internet (u otra red) y brinda la facilidad de mantener y actualizar dichas aplicaciones sin la necesidad de distribuir e instalar un software extra.
- También la posibilidad de ser ejecutadas en múltiples plataformas. (ALEGSA, 2015).

2.11 MÉTRICAS DE SOFTWARE

2.11.1 DEFINICIÓN.

Las métricas son aquella aplicación continua en todo proceso o etapa de la ingeniería de software, que relaciona o compara mediciones cuantitativas con un atributo que posee un sistema, proceso o componente. Con el objetivo de tener una mejor visión del proyecto, de la evaluación del producto y a su vez el nivel de aceptación, por medio de la cuantificación y gestión de una forma más efectiva de cada una de las variables a las que se necesita hacer un seguimiento (Vargas, 2016). Las magnitudes objeto de valoración son tres: la calidad, fiabilidad y productividad.

Por ende, las métricas de software utilizadas en un proyecto sirven para “ayudar en la estimación, control de calidad, evaluación y medición de la productividad y control de proyectos. Pero existen diferentes términos que componen este concepto como, medidas, indicadores, métricas que a continuación se definen cada una de ellas y sus diferencias:

Medida: Proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. Existen 4 razones para medir: caracterizar, evaluar, predecir, mejorar. Las medidas no sirven para comparar, ya que solo capturan una característica individual, la medición es la encargada de capturar esta medida. (Vargas, 2016). Por ejemplo 35.000 líneas de código (LDC).

Medición: Es el acto de determinar una medida. Ejemplo: María será la encargada de medir las LDC de cada módulo del sistema.

Métrica: En cambio las métricas son una medida cuantitativa que relaciona un valor numérico con una característica, es decir estas sirven para relacionar o comparar

medidas. Son el fundamento para los indicadores (Vargas, 2016) Ejemplo: La productividad de este proyecto fue de 500 líneas (LDC/persona-mes).

Indicadores: Es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso o proyecto del software, evaluando su estado y dar un seguimiento a los riesgos potenciales. (Vargas, 2016)

Ejemplo: La productividad media de una empresa es de 500 (LDC/pm).

2.11.2 IMPORTANCIA DE MEDIR SOFTWARE.

El uso de métricas en el desarrollo de software es una de las prácticas que menos beneficio emocional proporciona a corto plazo, pero mayores ventajas aportan a medio y largo plazo. (Rubio, 2013)

Tomando en cuenta que lo que no se puede medir no se puede mejorar, y en este caso es una tarea difícil la toma de datos por medio de métricas ya que la tarea de homogeneizarlos influye en muchos aspectos como si son varios o un solo proyecto, varios equipos de trabajo o las diferentes tecnologías que se utilizan. (Rubio, 2013) Los resultados cuantitativos siempre ofrecen el mejor resultado de una actividad y el uso de métricas sirve de ayuda para obtener estos resultados.

2.11.3 CARACTERÍSTICAS.

Una buena métrica debe poseer las siguientes características:

- Ser simples y definidas con precisión.
- Ser objetivas.
- De fácil obtención.
- Robustas.

Además, es importante que tengan una escala de medición apropiada. En especial se preocupan por evaluar la funcionalidad y el presupuesto de un proyecto. Sin embargo, existen otras 4 importantes características que afirman la utilidad de las métricas:

1. Cuantificables, las cuales deben basarse en hechos, no en opiniones.
2. Independientes, los recursos no deben poder ser alterados por los miembros que las apliquen o utilicen.
3. Explicable, debe documentarse información acerca de la métrica y de su uso.

4. Precisas, debe de conocerse un nivel de tolerancia permitido cuando se mide.
(Rubio, 2013)

2.11.4 MÉTRICAS DEL PROCESO Y DEL PROYECTO.

Se afirma que existen dos tipos de métricas: las métricas de producto y las métricas de proceso, y cabe recalcar que las métricas del proceso se extraen de las del proyecto en una recopilación. Refiriéndose como parte de las métricas de software a las métricas de proceso y producto. (Rubio, 2013)

2.11.4.1 MÉTRICAS DEL PROCESO

La medición del proceso implica la medición de las actividades relacionadas con el software como el esfuerzo, el coste y los defectos encontrados, con el objetivo de cuantificar el comportamiento del proceso. (Rubio, 2013).

Usualmente son objetivas, absolutas, explícitas y dinámicas y basadas en opinión de expertos, es decir que estas métricas estarán soportadas empíricamente. Siendo típicas las métricas que cuentan eventos como, por ejemplo: número de defectos detectados en inspección, número de requisitos modificados, Número de defectos de diseño detectados durante las pruebas. Se debe considerar además que las métricas del proceso son estratégicas ya que ayudan a determinar el curso del proceso de producción de software. (Vargas, 2016)

2.11.4.2 MÉTRICAS DEL PRODUCTO

A estas métricas se las definen como una medición del tamaño o cantidad de un producto que se produjo o se piensa producir. Desde los requerimientos hasta el sistema instalado. Hay varias razones para medir un producto.

De esta manera las métricas del producto deben medir en términos de calidad o productividad: la complejidad del diseño del software, el tamaño del programa final, o el número de páginas producidas por documentación.

2.11.5 CLASIFICACIÓN DE LAS MEDIDAS DE SOFTWARE.

El dominio de esta clasificación de estas medidas es de Proceso, Producto y Proyecto. La medición del software da una clasificación habitual la cual consiste en dividir las medidas de un atributo en dos tipos, directas e indirectas. (Rubio, 2013)

Medidas Directas: Estas medidas se obtienen sin la necesidad de utilizar otras medidas de otro atributo. Ejemplos de medidas directas en la ingeniería del software serían la longitud del código, costo, esfuerzo entre otras. (Rubio, 2013)

Medidas Indirectas: Son las medidas que se definen y calculan a partir de otras medidas directas de atributos. Ejemplo de estas podemos mencionar la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, productividad.

2.11.6 PRODUCTIVIDAD

2.11.6.1 DEFINICIÓN.

Productividad, es definida como las salidas producidas en relación a las entradas o recursos consumidos. Donde se considera a las salidas como el valor entregado y las entradas como los recursos gastados como por ejemplo el esfuerzo, o la influencia de factores externos como herramientas, complejidad, constantes de tiempo entre otras. Se cabe recalcar, que es mucho más fácil medir las entradas, que las salidas y su entorno.

La productividad del software es la relación entre el valor funcional del software producido y la mano de obra o gastos para producirlo (Rubio, 2013). En si el concepto de productividad de software viene a ser el resultado de la relación de lo producido por el programador y el esfuerzo empleado para producirlo, ambas siendo medidas directas de software. De acuerdo a la clasificación de medidas de software mencionada anteriormente, la productividad viene a ser parte de una medida indirecta, convirtiéndose en una métrica debido a que depende de otras medidas directas las cuales son las líneas de código y el esfuerzo o tiempo.

2.11.6.2 FACTORES QUE AFECTAN A LA PRODUCTIVIDAD.

Existen 4 factores importantes que inciden en la productividad del software. (Crespo., 2014)

Factor humano: Este factor va enfocado en el tamaño del equipo de trabajo, la capacidad individual, años de experiencia en cierto lenguaje de programación o en problemas similares entre otros.

Factor del proceso: dentro de este factor se incluye los lenguajes de programación, ambientes de trabajo, estrategias de programación, revisiones de diseño y código, herramientas case o frameworks, entre otros.

Factor del producto: incluye aspectos de fiabilidad y rendimiento del sistema como por ejemplo tamaño del proyecto y base de datos individualmente, requerimientos de tiempo real, complejidad, cantidad de código reusado, cantidad de documentación, entre otros.

Factor del recurso: Disponibilidad de herramientas CASE, y recursos.

Este estudio comparativo se centrará justamente en el factor del proceso, ya que los frameworks de estudio utilizan funcionalidades diferentes y al medir su productividad se obtendrá los resultados.

2.11.7 MÉTRICAS DE PRODUCTIVIDAD

La importancia de medir la productividad de productos de software, viene de la necesidad de identificar que recursos se están sobre utilizando y causan perdida para la empresa (Vargas, 2016)

Las métricas de productividad de software (standard-ieee-std-1045-ieee-standard-software-productivity-metrics, 2016), se definen a la métrica orientada al tamaño, Aunque existen otros modelos de métricas propuestas, pero la antes mencionada es las más usadas en cuanto al producto de software.

2.11.7.1 MÉTRICAS ORIENTADAS AL TAMAÑO.

Estas métricas se calculan dividiendo la medición del software desarrollado o entregado de conformidad, por el tiempo empleado siendo otra forma para medir la productividad de un producto de software. Donde las salidas utilizan como medida de tamaño al número de líneas de código, aunque existen otras medidas propuestas como número de clases, numero de métodos, número de páginas de documentación, número

de casos de uso, o requerimientos.(standard-ieee-std-1045-ieee-standard-software-productivity-metrics, 2016).

2.11.7.2 LÍNEAS DE CÓDIGO (LDC)

Las líneas de código es una medida física del producto de software, sin embargo, se debe definir qué se debe contar y que no. Es aquí donde aparecen dos opciones, las cuales son 1) si contar líneas de comentarios o no y 2) si contar líneas de código o sentencias. Definiendo a esta medida física como una medida de líneas de código no en blanco. (standard-ieee-std-1045-ieee-standard-software-productivity-metrics, 2016).

CAPITULO III

ANÁLISIS COMPARATIVO ENTRE LOS FRAMEWORK

LARAVEL Y CODEIGNITER

3.1 FRAMEWORKS DE OPEN SOURCE

Hoy en día las grandes empresas enfrentan una creciente competencia, y para poder estar al mismo nivel de las otras empresas toman otras alternativas para mejorar la productividad, seguridad, calidad y la flexibilidad el uso de software libre es una de las mejores opciones para las empresas, ya que los resultados son más eficientes a un menor costo, con el fin de incrementar la satisfacción de los clientes y la rentabilidad. Cuán mayor sea la productividad, seguridad, calidad y la flexibilidad de una empresa, más útil será para la comunidad gracias a que ésta se expande y genera empleo.

De acuerdo con Gartner, los frameworks representan un punto fundamental a la hora de desarrollar un software. Tomando en cuenta que un framework de Open-Source tiene muchas ventajas convincentes para el desarrollo. A continuación, detallamos las razones que evidencian el por qué compensa usar un Software hecho con frameworks de Open-Source para las empresas:

Seguridad. Es uno de los puntos más importantes para una empresa, de salvaguardar las informaciones que son muy valiosas. Los framework Open-Source tiene la posibilidad de adaptarse a los requerimientos necesarios para el desarrollo del software, brindando mayor seguridad y disminuyendo los riesgos de puertas traseras que introduzcan códigos maliciosos o de espionaje. Además, puede cifrar sus datos de manera que solo pueden ser revisados con una clave o contraseña que mantienes secreto y seguros.

Fiabiles. Existen buenas razones por las cuales se debe utilizar un software hecho con frameworks Open-Source. Una de las razones es que el software libre involucra a toda la comunidad en una labor conjunta para corregir los errores. Los usuarios no solo informan de errores, también los corrigen y envían correcciones. Trabajan juntos, para llegar al fondo del problema y hacer que el software funcione sin problemas.

Además, podemos mencionar que Según Gartner, en su informe Predicts 2011: Open Source Software, the Power Behind the Throne, menciona que, en este año, el 99% de

las empresas incluidas en el Global 2000 enterprises de Forbes están usando software libre. Este es el caso, por poner solamente dos ejemplos, de Apache y Linux. IBM tiene en su cartera de productos servidores UNIX, donde el cliente puede elegir entre AIX (el sistema operativo UNIX de IBM) o Linux. No hay ninguna distinción entre ellos. Son igual de fiables y se da el mismo nivel de servicio por parte de IBM. Pero Linux avanza más rápidamente que AIX y además tiene la ventaja de estar presente en todos los servidores, cosa que no le pasa a AIX, por lo cual para el cliente es una clara ventaja el no ligarse a un solo fabricante.

¿Qué mueve a las empresas a pasarse al software libre? Sin duda el cálculo del TCO es un factor determinante (el Total Cost of Ownership es un indicador para calcular el coste de un producto en una organización, desde el momento de su adquisición, hasta el mismo día en que se deja de usar). Aunque es obvio que tener un producto de software libre no significa coste cero en una empresa, el coste total suele ser menor que en producto propietarios.

Va a ser sin duda muy excitante, observar el progreso del software libre en las grandes empresas, donde ya es una realidad y todo hace pensar que lo seguirá siendo cada vez más.

Mario Gómez Molina es tutor y consultor del Máster Universitario de Software Libre de la UOC. Es además Project Manager de la consultora VASS menciona que en el desarrollo de un sistema informático lo más importante a tener en cuenta por parte de los desarrolladores, es la selección de la herramienta de desarrollo porque de esta depende el tiempo, facilidad, costo, entre otros aspectos para la implementación del sistema.

3.1.1 ANÁLISIS COMPARATIVO ENTRE LOS FRAMEWORK PHP: LARAVEL Y CODEIGNITER

Para el desarrollo de esta investigación se tomó como referencia los resultados de la investigación de Vilcaguano Zumba María Ibeth y Tierra Llamuca Juan Manuel, con experiencia en Análisis Comparativo de frameworks PHP (Vilcaguano Zumba, 2015) y en base las normas métricas de calidad 1045 (Vargas, 2016). Además, en el criterio de del investigador de Software Libre de la UOC Mario Gómez Molina.

En este capítulo se realiza el análisis comparativo de los frameworks PHP: Laravel y CodeIgniter con el fin de determinar la herramienta que brinda mayor productividad,

dando así una base que permitirá seleccionar la herramienta más adecuada para el desarrollo del Sistema de Gestión de concursos de méritos y oposición de la “Universidad Nacional de Chimborazo”.

3.2 LARAVEL

3.2.1 DEFINICIÓN

Laravel, es un framework Open-Source para desarrollar en PHP, con una filosofía muy clara enfocada para que el código sea lo más expresivo y elegante posible, para desarrollar aplicaciones y servicios web. (McDade, 2016)



Figura 5: logo del framework Laravel

Fuente: <http://gitsolution.com/laravel-development>

3.1.2 HISTORIA DEL FRAMEWORK LARAVEL

En el año 2011, el creador de Laravel Taylor Otwell vio la carencia de una importante funcionalidad en el framework CodeIgniter, la cual pensó que era indispensable en la construcción de aplicaciones. Es por eso que para completar esta característica lanzó la primera versión beta de Laravel en junio del mismo año, aun no siendo MVC, poseía otras funcionalidades como Eloquent ORM para operación en la base de datos, localización, modelos, sesiones, vistas, sesiones y mucho más. (McDade, 2016).

La versión 2 de Laravel apareció seis meses después, con una sintaxis limpia y simple para el desarrollo y la adición de convertirse en un framework MVC completo. En febrero del 2012 Laravel 3 fue lanzado enfocándose en integración de pruebas unitarias, la interfaz de línea de comandos Artisan, migración de base de datos, eventos, drivers de sesiones y bases de datos. Llegando a la actual versión 4 apodada Illuminate, presentando una diferente arquitectura del núcleo del framework.

La versión de Laravel 5.0 introduce una estructura nueva solicitud para el proyecto Laravel predeterminado. Esta nueva estructura sirve como una mejor base para la construcción de una aplicación robusta en Laravel, así como (PSR-4 Estándar de Auto carga) que cuentan en toda la aplicación.

Laravel 5.1 sigue las mejoras realizadas en Laravel 5.0 mediante la adopción (PSR-2 Codificación Guía de estilo) y la adición de retransmisión de eventos, parámetros de middleware, las mejoras del artesano, y mucho más.

Laravel 5.2 sigue las mejoras realizadas en Laravel 5,1 mediante la adición de soporte para múltiple controlador de autenticación, modelo implícito de unión, simplificados ámbitos global elocuentes, opt-in andamios autenticación, grupos de middleware, que limita la velocidad de middleware, las mejoras de validación de matriz, y mucho más. (McDade, 2016)

3.1.3 CARACTERÍSTICAS DEL FRAMEWORK LARAVEL

Entre las principales características tenemos:

- **Modular y extensible:** Permite agregar todo lo que necesitas a través de su directorio Packalyst que cuenta con más de 5,500 paquetes donde puedes encontrar lo que necesitas implementado.
- **Micro-servicios y APIs:** Lumen es un micro-framework derivado de Laravel para desarrollar de forma fácil y rápida micro-servicios y APIs. Lumen integra todas las características de Laravel con una mínima configuración.
- **HTTP routing:** Posee un sistema de enrutamiento rápido y eficiente, similar al que se usa en Ruby on Rails. Este nos permite relacionar las partes de nuestra aplicación con las rutas que ingresa el usuario en el navegador.
- **HTTP Middleware:** El uso de Middleware se encarga de analizar y filtrar las llamadas HTTP en tu servidor. Puedes instalarlo para que se encargue de verificar que se trate de un usuario registrado, de evitar problemas de tipo Cross-Site-Scripting (XSS) y otras medidas de seguridad.
- **Caché:** Robusto sistema de caché configurable.
- **Autenticación:** Autenticación de usuarios de forma nativa.
- **Integración con Stripe:** Laravel Cashier incluye todo lo necesario para integrar tu desarrollo con este servicio de cobro.

- **Tareas automatizadas:** Con el API Elixir de Laravel se pueden crear tareas de Gulp con las que podemos definir el uso pre-procesadores para comprimir nuestro CSS y JavaScript.
- **Encriptación:** Uso de seguridad OpenSSL y cifrado AES-256-CBC.
- **Eventos:** Definir, registrar y escuchar eventos en tu aplicación es muy sencillo
- Paginación
- Object-Relational-Map (ORM)
- **Unit Testing:** Unit Testing usando PHPUnit
- **Cola de tareas (Queues):** Opción de ejecutar procesos largos y complejos en segundo plano usando listas de tareas.

3.1.4 FILOSOFÍA.

La filosofía de Laravel se centra en la escritura de código PHP de forma elegante y simple, evitando lo que se conoce como (código espagueti), es decir evitar que el código sea incomprensible y complejo. El resultado de aplicaciones creadas con Laravel son aplicaciones basadas en PHP 5, con un código sencillo de entender. Sobre todo, interesante para leer el código, que al contrario cuando no se usa un framework, la comprensión toma tiempo. (Otwell, 2016). Siendo así el propósito de Laravel, de hacer agradable el proceso de desarrollo para el programador sin sacrificar.

3.1.5 ARQUITECTURA

Laravel sigue el patrón de arquitectura MVC tradicional, el cual hace cumplir una separación entre la lógica de negocios, de la lógica de entradas y presentación asociada con una interfaz gráfica de usuario (GUI). (McDade, 2016). Mientras tanto en las aplicaciones web creadas por Laravel, la lógica de negocios típicamente consiste de los modelos de datos como por ejemplo usuarios, publicaciones en blogs, y se podría decir que la interfaz gráfica es solo una página web en el navegador.

3.1.6 CAPAS MVC EN LARAVEL

3.1.6.1 CAPA DEL MODELO

La capa modelo en cualquier otro MVC framework, es el componente responsable de la comunicación de la base de datos. Es aquí donde interviene Eloquent que es una clase que se hereda a los modelos en Laravel, para trabajar de manera simple

con los objetos de la base de datos. Obviamente el uso de Eloquent es opcional, pues también este framework dispone de otros recursos que facilita interactuar con los datos, o específicamente la creación de modelos. (McDade, 2016)

3.1.6.2 CAPA DE VISTA

La capa vista en Laravel, usa su propio motor de plantillas llamado Blade, con el objetivo de dar un código limpio y fácil de comprender en las Vistas. Utilizado código PHP plano y permite incluir un sistema de Caché que lo hace mucho más rápido, para la creación de tal manera estas plantillas se debe crear el archivo dentro de la carpeta /views con la extensión nombrevista.blade.php (Otwell, 2016)

3.1.6.3 CAPA DEL CONTROLADOR

El controlador es donde la lógica de negocio está ubicada y gracias a esto se tienen funcionalidades como: recuperar todas las entradas de la base de datos para enumerarlas, actualizar, eliminar o realizar búsquedas de la información en las tablas del base de datos dado un identificador, añadir nuevos registros a la base de datos y crear mensajes de confirmación de las funciones básicas. Esta capa permite organizar el código en clases sin tener que escribirlo todo en las rutas. Todos los controladores deben extenderse de la clase BaseController. (Otwell, 2016)

3.1.6.4 CICLO DE PETICIÓN MVC EN LARAVEL

Al momento de interactuar con una aplicación desarrollada por Laravel, la primera acción que se realiza es cuando el navegador envía una petición al servidor web, y este se la envía al motor de enrutamiento. El siguiente paso es que este router redirecciona esta petición hacia la clase del controlador apropiado según el patrón URL del mismo. Es aquí donde la clase del controlador realiza su trabajo, siendo el tercer paso la interacción con el modelo, el cual es el encargado de comunicarse con la base de datos. Después de ser invocado el modelo, el controlador interpreta la vista final y regresa la página completa al navegador el usuario con toda la información solicitada. (McDade, 2016)

3.1.7 ELOQUENT ORM

Eloquent ORM es una librería del framework Laravel que se caracteriza como (Object-Relational Mapping), y que proporciona una manera sencilla de interactuar con

la base de datos agilitando a relacionar los modelos de la aplicación. Entre los usos que se le puede brindar con la librería Eloquent ORM son:

- Las funciones básicas CRUD (Create, Read, Update, Delete).
- Configuración y uso de la relación uno a uno.
- Configuración y uso de la relación uno a muchos.
- Configuración y uso de la relación muchos a muchos. (McDade, 2016)

El Eloquent ORM proporciona una gran implementación con Active Record para trabajar con la base de datos. Es decir que al momento de crear un modelo bajo la estructura MVC, este se corresponde directamente a una tabla en la base de datos. Por ejemplo, un modelo llamado Email corresponde a la tabla emails, permitiendo así la manipulación de información de una manera fácil. (Otwell, 2016)

3.1.8 REQUERIMIENTOS

Los requerimientos son mínimos para el uso del framework Laravel en su última, se detalla a continuación (Otwell, 2016)

3.1.8.1 LENGUAJE DE PROGRAMACIÓN.

Obviamente para el uso de este framework, se debe tener instalado el lenguaje PHP en su versión 5.3.7.

3.1.8.2 EXTENSIÓN DE MCRYPT PHP.

Es una interfaz para la biblioteca Mcrypt, que admite una gran variedad de algoritmos de bloques. En la mayoría de servidores web, en el caso de apache que ya viene incluido.

3.1.8.3 COMPOSER

Es un administrador de dependencias para los proyectos.

3.1.8.4 SERVIDOR WEB.

El servidor, Apache es compatible con Nginx el cual genera URLs amigables.

3.1.8.5 SERVIDOR DE BASES DE DATOS

Actualmente Laravel soporta cuatro sistemas de bases de datos los se menciona a continuación: MySQL, Postgres, SQLite, y SQL Server.

3.2 CODEIGNITER

3.2.1 DEFINICIÓN FRAMEWORK CODEIGNITER.

A CodeIgniter se define como a un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. Su objetivo principal es permitir que los desarrolladores puedan realizar proyectos mucho más rápido. Esta herramienta brinda un conjunto de bibliotecas para tareas comunes, además brinda una interfaz simple y una estructura lógica para acceder a las bibliotecas. Además, es un producto de código libre, que se le puede dar libre uso para cualquier aplicación. (Lerdorf, 2016)

También se debe destacar que CodeIgniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba al framework CodeIgniter porque es rápido y ligero, para la implementación del proceso de desarrollo llamado Model View Controller (MVC), que es un estándar de programación para aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales.



Figura 6: logo del framework CodeIgniter

Fuente: <http://desarrollolibre.net/blog/tema/242/codeigniter/template-parser-class-en-codeigniter>

3.2.2 HISTORIA DEL FRAMEWORK CODEIGNITER

El framework CodeIgniter fue desarrollado originalmente por Rick Ellis. Con el propósito de obtener buen rendimiento en el mundo real, donde muchas de las bibliotecas de clases, helpers, y subsistemas se tomaron prestados del código base de

ExpressionEngine. Actualmente, el Equipo de Desarrollo de ExpressionEngine lo desarrolla y mantiene. De tanto que actualmente la última versión estable a la fecha de elaboración del presente proyecto es la versión 3.0 que sirve de base para la elaboración, del módulo para su respectivo análisis comparativo de este trabajo de investigación. (Lerdorf, 2016)

3.2.3 CARACTERÍSTICAS GENERALES DE CODEIGNITER

Las características en sí mismas son una forma muy pobre para juzgar una aplicación, ya que le dicen nada acerca de la experiencia del usuario, o cómo intuitivamente o inteligentemente se ha diseñado. Características no revelan nada sobre la calidad del código, o el rendimiento, o la atención al detalle, o las prácticas de seguridad. La única manera de juzgar realmente una aplicación es probarlo y conocer el código. A continuación, esta las principales características de CodeIgniter. (Lerdorf, 2016)

- **Versatilidad:** Quizás la característica principal de CodeIgniter, en comparación con otros frameworks PHP. CodeIgniter es capaz de trabajar la mayoría de los entornos o servidores, incluso en sistemas de alojamiento compartido, donde sólo tenemos un acceso por FTP para enviar los archivos al servidor y donde no tenemos acceso a su configuración.
- **Facilidad de instalación:** No es necesario más que una cuenta de FTP para subir CodeIgniter al servidor y su configuración se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos, que no suelen estar disponibles en todos los alojamientos.
- **Flexibilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces nos las podemos saltar para trabajar como más a gusto encontremos. Algunos módulos como el uso de plantillas son totalmente opcionales. Esto ayuda muchas veces también a que la curva de aprendizaje sea más sencilla al principio.
- **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código. La

mayoría de los módulos o clases que ofrece se pueden cargar de manera opcional, sólo cuando se van a utilizar realmente.

- **Documentación:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida, cuando ya sabemos acerca del framework y queremos consultar sobre una función o un método en concreto, pero para iniciarnos sin duda se agradece mucho.

3.2.4 ARQUITECTURA DEL FRAMEWORK CODEIGNITER

Desde el punto de vista técnico y arquitectónico, CodeIgniter fue creado con los siguientes objetivos: (Lerdorf R. , 2016)

- **Dinámica de instancias.** En CodeIgniter, los componentes se cargan y las rutinas ejecuta sólo cuando se le solicite, en lugar de a nivel mundial. No se hacen suposiciones por el sistema en cuanto a lo que puede ser necesario más allá de los recursos básicos mínimos, por lo que el sistema es muy ligero por defecto. Los eventos, que se activa por la petición HTTP, y los controladores y vistas que diseñe determinarán lo que se invoca. (SÁNCHEZ & VERA , 2011)
- **Acoplamiento.** El acoplamiento es el grado en que los componentes de un sistema dependen unos de otros. Los menos componentes dependen unos de otros más reutilizable y flexible.
- **Componente Singularidad.** Singularidad es el grado en el que los componentes tienen un propósito estrechamente enfocado. En CodeIgniter, cada clase y sus funciones son muy autónomas con el fin de permitir la máxima utilidad. CodeIgniter es un sistema dinámicamente una instancia, débilmente acoplado con un alto componente de singularidad. Se esfuerza por simplicidad, flexibilidad y alto rendimiento en un paquete pequeño huella. (Lerdorf R. , 2016).

3.3.5 MODELO - VISTA - CONTROLADOR EN CODEIGNITER

El Modelo, Vista, Controlador es típicamente utilizado para la creación de aplicaciones web y no sólo CodeIgniter lo implementa, sino también otra serie de frameworks de desarrollo web, en PHP u otros lenguajes. Es interesante porque separa en varios grupos las complejidades de las distintas partes que componen una página web, como la vista y la lógica, así como el acceso a la base de datos. (Lerdorf R. , 2016)

Quizás lo que más obligue a cambiar estos hábitos de programación en PHP es el hecho de tener que basar nuestros scripts en este modelo de programación, porque fija un nuevo estilo de desarrollo de aplicaciones, que obliga a separar código fuente según su ámbito. Sin embargo, estas nuevas costumbres de codificación también ayudan a que los programas sean mejores y disfruten de varias ventajas como ser más organizados, escalables, reutilizables y entendibles por otros desarrolladores, lo que resultara mayor facilidad en el mantenimiento.

Para los que no lo conocen, el Modelo - Vista - Controlador es un patrón de desarrollo o un estilo de arquitectura de software que separa el código fuente de las aplicaciones en tres grupos:

Modelo: Todo el código que tiene que ver con el acceso a base de datos. En el modelo mantendrá encapsulada la complejidad de la base de datos y simplemente creará funciones para recibir, insertar, actualizar o borrar información de nuestras tablas. Al mantenerse todas las llamadas a la base de datos en un mismo código, desde otras partes del programa se puede invocar las funciones que necesita del modelo y éste se encargará de procesarlas. (Lerdorf, 2016) En el modelo se podrá preocupar cosas como el tipo de base de datos con la que se trabaja o las tablas y sus relaciones, pero desde las otras partes del programa simplemente las funciones del modelo sin importar qué tiene que hacer éste para realizar las acciones invocadas.

Vista: La vista codifica y mantiene la presentación final de la aplicación de cara al usuario. Es decir, en la vista se coloca todo el código HTML, CSS, Javascript, entre otros, que se tiene que generar para producir la página como requiera el usuario, en la práctica la vista no sólo sirve para producir páginas web, sino también cualquier otra salida que se quiera enviar al usuario, en formatos o lenguajes distintos, como pueden ser feeds RSS, archivos JSON, XML, y otros.

Controlador: El controlador se podría decir que es la parte más importante, porque hace de enlace entre el modelo, la vista y cualquier otro recurso que se tenga que procesar en el servidor para generar la página web. En resumen, en el controlador se guarda la lógica de las páginas y se realiza todas las acciones que sean necesarias para generarlas, ayudados del modelo o la vista. (Lerdorf R. , 2016).

3.3 DETERMINACIÓN DE PARÁMETROS DE COMPARACIÓN

Para el análisis comparativo de los frameworks PHP: Laravel y CodeIgniter, se va a considerar varios parámetros de programación, cada hito seleccionado es importante para el cálculo de la productividad en el desarrollo del sistema de Gestión de concursos de méritos y oposición de la “Universidad Nacional de Chimborazo”, se ha establecido tres parámetros cada uno con sus respectivos indicadores, mismos se seleccionaron por los autores de la presente investigación, en base a los aspectos relevantes y características más importantes que se toman en cuenta al desarrollar una aplicación Web.

Para una mejor comprensión se describe más adelante cada uno de los parámetros con sus respectivos indicadores a mayor detalle.

3.3.1 NÚMERO DE LÍNEAS DE CÓDIGO

Para crear una aplicación web mediante un framework es necesario que el mismo proporcione el menor número posible en cuanto a inserción de código, y el código generado en esta herramienta sea útil en su totalidad. Por lo cual se ha visto necesario evaluar este parámetro para determinar cuál de los framework PHP: Laravel o CodeIgniter, brinda mejor productividad. Tabla 5.

Tabla 5: Descripción de los indicadores del Parámetro: Número de líneas de código

NÚMERO DE LÍNEAS DE CÓDIGO	
INDICADORES	DESCRIPCIÓN
Módulo de subida de archivos de los postulante	Líneas de código utilizado para implementar el modelo, vista y controlador del módulo archivos de subida del postulante.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.3.2 TIEMPO DE RESPUESTA DEL SISTEMA

El tiempo de respuesta del sistema es aquel que está comprendido entre el inicio de una consulta a la base de datos de PostgreSQL o demanda sobre la aplicación web y la visualización o repuesta de los datos. El tiempo de respuesta debe ser valores límites normales para que el usuario logre sentir la relación que existe en el sistema de manera instantánea, por su importancia se evalúa este parámetro para determinar la

productividad, lo que significa que a menor tiempo de respuesta es mejor pues demuestra eficiencia y efectividad. Tabla 6.

Tabla 6: Descripción de los indicadores del Parámetro: Tiempo de respuesta del sistema

TIEMPO DE RESPUESTA DEL SISTEMA	
INDICADORES	DESCRIPCIÓN
Tiempo de respuesta transacción de inserción de datos de los postulantes.	Identificar el tiempo que se tardan los frameworks PHP: Laravel y CodeIgniter al ingresar datos con un número de postulantes.
Tiempo de respuesta en transacciones de actualización de datos de los postulantes.	Identificar el tiempo que se tardan los frameworks PHP: Laravel y CodeIgniter en editar datos del postulante.
Tiempo de respuesta en transacciones de consulta de datos de los postulantes.	Identificar el tiempo que se tardan los frameworks PHP: Laravel y CodeIgniter en listar datos con un número de postulantes.
Tiempo de respuesta en transacciones de eliminación de los datos del postulante.	Identificar el tiempo que se tardan los DreamWorks PHP: Larval y CodeIgniter en eliminar datos con un número de postulantes concurrentes.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.3.3 MANEJO DE SEGURIDAD

La seguridad es otro punto importante en una aplicación web porque la información que se maneja está a disposición de los usuarios de la web, es necesario mantener un nivel de seguridad para evitar pérdidas o violaciones en nuestra información. Para evaluar este parámetro se toma en cuenta los principales aspectos que intervienen en la seguridad de datos; roles, autenticación y autorización. Con el análisis se determinar el framework que brinde mayor productividad en la implementación de la seguridad del sistema.

Tabla 7: Descripción de los indicadores del Parámetro: Manejo de seguridad

MANEJO DE SEGURIDAD	
INDICADORES	DESCRIPCIÓN
Uso de sesiones	Disponibilidad de “helpers” o ayudantes para disminuir el tiempo de desarrollo de aplicaciones en base a sesiones.
Manejo de autenticación	facilita la autenticación de usuarios, es decir el login/logout de usuarios
Uso de ACL	Facilidad de uso de las librerías de Listas de Control de Acceso (ACL), en el desarrollo de aplicaciones seguras.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.4 PONDERACIÓN DE LOS INDICADORES DE LOS PARÁMETROS DE COMPARACIÓN

Para lograr determinar la productividad en cada uno de los frameworks, se definió una ponderación para la valoración de cada parámetro con sus respectivos indicadores. Esta ponderación se lo realizó en base al criterio de los investigadores, tomando en cuenta el nivel de importancia de cada uno.

Mientras más útil sea el código en la programación menor será el número de líneas de código, por tanto, se dio un peso de 40% al módulo de subida de archivos para los postulantes ya que es de suma importancia este módulo para el sistema ya que su programación es importante porque debe estar estructurada correctamente, dando como resultado total, un peso porcentual del 40% para el parámetro número de líneas de código.

De igual manera el tiempo de respuesta del sistema es importante pues los usuarios realizan múltiples solicitudes a la página web del sistema de gestión de concursos de mérito y opción de la Universidad Nacional de Chimborazo y es necesario saber cuán rápida es dicha aplicación al momento de realizar Transacciones de inserción, actualización, consultas y eliminación que se efectúan con mayor frecuencia y se requiere un nivel alto de eficiencia en la inserción de datos, por lo tanto, se asigna un peso del 15% respectivamente.

Transacciones de actualización de datos se efectúan con menor frecuencia por lo tanto se asigna un peso del 5%. Transacciones de consultas se efectúa con frecuencia y se asignó con 15%. Las transacciones de eliminación se efectúan rara vez por lo tanto se

asigna un peso del 5%. Dando como resultado total un peso porcentual del 40% para el parámetro tiempo de respuesta del sistema.

Mantener un nivel medio de seguridad es importante para evitar pérdidas o violaciones en nuestra información, por tanto, se dio un peso del 15% a Manejo de autenticación, 5%. Uso de ACL y 5% Uso de sesiones, pues todos ellos tienen el mismo nivel de importancia al implementar el modulo seguridad del sistema, dando como resultado total un peso porcentual del 20% para el parámetro manejo de seguridad.

El resultado final es el 100 % de ponderación para la presente investigación. En la Tabla 8 se muestra el peso asignado a cada uno de los indicadores de los parámetros de comparación establecidos anteriormente, está dado por porcentajes.

Tabla 8: Ponderación de los Indicadores de los Parámetros de comparación

PARÁMETRO	INDICADOR	PESO	TOTAL
Número De Líneas De Código	I1: Módulo de archivos	40%	40%
	Tiempo de respuesta del sistema	I2: Tiempo de respuesta durante el proceso de inserción de datos de los postulantes.	15%
	I3: Tiempo de respuesta durante el proceso de actualización de datos de los postulantes.	5%	
	I4: Tiempo de respuesta durante el proceso de consulta de los datos de los postulantes.	5%	
	I5: Tiempo de respuesta durante el proceso de eliminación de datos de los postulantes.	15%	
Manejo de seguridad.	I6: Uso de sesiones	5%	20%
	I7: Manejo de autenticación	10%	
	I8: Uso de ACL (Listas de Control de Acceso)	5%	

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.5 INSTRUMENTOS DE MEDICIÓN

Los instrumentos de medición permiten evaluar los indicadores de los parámetros de comparación planteados anteriormente, sometiendo a prueba a los prototipos de cada

framework PHP: Laravel y CodeIgniter. Estos instrumentos se han seleccionado en base a los parámetros de comparación. Las herramientas a emplearse son:

Quick Line Counter QLC: es un contador de Líneas de Código (LOC). Permite contar Líneas de código, líneas en blanco y las líneas de comentario se cuentan por separado. Cuenta línea de código en C ++, C #, C, JAVA, JSP, BASIC, HTML, XML, CSS, AS, Ruby y programas PHP.

JMeter: es una herramienta de carga diseñada para realizar Pruebas de Rendimiento y Pruebas Funcionales sobre Aplicaciones Web.

Características

- Permite el manejo de datos de entrada para las pruebas.
- Las pruebas que realizan son de forma rápida.
- Permite realizar pruebas de testing a la interfaz gráfica GUI.
- Presenta una interfaz gráfica amigable, al momento en que lanzan los errores.
- Ver de distintas formas los resultados de la ejecución de un test plan (vía listeners)

Para visualizar los resultados de forma más clara cada transacción realizada de inserción, actualización de listado, consulta y eliminación de usuarios, se utiliza los siguientes receptores: el Informe Agregado y el Reporte Resumen.

Reporte Resumen: Este permite visualizar los resultados del test realizado, en una tabla. Presentando los siguientes datos:

- **Etiqueta:** Etiqueta de la muestra
- **#Muestras (#n):** Cantidad de peticiones utilizadas para la URL.
- **Media (\bar{x}):** Tiempo promedio en milisegundos para un conjunto de resultados.
- **Min:** Tiempo mínimo que demora un thread en acceder a una página.
- **Max:** Tiempo máximo que demora un thread en acceder a una página
- **Media en bytes:** Tamaño medio de respuesta del servidor (en bytes).

Informe Agregado: Siendo similar al anterior, este permite obtener resultados más precisos. Presenta los siguientes datos:

- **Etiqueta:** Etiqueta de la muestra.
- **#Muestras (#n):** Cantidad de peticiones utilizadas para la URL.
- **Media (\bar{x}):** Tiempo promedio en milisegundos para un conjunto de resultados.
- **Mediana (Me):** Valor en tiempo del percentil 50.
- **%Error:** Porcentaje de requerimientos con errores.
- **Min:** Tiempo mínimo de la muestra de una determinada URL.
- **Max:** Tiempo máximo de la muestra de una determinada URL.

3.6 DESCRIPCIÓN DEL ESCENARIO DE PRUEBAS

Los escenarios de pruebas en este caso de los dos sistemas, permiten verificar y obtener datos de manera que ayudan a determinar que Framework es el mejor en productividad en base a los parámetros de comparación, los cuales serán implementados bajo el modelo de arquitectura de desarrollo llamado MVC (modelo, vista, controlador), en los dos frameworks seleccionados Laravel y CodeIgniter.

Para realizar las pruebas de las aplicaciones, utilizamos tanto Hardware Como Software:

Tabla 9: Herramientas de Hardware Como Software para realizar las pruebas.

Hardware:	Software
Laptop Core I7.	Windows 10
Procesador: 3.4 GHz	Java JDK
RAM: 8GB	Line Of Code QLC
Tarjeta de Vídeo: 1GB	Xampp
Disco Duro de 1Tb	Navegador
	JMeter 2.11

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.7 SISTEMA UTILIZANDO EL FRAMEWORK LARAVEL

Es una aplicación Web desarrollada con el Framework Laravel bajo la arquitectura flexible MVC (Modelo, Vista, controlador), la misma que consta de Módulo de Administración de usuarios, Módulo categorías y concurso, Módulo de matrícula para los postulantes y el Módulo de archivos.

Los principales componentes de la arquitectura Modelo, Vista, Controlador (MVC) del sistema con el Framework Laravel se visualiza en el Anexo A.

En la Figura 7 se muestra la página principal del prototipo con el Framework Laravel, para la autenticación y administración de los usuarios.

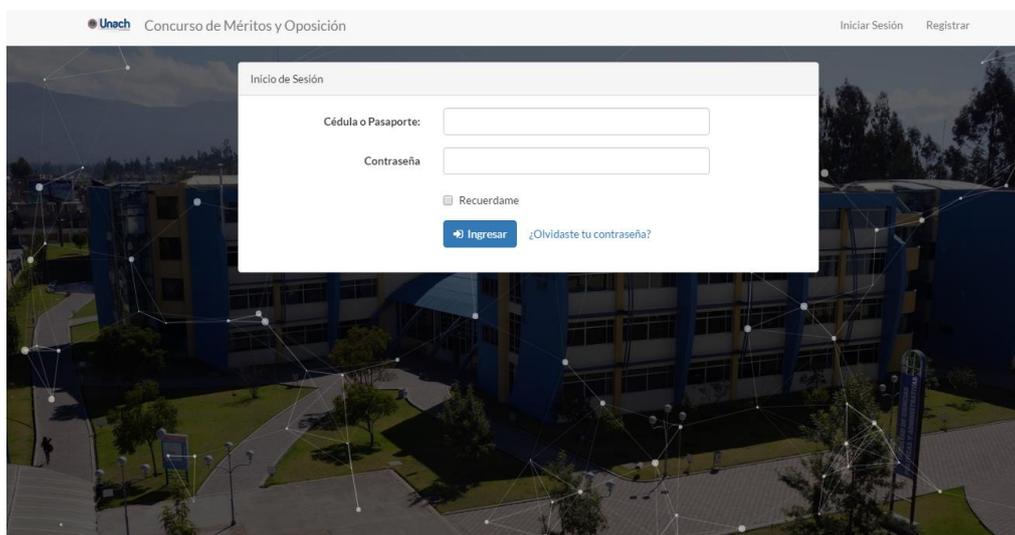


Figura 7: Interfaz Principal del prototipo con el Framework Laravel

Elaborado por: Diego Chicaiza & Jhonatan Arcos

En la Figura 8, se muestra la página de listado de usuarios del sistema con el Framework Larval, que permite validar el resultado de las pruebas ejecutadas en Meter.

3.8 SISTEMA UTILIZANDO EL FRAMEWORK CODEIGNITER

Es una aplicación Web desarrollada con el framework CodeIgniter bajo la arquitectura flexible MVC (Modelo Vista controlador), la misma que consta de Módulo de Administración de usuarios, Módulo categorías y concurso, Módulo de matrícula para los postulantes y el Módulo de archivos.

Los principales componentes de la arquitectura Modelo, Vista, Controlador (MVC) del sistema con el framework CodeIgniter se visualiza en el Anexo A.

En la Figura 8 se muestra la página principal del sistema con el framework CodeIgniter.



Figura 8: Interfaz Principal del sistema con el Framework CodeIgniter

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.9 DESARROLLO DE LAS PRUEBAS CON LOS PARÁMETROS DE COMPARACIÓN

En cada sistema de los frameworks PHP: Laravel y CodeIgniter se ejecuta las pruebas correspondientes a cada uno de los parámetros de comparación: número de líneas de código, tiempo de respuesta del sistema, manejo de seguridad.

Los resultados obtenidos de las pruebas de cada parámetro realizadas en cada sistema serán analizados entre los dos frameworks PHP: Laravel y CodeIgniter, comprobados, graficados e interpretados, con ello se determina que Framework proporciona mejor productividad y elegir el adecuado para la implementación del sistema de gestión de concursos de méritos y oposición en la “Universidad Nacional De Chimborazo”.

3.10 PARÁMETRO 1: NÚMERO DE LÍNEAS DE CÓDIGO

El parámetro de Número de Líneas de Código posee el indicador de la (Tabla 5), de los cuales mantuvo tres pruebas diferentes (Modelo, Vista, Controlador). Los resultados expuestos serán evaluados tomando sus valores mínimos y máximos para cada indicador.

3.10.1 RESULTADOS

Después de la ejecución de las diferentes pruebas, se obtuvo la cantidad de líneas de código (Ver Anexo A) validas, descritos en la Tabla 8.

Tabla 10: Resultados del Parámetro 1: Número de líneas de código

NÚMERO DE LÍNEAS DE CÓDIGO								
Indicador	Framework Laravel (Taylor Otwell)				Framework CodeIgniter (Lerdorf, Rasmus)			
	M	V	C	Total	M	V	C	Total
Módulo de subida de archivos para los postulantes.	19	34	52	105	42	45	58	145

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.10.2 CÁLCULO DE PESOS

En base a los resultados de la Tabla 10, se calcula el peso porcentual del indicador, para ello se asigna el máximo peso porcentual al Framework que tenga menor número de líneas de código y para el mayor número de líneas de código se aplica la regla de tres inversas, el valor se detalla en la Tabla 11.

Tabla 11: Peso porcentual de los indicadores del Parámetro 1: Número de líneas de código

Parámetro	Indicador	Peso porcentual del framework Larval	Peso porcentual del framework CodeIgniter	Peso Porcentual Máximo
Número de líneas de código	I1	40%	28.96%	40%
Total Porcentual:		40%	28.96%	40%

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.10.3 GRÁFICO ESTADÍSTICO

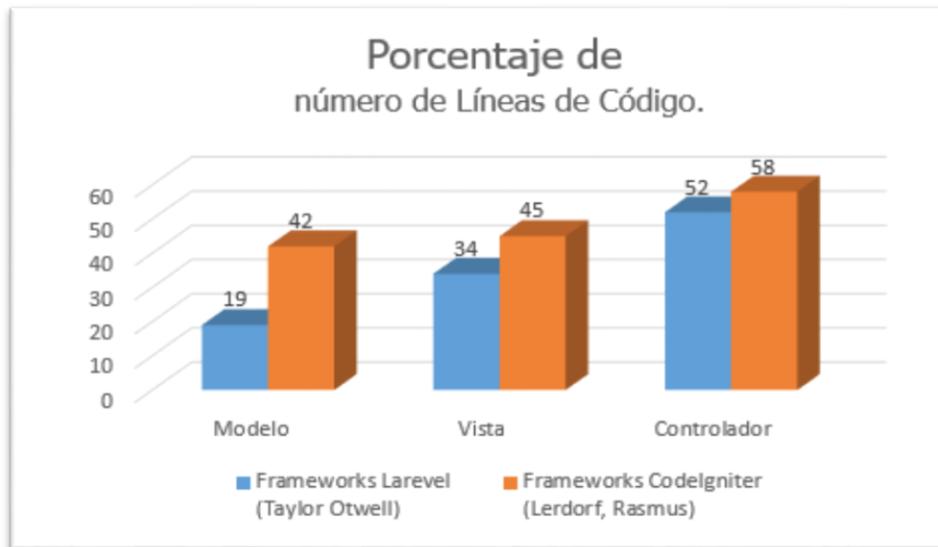


Figura 9: resultado de las líneas de código de los dos Frameworks PHP: Laravel y CodeIgniter

Elaborado por: Diego Chicaiza & Jhonatan Arcos

En la figura 12, se muestra los resultados porcentuales del nivel de cumplimiento de los frameworks: Laravel y CodeIgniter, con respecto al parámetro Número de líneas de código como en el modelo, vista y controlador.

3.10.4 GRÁFICO ESTADÍSTICO DE LOS RESULTADOS FINALES DEL PARÁMETRO 1.

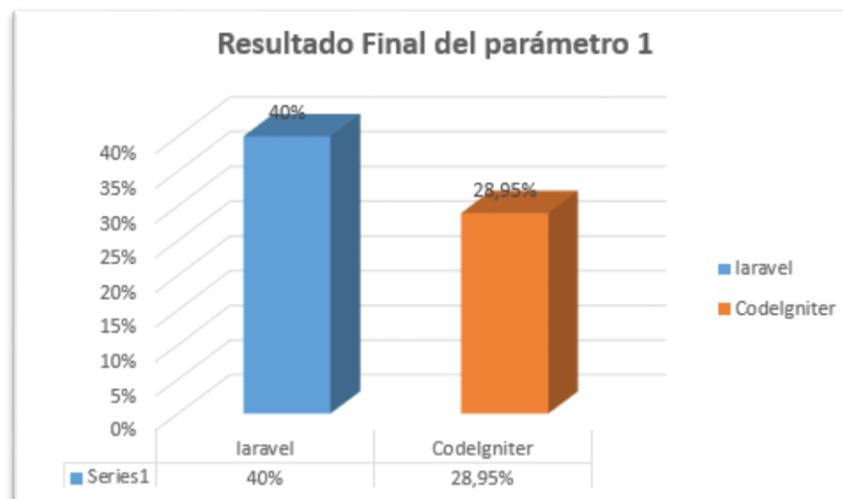


Figura 10: Resultado Final del parámetro 1.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.11 PARÁMETRO 2: TIEMPO DE RESPUESTA DEL SISTEMA.

El parámetro Tiempo de Respuesta del Sistema posee cuatro indicadores (Tabla 6). Los resultados obtenidos serán evaluados tomando mucho en cuenta sus valores mínimos y máximos para cada indicador.

Para hacer las pruebas a la inserción, actualización, consulta y eliminación de datos de los postulantes en cada uno de los sistemas, se tomó en cuenta los siguientes aspectos que a continuación se presenta:

- prueba con las 385 peticiones al servidor.
- Agregar al grupo de hilos, el número de peticiones http realizadas al sistema.
- Agregar dentro del grupo de hilos receptores como: Reporte Resumen e Informe Agregado, para poder observar los resultados.

3.11.1 TIEMPO DE RESPUESTA DEL SISTEMA DE TRANSACCIÓN DE INSERCIÓN DE DATOS DE LOS POSTULANTES.

Luego de ejecutar de las pruebas con el software Jmeter, se obtuvieron los valores (Anexo B) correspondientes al tiempo de respuesta del sistema de transacción de inserción de datos de los postulantes. En la Tabla 12, se indica en resumen los resultados obtenidos tanto en el informe agregado como en el reporte resumen.

Tabla 12: Resultados del indicador Tiempo de respuesta del sistema en transacciones de inserción

Framework	#n	\bar{x} (s)	Min(s)	Max(s)	Error %	S(ms)
Laravel	385	5,16	1,691	8,433	0,00%	1,5383
CodeIgniter	385	7,006	0,11	46,272	0,00%	13,3863

Elaborado por: Diego Chicaiza & Jhonatan Arcos

De los datos obtenidos se observa que el tiempo promedio de respuesta del sistema en transacciones de inserción de datos de los postulantes en el Framework Laravel es 5,16(s) y en el Framework CodeIgniter es 7,006(s); Framework CodeIgniter muestra mayor dispersión en los tiempos de repuestas del sistema en transacciones de inserción en relación al framework Laravel.

En base a las ponderaciones (Tabla 8), al framework Laravel en inserción se dio el máximo peso porcentual 15% ya que tiene el menor tiempo y para el Framework

CodeIgniter que tiene el mayor tiempo se aplica la regla de tres inversas, dando el 11,04% como peso porcentual.

3.11.2 TIEMPO DE RESPUESTA DEL SISTEMA EN TRANSACCIONES DE MODIFICACIÓN DE DATOS DE LOS POSTULANTE.

Después de haber realizado la ejecución de las pruebas con el software Jmeter, se obtuvieron los valores (Anexo B) correspondientes al tiempo de respuesta del sistema en transacciones de actualización de datos del postulante. En la Tabla 12, se indica en resumen los resultados obtenidos en el reporte resumen y el informe agregado.

Tabla 13: Resultados del indicador Tiempo de respuesta del sistema en transacciones de modificación de datos de los postulantes.

Framework	#n	$\bar{x}(s)$	Min(s)	Max(s)	Error %	S(s)
Laravel	385	4,993	2,022	8,624	0,00%	0,8213
CodeIgniter	385	6,607	0,068	53,971	0,00%	14,2672

Elaborado por: Diego Chicaiza & Jhonatan Arcos

De los datos obtenidos se observa que el tiempo promedio en el Framework Laravel es 4,993(s) y en el Framework CodeIgniter es 6,607(s); de tal Manera que en el Framework CodeIgniter se ve mayor dispersión en los tiempos de repuestas del sistema durante el proceso de actualización de datos de los postulantes, en relación al framework Laravel.

En base a las ponderaciones (Tabla 8), se dio al Framework Laravel peso del 5% ya que tiene el menor tiempo en comparación con el framework CodeIgniter por tal razón al framework CodeIgniter se aplica la regla de tres inversas, dando el 3.8% como peso porcentual.

3.11.3 TIEMPO DE RESPUESTA DEL SISTEMA EN TRANSACCIONES DE CONSULTA DE DATOS DE LOS POSTULANTES.

Después de la ejecución de las pruebas con el software Jmeter, se obtuvieron los valores (Anexo B) correspondientes al tiempo de respuesta del sistema en transacciones de consulta de los datos de los postulantes. En la Tabla 13, se indica en resumen los resultados obtenidos del reporte resumen y del informe agregado.

Tabla 14: Resultados del indicador Tiempo de respuesta del sistema en transacciones de consulta

Framework	#n	$\bar{x}(s)$	Min(s)	Max(s)	Error %	S(s)
Laravel	385	5,094	2,399	8,199	0,00%	0,7995
CodeIgniter	385	7,416	0,127	35,829	0,00%	11,4347

Elaborado por: Diego Chicaiza & Jhonatan Arcos

De los datos obtenidos se observa que el tiempo promedio en el framework Laravel es 5,094(s) y en el framework CodeIgniter es de 7,416(s); igualmente en el framework CodeIgniter se ve mayor dispersión en los tiempos de repuestas del sistema en transacciones de consulta en relación al framework Laravel.

En base a la ponderación realizada en la (Tabla 8), se dio al framework Laravel en transacciones de consulta de los datos de los postulantes un peso del 5%, ya que tiene el menor tiempo y para el framework CodeIgniter que tiene el mayor tiempo se aplica la regla de tres inversas, dando el 3,43% como peso porcentual.

3.11.4 TIEMPO DE RESPUESTA DEL SISTEMA EN TRANSACCIONES DE ELIMINACIÓN DE DATOS DE LOS POSTULANTES.

Después de la ejecución de las pruebas con el software Jmeter, se obtuvieron los valores (Anexo B) correspondientes al tiempo de respuesta del sistema en transacciones de eliminación. En la Tabla 14, se indica en resumen los resultados obtenidos del reporte resumen y del informe agregado.

Tabla 15: Resultados del indicador Tiempo de respuesta del sistema en transacciones de eliminación de datos de los postulantes.

Framework	#n	$\bar{x}(s)$	Min(s)	Max(s)	Error %	S(s)
Laravel	385	4,806	1,966	8,343	0,00%	1,5237
CodeIgniter	385	7,109	0,085	40,267	0,00%	12,4804

Elaborado por: Diego Chicaiza & Jhonatan Arcos

De los datos obtenidos se observa que el tiempo promedio en el framework Laravel es 4,806(s) y en el framework CodeIgniter es de 7,109 (s); igualmente en el framework CodeIgniter se ve mayor dispersión en los tiempos de respuesta del sistema en

transacciones de eliminación de datos de los postulantes en relación al framework Laravel.

En base a la ponderación realizada en la (Tabla 8), se dio al framework Laravel en eliminación de datos de los postulantes un peso del 15% ya que tiene el menor tiempo en comparación con el framework CodeIgniter que tiene el mayor tiempo se aplica la regla de tres inversas dando el 10,14% como peso porcentual.

En la (Tabla 29-3) se detallan los resultados del cumplimiento de cada uno de los indicadores establecidos para el parámetro tiempo de respuesta del sistema.

Tabla 16: Peso porcentual de los indicadores del Parámetro 6: Tiempo de Respuesta del Sistema

Parámetro	Indicador	Peso Porcentual Del Framework Laravel	Peso Porcentual Del Framework CodeIgniter	Peso Porcentual Máximo
Tiempo de Respuesta del Sistema	I2	15%	11,04%	15%
	I3	5%	3,8%	5%
	I4	5%	3,43%	5%
	I5	15%	10,14%	15%
Total, Porcentual:		40%	28,41%	40%

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.11.5 GRÁFICO ESTADÍSTICO

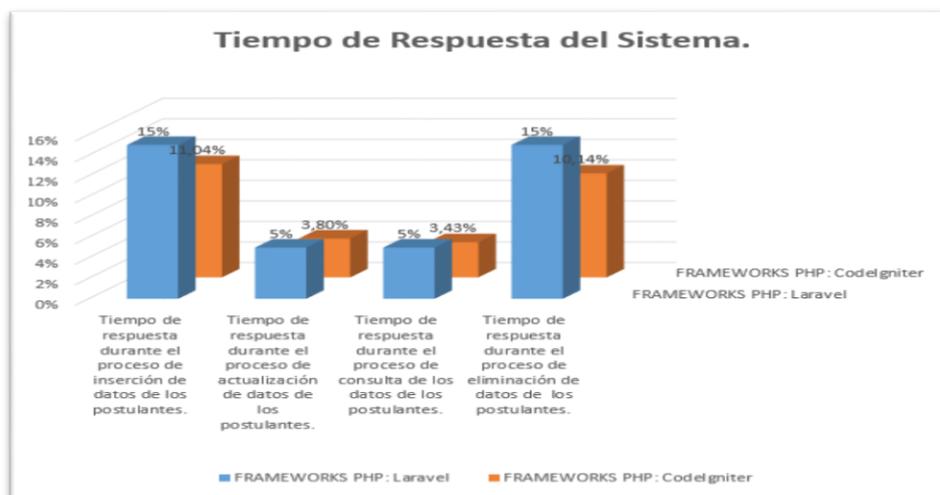


Figura 11: Resultado por indicador del Parámetro 2: Tiempo de Respuesta del Sistema.

3.11.6 GRÁFICO ESTADÍSTICO



Figura 12: Resultado Final del Parámetro 2: Tiempo de Respuesta del Sistema.

Elaborado por: Diego Chicaiza & Jhonatan Arco

3.11.7 DESCRIPCIÓN DE RESULTADOS

En base a la Tabla, se observa que al responder a 55 usuarios concurrentes realizando 385 peticiones, durante el proceso de inserción de datos de los postulantes. el framework Laravel es más productivo con un peso porcentual del 15% en relación al framework CodeIgniter que tiene un peso porcentual del 11.04%; durante el proceso de actualización de datos de los postulantes el framework Laravel es más productivo con un peso porcentual del 5% en relación al framework CodeIgniter que tiene un peso porcentual del 3,80%; en consulta el framework Laravel es más productivo durante el proceso de consulta de los datos de los postulantes con un peso porcentual del 5% en relación al framework CodeIgniter que tiene un peso porcentual del 3,43%; durante el proceso de eliminación de datos de los postulantes el framework Laravel es más productivo con un peso porcentual del 15% el cual es mayor en relación al framework CodeIgniter con un peso porcentual del 10.14%.

Como se indica en la Figura 10, el framework Laravel demuestra mejor productividad en tiempo de repuesta del sistema con un peso porcentual total del 40% de

cumplimiento máximo, mientras que el framework CodeIgniter presenta un cumplimiento total del 28,41%.

3.11 PARÁMETRO 3: MANEJO DE SEGURIDAD.

3.11.1 MANEJO DE SEGURIDAD.

El parámetro Manejo de Seguridad del Sistema posee tres indicadores (Manejo de autenticación, Uso de ACL y Uso de sesiones), se evaluará la facilidad de su implementación, pues esto permite que su desarrollo sea más productivo.

Tabla 17: Escala de Evaluación Parámetro 3: Manejo de seguridad

Valor Cualitativo	Valor Cuantitativo	Calificación Porcentual	DESCRIPCIÓN
No Satisfactorio	1	$\geq 0\%$ y $\leq 25\%$	Ninguna expectativa cumplida
Poco Satisfactorio	2	$> 25\%$ y $\leq 50\%$	Pocas de las expectativas cumplidas
Satisfactorio	3	$> 50\%$ y $\leq 75\%$	Mayoría de las expectativas cumplidas
Muy Satisfactorio	4	$> 75\%$ y $\leq 100\%$	Todas las expectativas cumplidas

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.11.2 CALIFICACIÓN

La calificación se obtendrá en base a la Tabla 16, teniendo en cuenta el nivel de facilidad que permitió hacer productivo la implementación de Seguridad. Ver Anexo C.

Tabla 18: Calificación del parámetro 5: Manejo de Seguridad

INDICADORES	EL MANEJO DE SEGURIDAD	
	Laravel	CodeIgniter
Manejo de autenticación	4	3
Uso de ACL	4	2
Uso de sesiones	4	3

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.11.3 CÁLCULO DE PESOS

En base a los resultados de la Tabla 17, se calcula el peso porcentual de cada indicador, para ello se asigna el máximo peso porcentual al Framework que tenga mayor calificación y

para la menor calificación se aplica la regla de tres inversas, los valores son detallados en la Tabla 18.

Tabla 19: Peso porcentual de los indicadores del Parámetro 3: Manejo de Seguridad

Parámetro	Indicador	Peso Porcentual del framework Laravel	Peso Porcentual del framework CodeIgniter	Peso Porcentual Máximo
Manejo de Seguridad	I6	5%	3,75%	5%
	I7	10%	5%	10%
	I8	5%	3,75%	5%
Total, Porcentual:		20%	12.5%	20%

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.11.4 GRÁFICO ESTADÍSTICO

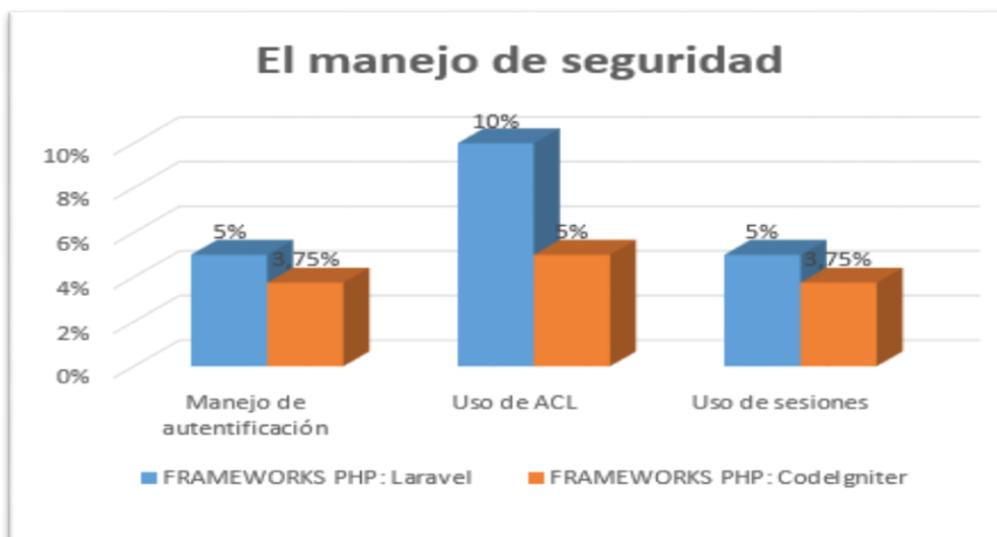


Figura 13: Resultado por indicador del Parámetro 3: Manejo de Seguridad.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.11.5 GRÁFICO ESTADÍSTICO

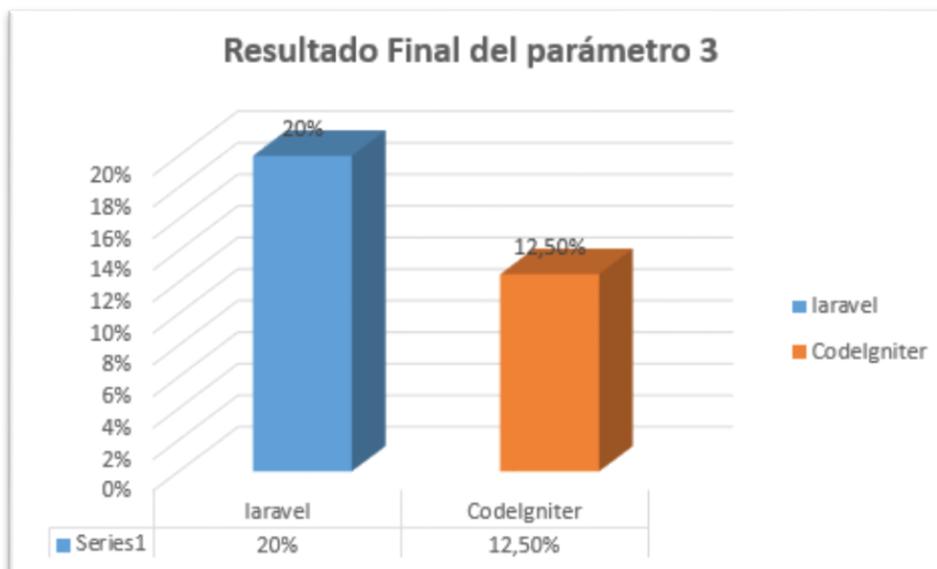


Figura 14: Resultado Final del Parámetro 3: Manejo de Seguridad.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.12 ANÁLISIS DE RESULTADOS

Luego de haber realizado las pruebas con los parámetros de comparación, se presenta los resultados obtenidos en forma porcentual en la Tabla 22, los mismos que serán interpretados.

Tabla 20: Resultados Finales de los Parámetros de Comparación

Parámetro	FRAMEWORK		Máximo Peso Porcentual
	%Laravel	%CodeIgniter	
Número de líneas de código	40%	28.95%	40%
Tiempo de respuesta del sistema.	40%	28.41%	40%
Manejo de seguridad.	20%	12.5%	20%
TOTAL	100%	68.86%	100%

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.12.1 GRÁFICO ESTADÍSTICO

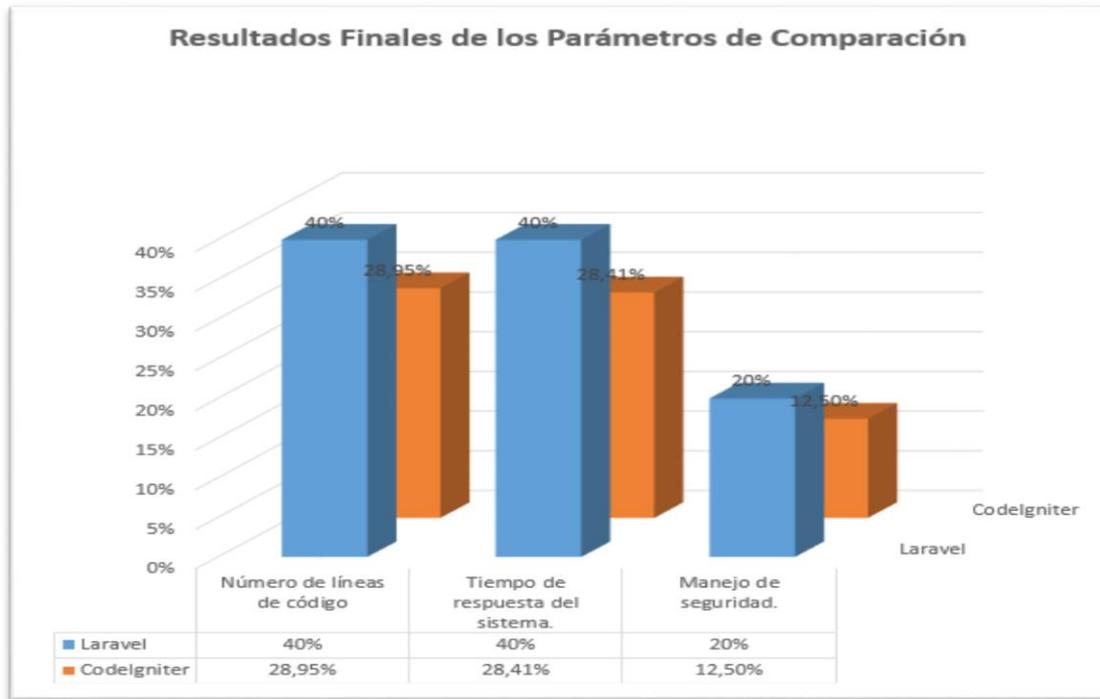


Figura 15: Resultados porcentuales de cada parámetro de comparación.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

3.12.2 GRÁFICO ESTADÍSTICO

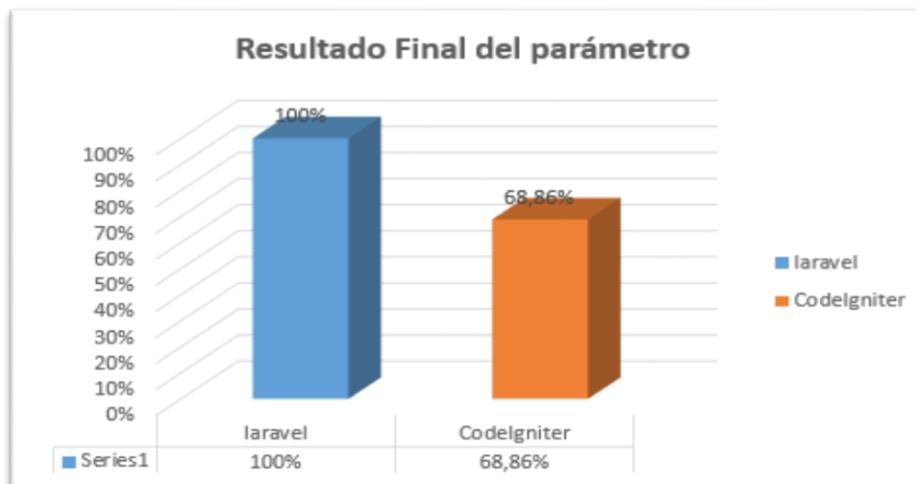


Figura 16: Resultado Final del Análisis

Elaborado por: Diego Chicaiza & Jhonatan Arcos

De acuerdo a los resultados obtenidos después de haber realizado el respectivo análisis comparativo de los frameworks PHP: Laravel y CodeIgniter mejora la productividad en

el desarrollo del Sistema de Gestión de concursos de méritos y oposición en la Universidad Nacional De Chimborazo, cumpliendo con el 100% de los parámetros establecidos para el análisis el mismo que es mayor, en comparación al 68,86% de cumplimiento del Framework CodeIgniter.

CAPITULO IV

IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE PROCESOS DE CONCURSOS DE MÉRITOS Y OPOSICIONES EN LA UNIVERSIDAD NACIONAL DE CHIMBORAZO CON EL FRAMEWORKS SELECCIONADO.

El framework de PHP más apropiada para el desarrollo de aplicaciones web con respecto a la productividad es Laravel. Por tal razón este framework de componentes se utilizará para el desarrollo del Sistema de Gestión Concursos de mérito y oposición con una metodología de desarrollo de software eXtreme Programming (XP).

El sistema se desarrolló en la Universidad Nacional de Chimborazo, y lo utilizara las cuatro Facultades de la institución antes mencionada, el mismo que ayudara a realizar la calificación de postulantes a docentes de forma automática, rápida y eficiente mediante la implementación del sistema informático.

Todos estos temas, conceptos, metodologías y características serán tratados en el transcurso del presente capítulo y el desarrollo del sistema en sí.

4.1 METODOLOGÍA XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, XP se basa en realimentación continua entre el cliente y el equipo de desarrollo. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

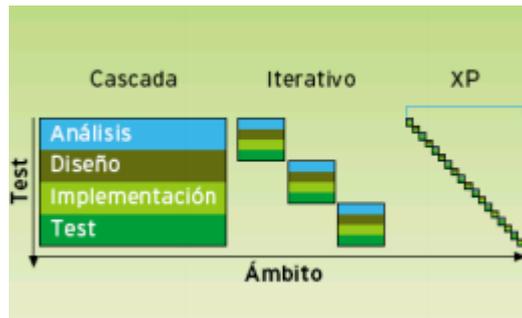


Figura 17: Fases de la Metodología XP

Fuente: <http://bit.ly/1dUm0Yp>

Características XP

- Metodología basada en prueba y error
- Fundamentada en Valores y Prácticas
- Expresada en forma de 12 Prácticas–Conjunto completo–Se soportan unas a otras–Son conocidas desde hace tiempo. La novedad es juntarlas

4.1.1 PROPÓSITO

El propósito de la Especificación es definir de manera clara, precisa, ordenada y verídica todas las funcionalidades y restricciones del sistema a desarrollarse.

El documento debe reflejar en las aspiraciones para el sistema de gestión de procesos de concurso de méritos y oposiciones para la Universidad Nacional De Chimborazo, como también de los usuarios finales para evitar problemas a futuro.

Es decir, las especificaciones de este documento están sujetas a continuas revisiones por parte de los Investigadores como del tutor, hasta alcanzar su aprobación.

Luego de esta etapa el documento servirá de base para que los Investigadores puedan construir el nuevo sistema.

4.1.2 ÁMBITO DEL SISTEMA

El desarrollo del sistema es de suma importancia para la institución para realizar la calificación de postulantes a docentes de forma automática, rápida y eficiente mediante la implementación de un sistema informático, este futuro sistema recibirá el nombre de Sistema de gestión de concursos de Méritos y Oposiciones Académica SIGCOM.

Se debe tener en cuenta que el Sistema de Méritos y Oposiciones Académica SIGCOM. Ofrece varios servicios, los mismos que pueden ser utilizados por el personal encargado como administrador del sistema, por los postulantes al concurso, y la comisión de evaluación.

4.1.3 DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

Definiciones:

Tabla 21: Definiciones

Administrador	Persona encargada de la manipulación de toda la información del sistema.
Postulantes	Personas que utilizan los servicios del Sistema para el respectivo concurso.
Comisión de evaluación.	Son las personas encargadas de evaluar al postulante durante el proceso de concursos de méritos y oposiciones.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Abreviaturas:

Tabla 22: Abreviaturas

UNACH	Universidad Nacional de Chimborazo
CISYC	Carrera de Ingeniería en Sistemas y Computación
SIGCOM	Sistema de gestión de procesos de concursos de méritos y oposiciones para la Universidad Nacional De Chimborazo,

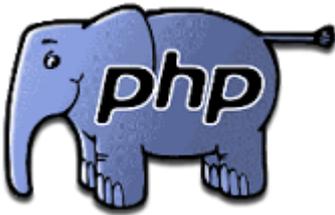
Elaborado por: Diego Chicaiza & Jhonatan Arcos

4.2 DESARROLLO DEL SISTEMA

4.1.1 HERRAMIENTAS DE DESARROLLO

Para la implementación del Sistema de Gestión de concursos de Méritos y Oposiciones SIGCOM.se utilizará las siguientes tecnologías y herramientas.

Tabla 23: Herramientas de Desarrollo para SIGCOM

HERRAMIENTA	CONCEPTO	VERSION UTILIZADA
<p>PHP</p> 	<p>PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.</p>	<p>PHP 7.0.8</p>
<p>Framework PHP: Laravel</p> 	<p>Laravel es un framework PHP desarrollado por Taylor Otwell, que promete llevar al lenguaje PHP a un nuevo nivel.</p>	<p>Laravel 5.2</p>
<p>Postgres</p> 	<p>Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado.</p>	<p>Postgres9.3</p>
<p>Bootstrap</p> 	<p>Bootstrap, es un framework que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice.</p>	<p>Bootstrap 2.3.2</p>

Elaborado por: Diego Chicaiza & Jhonatan Arcos

4.3 GESTIÓN DEL PROYECTO

4.3.1 PLANIFICACIÓN DEL PROYECTO

Esta planificación del proyecto se realizó tras el estudio del problema y los requerimientos, mediante la representación de las historias se efectuó la planificación inicial la cual fue variando en el transcurso de la misma cambiando y mejorando las historias en base a concepción del problema.

4.3.2 INTEGRANTES Y ROLES

Con la participación del Director del proyecto, los miembros, los usuarios y desarrolladores, se formará el equipo encargado de la implementación del sistema.

Esto implicara que los diseños deberán ser sencillos y claros, los usuarios dispondrán de versiones de prueba del software para que puedan participar en el proceso de desarrollo mediante sugerencias y aportaciones, dicho equipo de trabajo se ve ilustrado en la Tabla 24 definiendo Integrantes y Roles.

Tabla 24: Integrantes y Roles

MIEMBRO	GRUPO	ROLES XP	METODO LOGÍA
Diego Chicaiza	investigador	Rastreador, Testeador, Programador	Xp
Jhonatan Arcos	investigador	Rastreador, Testeador, Programador	
Javier Haro	Tutor	Consultor	

Elaborado por: Diego Chicaiza & Jhonatan Arcos

4.3.3 DESCRIPCIÓN DEL SISTEMA

El Sistema de Méritos y Oposiciones Académica SIGCOM, realiza las siguientes funciones, registro de usuario, ingreso de requisitos, concurso de mérito, concurso de oposición, registro de postulantes, registro de materias, registro de cronograma, registro de temas, muestreo de las calificaciones obtenida por los postulantes, y envió de notas a sus respectivos correos de los postulantes.

El sistema es una aplicación web cliente PHP, desarrollada con el framework Laravel organizado por módulos, cada uno con sus propias capas Modelo, Vista y Controlador.

Los módulos del sistema son:

- Módulo de administración de usuarios
- Módulo de requisitos
- Módulo de concurso de merito
- Módulo de concurso de opción
- Módulo de postulantes
- Módulo de materias
- Módulo de cronograma
- Módulo de asignatura
- Módulo de temas
- Módulo de Calificaciones
- Módulo de correo

4.3.4 ESPECIFICACIÓN DE REQUERIMIENTOS DEL SISTEMA

Con la especificación de requerimientos se identifica las necesidades funcionales, que permitirá dar a conocer los componentes que integran el sistema de Méritos y Oposiciones Académica SIGCOM, a fin de implementarlos más adelante.

4.3.5 REQUERIMIENTOS FUNCIONALES

Se utiliza el prefijo “Req” que representa el requerimiento, la finalidad es establecer un identificador y poder referenciar a los mismos. Los requerimientos establecidos en concordancia con el administrador de la empresa son:

Req1. El sistema permitirá autenticar a los usuarios previamente registrados.

Req2. El sistema permitirá la Gestión de postulantes (inserción, actualización, listado y eliminación).

Req3. El sistema permitirá la Gestión de la comisión evaluadora (inserción, actualización, listado y eliminación).

Req4. El sistema permitirá la Gestión de archivos (listar y descargar).

Req5. El sistema permitirá la Gestión de requisitos (Adjuntar, actualización, eliminación).

Req6. El sistema permitirá la Gestión de concursos de méritos (inserción, actualización, listado y eliminación).

Req8. El sistema permitirá la Gestión de concursos de oposición (inserción, actualización, listado y eliminación).

Req9. El sistema permitirá la Gestión de asignación de materiales (inserción, actualización, listado y eliminación).

Req10. El sistema permitirá la Gestión de asignación de cronogramas (inserción, actualización, listado y eliminación).

Req11. El sistema permitirá la Gestión de asignación de asignaturas (inserción, actualización, listado y eliminación).

Req12. El sistema permitirá la Gestión de asignación de temas (inserción, actualización, listado y eliminación).

Req13. El sistema permitirá la Gestión de actas de Calificación (inserción, actualización, listado y eliminación).

Req14. El sistema permitirá listar e imprimir los datos de los postulantes.

Req15. El sistema permitirá listar e imprimir las actas de calificaciones de los postulantes.

Requerimientos No funcionales

Los requerimientos no funcionales del sistema, definen el comportamiento interno del mismo, a continuación, se describen y son los siguientes:

- **Disponibilidad**

La disponibilidad del sistema será continua con un nivel de servicio para los postulantes de 24 horas al día, durante el proceso de concursos.

Escalabilidad

El sistema será capaz de adaptarse según el nivel de crecimiento de la empresa, con esto permitirá agregar nuevas funcionalidades.

- **Seguridad**

El sistema brindará patrones de seguridad tomando en cuenta la sensibilidad alta de información que maneja de acuerdo a las especificaciones funcionales dadas. Tendrá un formulario de autenticación con manejo de sesiones.

- **Integridad**

Los procesos de validación que se realizan en cada uno de los campos pueden ocasionar problemas de integridad de los datos.

- **Facilidad de uso**

El sistema ofrecerá interfaces amigables, intuitiva para el usuario con facilidad de brindar usarlo.

4.2 FASE II. DISEÑO DE SOFTWARE

En esta fase se creará la parte física del proyecto, así como la interfaz, los diseños deben ser simples y sencillos para que los postulantes y la comisión evaluadores puedan fácilmente interactuar con el sistema. El diagrama de base de datos, diccionario de datos, diagrama de componentes, diagrama de despliegue y las interfaces son los elementos más importantes del diseño.

4.2.1 DISEÑO DE LA BASE DE DATOS

A continuación, se visualiza el diseño de la base de datos se define las tablas, los campos y las relaciones (Figura 18), esto permite conocer la forma de almacenamiento de los datos provenientes del Sistema de gestión de concurso de mérito y opción en la Universidad Nacional de Chimborazo. El modelo fue analizado conjuntamente con el equipo de desarrollo en base a los requerimientos establecidos inicialmente.

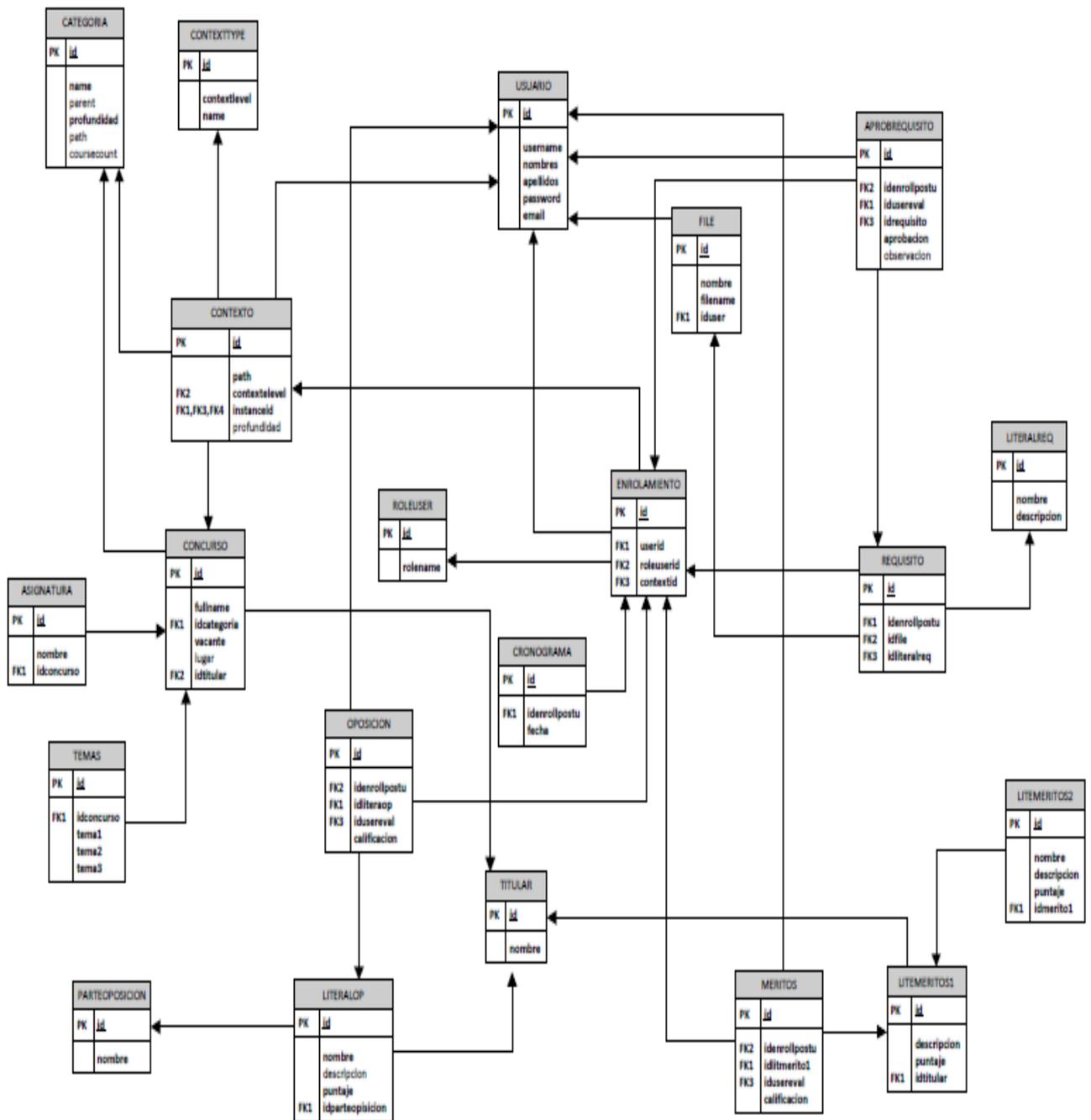


Figura 18: Diseño de la Base de Datos.

Elaborado por: Diego Chicaiza & Jhonatan Arcos

4.2.2 DICCIONARIO DE DATOS

Es la estructura lógica de almacenamiento de la base de dato que contiene una agrupación de metadatos. En esta se describe cada tabla de la base de datos empleada en el desarrollo del sistema, a continuación, se presentan los campos que se deben tomar en cuenta:

Atributo.- Campo de la base de una tabla de la base de datos

Descripción.- significado del campo para la base de datos.

Tipo.- Tipo de datos para la base de datos.

PK.- Si el campo en cuestión es clave primaria.

FK.- Si el campo en cuestión es clave foránea.

NULL.- Admite o no valores nulos.

Esta información permite determinar la forma de acceso desde la aplicación. Se estableció un estándar de la meta data, se usó “Id” para nombrar los identificadores de las tablas y que sean de tipo entero, para campos alfanuméricos el tipo de dato es varchar y para datos de fecha el tipo es datetime, ejemplo tabla Postulante (Tabla 25), las demás descripciones se pueden apreciar en el Anexo.

Tabla 25: Descripción tabla Usuarios Base de datos SIGCOM.

Atributo	Descripción	Tipo	Tipo Columna	NULL	Tabla Foránea
IdPostulante	Identifica al postulante	Int	PK	NOT NULL	
IdPerfil	Identifica el Perfil	Int	FK	NOT NULL	Perfil
Cedula	Numero cedula del postulante	varchar(11)			
Nombres	Nombres del postulante	varchar(50)			
Apellidos	Apellidos del postulante	varchar(50)			
UserName	Cuenta de usuario	varchar(25)			
Password	Clave para login	varchar(25)			
Email	Dirección electrónica del postulante	varchar(50)			
Estado	Estado del postulante	char(1)			

Elaborado por: Diego Chicaiza & Jhonatan Arcos

4.2.3 DISEÑO DE ARQUITECTURA

La arquitectura es la estructura base para el desarrollo e implementación de una aplicación, la misma se divide en dos: arquitectura lógica y arquitectura física.

LA ARQUITECTURA LÓGICA

Es muy importante establecer una arquitectura lógica de componentes ya que permitan tener una visión sobre la organización y las dependencias del sistema. Esto se refiere un modelo de componentes para el desarrollo de la aplicación.

Base de datos Centralizada. - Base de datos que sirve para almacenar todos los datos de cada una de las entidades que forman parte del diseño de la base de datos.

Componente Modelo. - Proporciona los métodos y clases de acceso a los datos de la base de datos centralizada.

Componente Controlador. - esta capa responde a eventos, generalmente peticiones de los usuarios (presionando un link, ejecutando un formulario, etc), e interactúa a su vez con el Modelo y la Vista (para pedirle información o para representar la información, respectivamente).

Componente Vista. - Esta capa es la encargada de presentar los datos en un formato adecuado para interactuar, usualmente cumpliendo con la "interfaz de usuario".

LA ARQUITECTURA FÍSICA O DE IMPLEMENTACIÓN

Se refiere a un modelo de Despliegue el cual permite mostrar cómo y dónde se desplegará el Sistema de Gestión Merito y oposición académica SIGCOM. Los servidores en los cuales se alojará la aplicación, las máquinas físicas y los ambientes donde se ejecutará el sistema

Servidor de base de datos PostgreSQL. - Para administrar la base de datos centralizada se tiene instalado un servidor de base de datos PostgreSQL Server.

Servidor de aplicaciones. - Para recuperar los datos del usuario, se tiene instalado un servidor de aplicaciones CentOS 7.

PC Usuario. - Tiene instalado un explorador web, con el cual se conecta al Sistema de Gestión Merito y oposición académica SIGCOM.

4.2.4 DIAGRAMA DE CASOS DE USOS

Proceso de planificación y envío de cronogramas

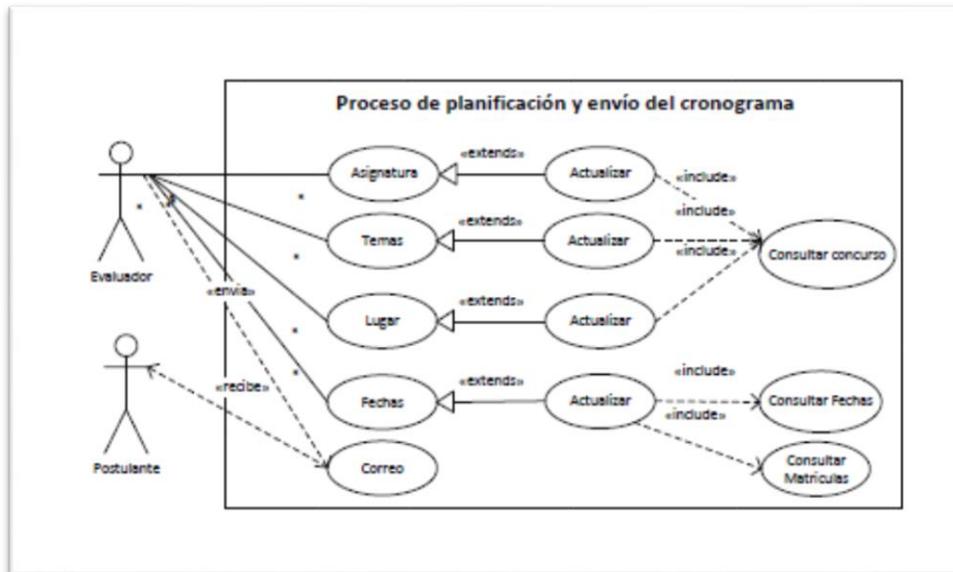


Figura 19: Proceso de planificación y envío de cronogramas

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Gestión de categorías

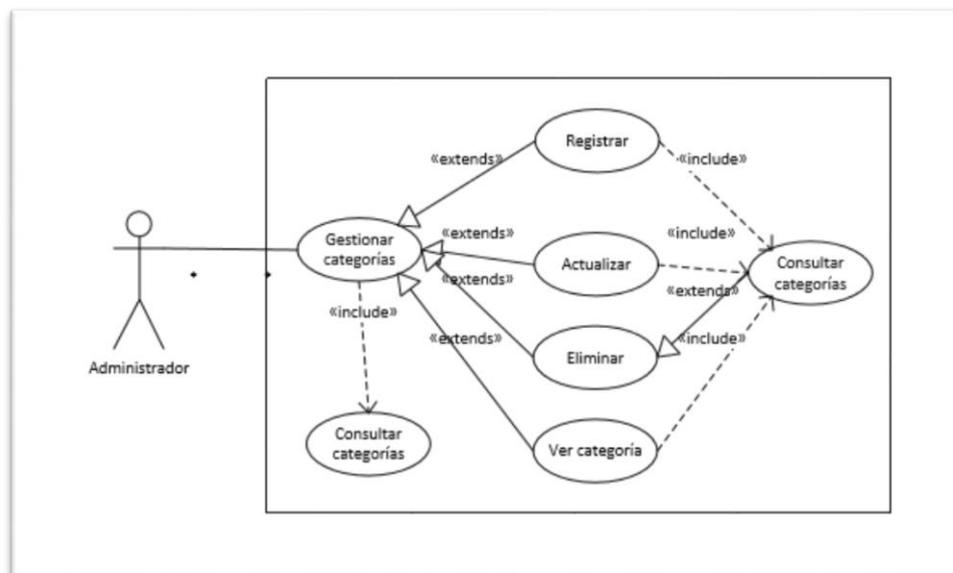


Figura 20: Gestión de categorías

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Gestión de concurso

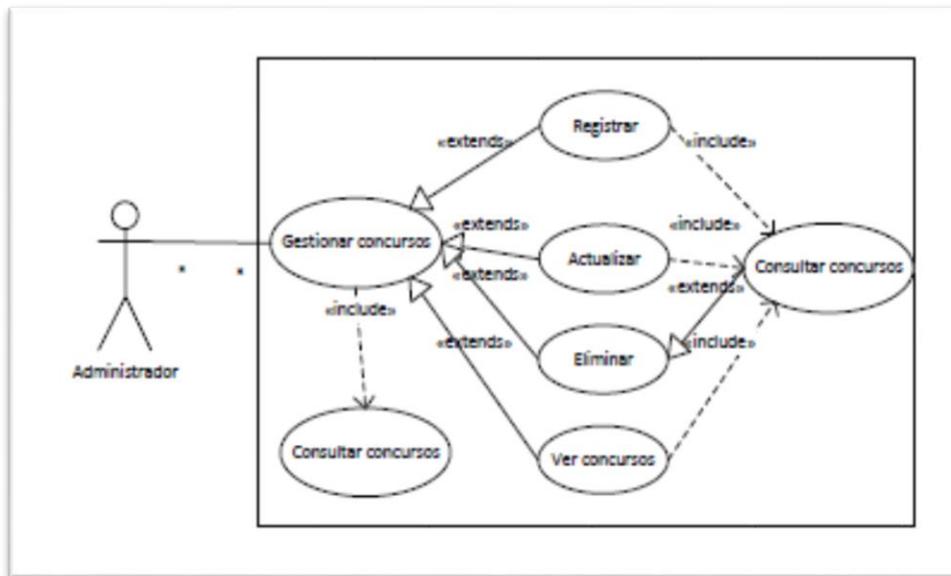


Figura 21: Gestión de concurso

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Gestión de matriculas

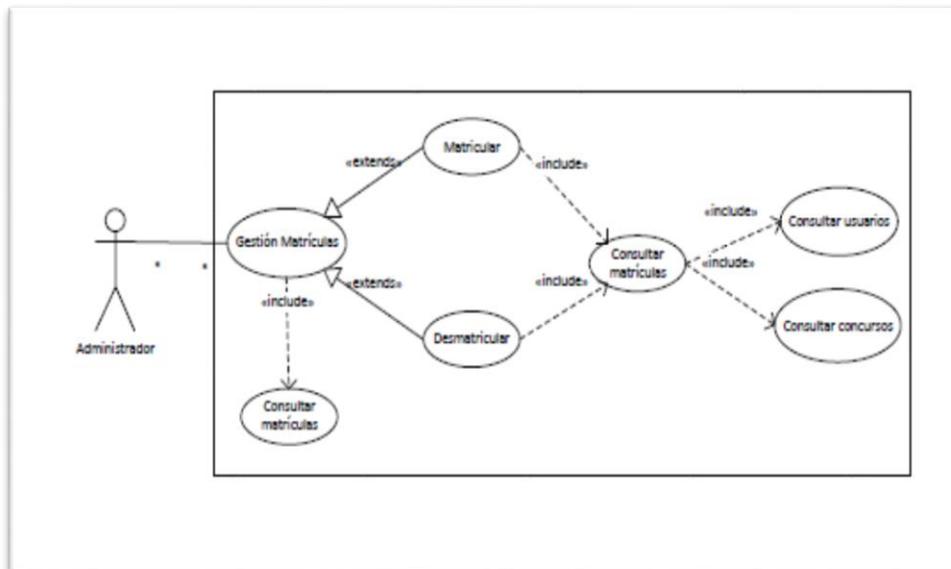


Figura 22: Gestión de matriculas

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Gestión de usuarios

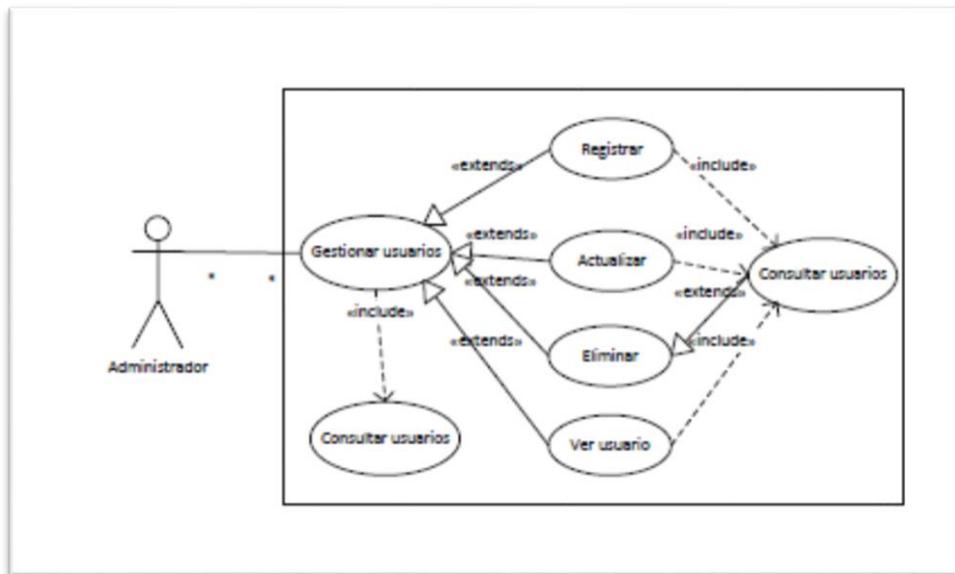


Figura 23: Gestión de usuarios

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Procedimiento para el reporte de resultados

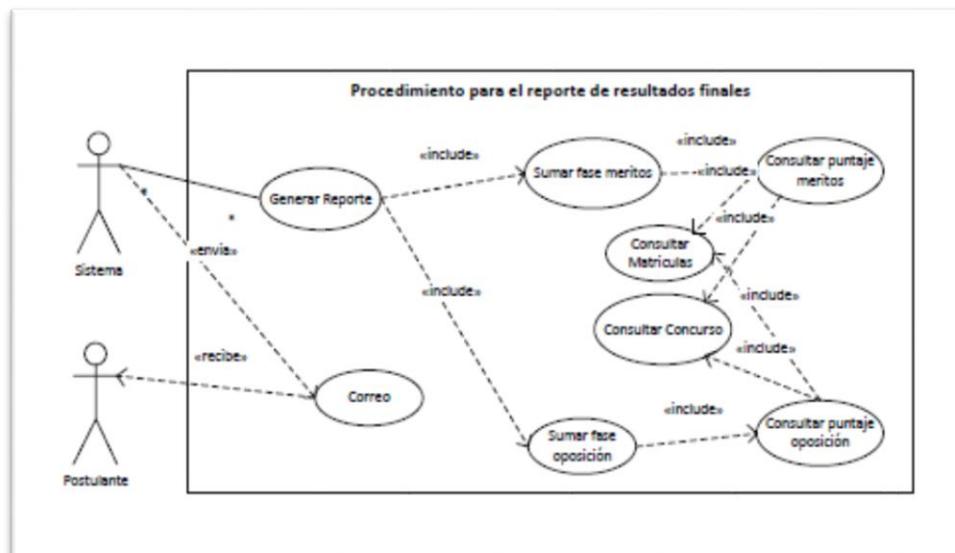


Figura 24: Procedimiento para el reporte de resultados

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Proceso de calificación de requisitos

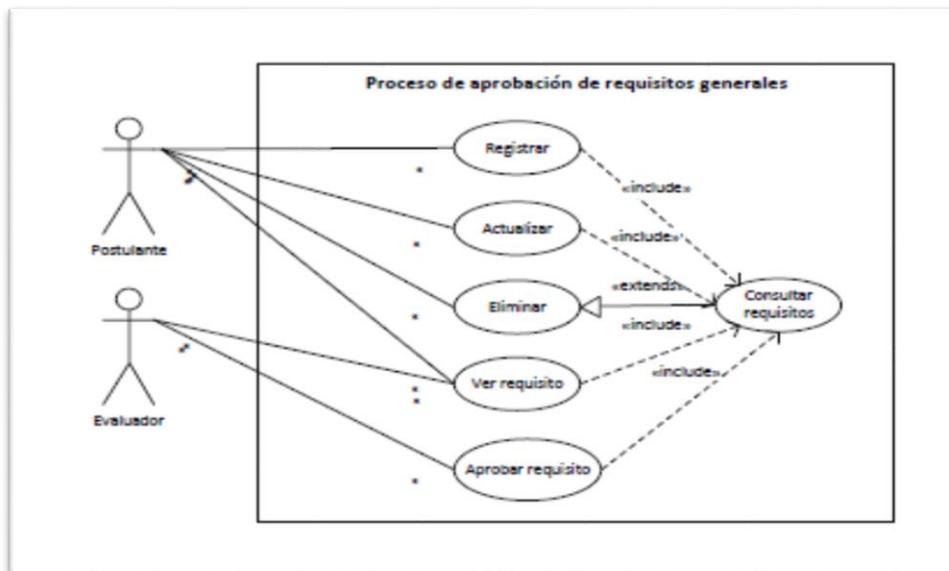


Figura 25: Proceso de calificación de requisitos

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Procesos de calificación de mérito y oposición

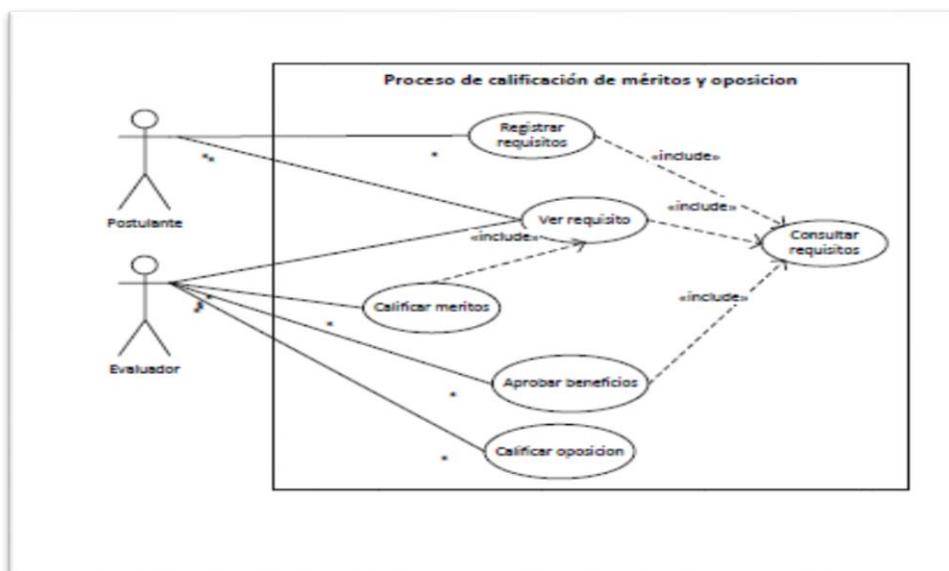


Figura 26: Procesos de calificación de mérito y oposición

Elaborado por: Diego Chicaiza & Jhonatan Arcos

4.3 PROGRAMACIÓN

Patrón de arquitectura Modelo Vista Controlador (MVC)

La programación se realizó utilizando el patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

Modelo

- Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
- Encapsula el estado de la aplicación.
- No sabe nada / independiente del Controlador y la Vista.

Vista

- Es la presentación del Modelo.
- Puede acceder al Modelo, pero nunca cambiar su estado.
- Puede ser notificada cuando hay un cambio de estado en el Modelo.

Controlador

- Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente

Se puede verificar los pasos a seguir de programación detallado en el (ANEXO D).

4.4 IMPLEMENTACIÓN

- Preparación del Servidor
- Instalación de Centos 7 Sistema Operativo de Servidor.
- Instalación y configuración de servidor glassfish.
- Subir el archivo war de la aplicación.
- Se puede verificar los pasos a seguir detallado en el Manual Técnico (ANEXO E).

CAPITULO V

METODOLOGÍA

5.1 TIPO DE ESTUDIO

Para el desarrollo del presente trabajo investigativo se tomó la consideración varios tipos de investigación, que se detallan a continuación:

5.1.1 SEGÚN EL OBJETO DE ESTUDIO:

- **Investigación aplicada:** La investigación aplicada busca la generación de conocimiento con aplicación directa al problema de la sociedad o el sector productivo. Realizando descubrimientos y avances de la investigación básica que enriquece con ellos, y se caracteriza por su interés. La investigación aplicada, busca el conocer para hacer, actuar, construir y modificar.

5.1.2 SEGÚN LA FUENTE DE INVESTIGACIÓN:

- **Investigación bibliográfica:** Conjunto de técnicas y estrategias que se emplean para localizar, identificar y acceder a aquellos documentos que contiene la información permitiendo para la investigación, estos medios son: libros, revistas, publicaciones, tesis, etc.

5.1.3 SEGÚN LAS VARIABLES:

- **Descriptiva Aplicada:** Mediante la descripción y análisis del objeto de estudio, determinar su forma de aplicación en un entorno real.

5.2 HIPÓTESIS

5.2.1 FORMULACIÓN DE LA HIPÓTESIS

El análisis comparativo de los frameworks Laravel y CodeIgniter permitirá determinar el framework más adecuado para la implementación del Sistema de Gestión de los procesos de concursos de méritos y oposiciones de la UNACH.

5.2.2 TIPO DE HIPÓTESIS

La Hipótesis de esta indagación es de tipo Causa – Efecto.

5.2.3 DETERMINACIÓN DE LAS VARIABLES

- **Variable Independiente:**

Análisis comparativo de los frameworks Laravel y CodeIgniter

- **Variable Dependiente**

Sistema de gestión de concursos de mérito y oposición en la “Universidad Nacional de Chimborazo”.

5.2.4 OPERACIONALIZACIÓN CONCEPTUAL

Tabla 26: Operacionalización Conceptual

VARIABLE	TIPO	CONCEPTO	DIMENSIÓN	INDICADORES
Análisis comparativo de los frameworks Laravel y CodeIgniter	Variable Independiente	Framework PHP Open-Source son herramienta de soporte que permite que otras aplicaciones sean implementadas y desarrolladas.	- Laravel - CodeIgniter	- Número de líneas de Código - Seguridad - Tiempo de respuesta
Sistema de gestión de concursos de méritos y oposición de la UNACH	Variable Dependiente	Capacidad de obtener resultados deseados en los recursos de información, mediante la óptima utilización de los recursos disponibles.	- Acoplamiento entre módulos - Independencia entre módulos - Eficiencia - Reportes	- Usabilidad. - Accesibilidad. - Simplicidad. - Funcionalidad. - Experiencia del Usuario

Elaborado por: Diego Chicaiza & Jhonatan Arcos

5.2.5 COMPROBACIÓN DE LA HIPÓTESIS

Formulación de Hipótesis Nula e Hipótesis de Investigación

- H_i = El análisis comparativo de los frameworks Laravel y CodeIgniter permitirá determinar el framework más adecuado para la implementación del Sistema de Gestión de los procesos de concursos de méritos y oposiciones de la UNACH.
- H_o = El análisis comparativo de los frameworks Laravel y CodeIgniter no permitirá determinar el framework más adecuado para la implementación del Sistema de Gestión de los procesos de concursos de méritos y oposiciones de la UNACH.

Para la comprobación de la Hipótesis se aplica T de Student para dos muestras independientes donde cada una de las muestras representa a los frameworks en estudio (Laravel y CodeIgniter). La fórmula a utilizar para el cálculo del valor Z (valor calculado de T de Student) se detalla a continuación

$$Z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Dónde:

\bar{x}_1 = media aritmética muestra 33,33

\bar{x}_2 = media aritmética muestra 23,99

σ_1^2 = varianza de la muestra 266,67

σ_2^2 = varianza de la muestra 174,67

n_1 = cantidad de elementos de la muestra 3

n_2 = cantidad de elementos de la muestra 3

Cálculos

Tabla 27: Cálculos

Parámetros	Frameworks	
	Laravel	CodeIgniter
Número de líneas de código	40	28,95
Tiempo de respuesta del sistema.	40	28,41
Manejo de seguridad.	20	12,5
Media aritmética	33,33	23,29
Varianza	266,67	174,67

Elaborado por: Diego Chicaiza & Jhonatan Arcos

$$Z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

$$Z = \frac{33,33 - 23,29}{\sqrt{\frac{266,67}{3} + \frac{174,67}{3}}}$$

$$Z = \frac{10,05}{12,12904}$$

$$Z = 0,8283$$

Una vez calculado el valor Z se procede a encontrar el valor T de Student de la Tabla con un nivel de significancia de 0,5 y con N grados de libertad.

Cálculo de los grados de libertad:

$$N = (n_1 - 1) + (n_2 - 1)$$

$$N = (3 - 1) + (3 - 1)$$

$$N = 4$$

El valor de la Tabla de T de Student con 4 Grados de libertad y un nivel de significancia de 0,5 es igual a $Z_c = 0,678$

Tabla distribución t de Student

x \ n	1	2	4	5	6	7
0,30	0,593	0,604	0,610	0,612	0,613	0,614
0,35	0,607	0,620	0,628	0,630	0,631	0,632
0,40	0,621	0,636	0,645	0,647	0,648	0,649
0,45	0,635	0,652	0,662	0,664	0,666	0,667
0,50	0,648	0,667	0,678	0,681	0,683	0,684
0,55	0,660	0,681	0,694	0,697	0,699	0,700
0,60	0,672	0,695	0,710	0,713	0,715	0,716

Figura 27: Distribución t de Student

Elaborado por: Diego Chicaiza & Jhonatan Arcos

Decisión

La **H₀** se rechaza y se acepta la **H₁** cuando **Z > Z_c**, en la presente investigación $0,88283 > 0,678$ se rechaza la **H₀**, es decir en la presente investigación se comprueba que “el análisis comparativo de los frameworks Laravel y CodeIgniter permitirá determinar el framework más adecuado para la implementación del Sistema de Gestión de los procesos de concursos de méritos y oposiciones de la UNACH”.

Esta afirmación es verdadera puesto que el análisis comparativo arrojó a Laravel como el framework óptimo para la implementación del sistema de Gestión de los procesos de concursos de méritos y oposición en la UNACH.

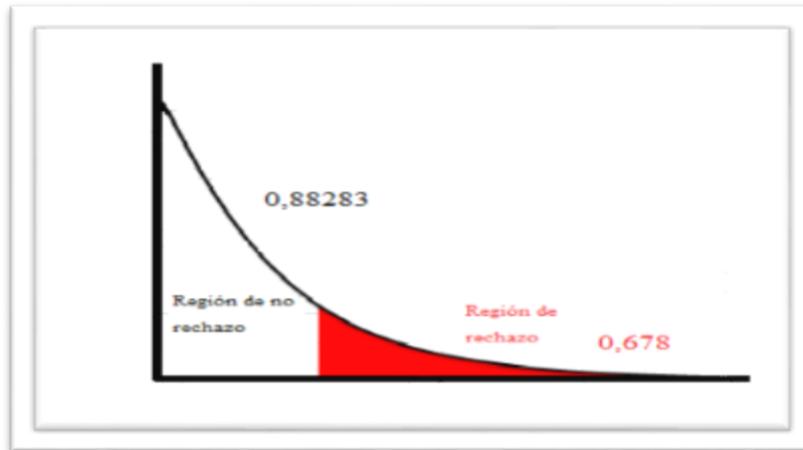


Figura 28: Grafica de T de Student

Elaborado por: Diego Chicaiza & Jhonatan Arcos

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- El análisis de los frameworks Laravel y CodeIgniter lleva a determinar que cada framework maneja archivos de configuración diferentes, de igual manera se pudo observar que cada una de las herramientas ofrece diferentes tipos de librerías para facilitar al programador, en su entorno de trabajo.
- El contar con una lista de parámetros medibles facilitó la comparación entre los framework de estudio, de acuerdo los parámetros establecidos: en cuanto al número de línea de código Laravel es superior con una diferencia del 11.05% a CodeIgniter, en tiempo de respuesta del sistema Laravel supera con un 11.59% a CodeIgniter y en manejo de seguridad Laravel brinda mayor seguridad con una diferencia del 7.6% con respecto a CodeIgniter.
- La investigación realizada, nos permitió determinar que el Framework Laravel es el adecuado en cuanto a la productividad que brinda para el desarrollo del sistema de gestión de concursos de mérito y oposición SIGCOM en un 31.14% a CodeIgniter.

6.2 RECOMENDACIONES

- Se recomienda que en un estudio comparativo se debe definir de manera correcta los parámetros de evaluación de acuerdo al objetivo de su estudio, para obtener resultados confiables.
- Se recomienda el uso del Framework Laravel para optimizar el desarrollo web usando PHP, como se demostró en este estudio brinda mayor productividad en relación al Framework CodeIgniter.
- Realizar posteriormente un estudio de productividad de los frameworks: Laravel y CodeIgniter puesto que la tecnología avanza continuamente y cada una de las herramientas estudiadas proveen nuevas versiones, las cuales pueden cubrir las deficiencias que poseen actualmente e incorporar nuevas funcionalidades.

7 BIBLIOGRAFÍA

- ALEGA. (20 de 01 de 2015). *DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA*. Obtenido de DICCIONARIO DE INFORMÁTICA Y TECNOLOGÍA:
<http://www.alegsa.com.ar/Dic/aplicacion%20web.php>
- Alvarez, M. A. (20 de 02 de 2016). *desarrolloweb*. Obtenido de desarrolloweb:
<http://www.desarrolloweb.com/articulos/introduccion-css3.html>
- Calero. (01 de 01 de 2004). *Métricas de Software*. Recuperado el 4 de 07 de 2016, de Métricas de Software: <http://ldc.usb.ve/~abianc/materias/ci4712/metricas.pdf>
- Caplan, P. (2009). *Entender PREMIS*. U.S.A: The Library of Congress, excepto en U.S.A.
- Card, D. N. (25 de 01 de 2006). *The Challenge of Productivity Measurement* . Recuperado el 03 de 07 de 2016, de The Challenge of Productivity Measurement :
http://www.netvalence.net/sites/default/files/resources/recom_read/TheChallengeofProductivityMeasurement.pdf
- CASTILLO, L. G. (7 de 01 de 2014). *ANÁLISIS COMPARATIVO DE METRO Y AXIS2 PARA EL*. Obtenido de ANÁLISIS COMPARATIVO DE METRO Y AXIS2 PARA EL:
<http://dspace.esPOCH.edu.ec/bitstream/123456789/3325/1/18T00548.pdf>
- Castrejón, E. (30 de 12 de 2012). *webadictos*. Obtenido de webadictos:
https://webadictos.com/2012/12/30/breve-historia-del-html/?wa_count=2
- CHAIN, G. (01 de 01 de 2014). *Dchain*. Obtenido de Dchain:
<http://web.dchain.com/2015/11/02/codigos-ventajas-y-desventajas-del-html/>
- Chicaiza, A. J. (4 de 07 de 2016). *Medicion de busqueda de los frameworks en tendencias de google* . Obtenido de Medicion de busqueda de los frameworks en tendencias de google :
<http://www.google.com/trends/explore#q=%2Fm%2F0jwy148%2C%20%2Fm%2F0c41gkv%2C%20cakephp%2C%20cake%20php%2C%20zend%20framework&date=today%2012-m&cmpt=q&tz=Etc%2FGMT%2B5>
- comunidadcodeignite. (05 de 07 de 2016). *comunidadcodeignite*. Obtenido de comunidadcodeignite: <https://comunidadcodeigniter.wordpress.com/>
- Crespo., Á. G. (04 de 02 de 2014). *Factores que afectan la productividad*. Obtenido de Factores que afectan la productividad:
https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEWj5nP708-LNAhWE0h4KHf0NDpYQFggcMAE&url=http%3A%2F%2Farchivo.uc3m.es%2Fbitstream%2Fhandle%2F10016%2F19367%2Ftesis_adrian_hernandez_lopez_2014.pdf%3Fsequence%3D1&u

- Dulio. (27 de 06 de 2016). *Porqué elegir Laravel en vez de Codeigniter*. Obtenido de Porqué elegir Laravel en vez de Codeigniter: <https://styde.net/porque-elegir-laravel-en-vez-de-codeigniter/>
- española, C. (05 de 07 de 2016). *Comunidad de framework Laravel*. Obtenido de Comunidad de framework Laravel: Comunidad de framework Laravel
- Gaitán, F. (20 de 02 de 2016). *modelo-vista-controlado*. Obtenido de modelo-vista-controlado: <http://fernando-gaitan.com.ar/laravel-parte-3-modelo-vista-controlador/>
- García, Y. (16 de Noviembre de 2014). *CONCEPTO DEFINICION*. Obtenido de CONCEPTO DEFINICION.DE: <http://conceptodefinicion.de/php/>
- Gartner. (09 de 01 de 2016). *El Software Libre en la empresa*. Obtenido de El Software Libre en la empresa: <http://informatica.blogs.uoc.edu/2011/06/28/el-software-libre-en-la-empresa/>
- González, E. (15 de 01 de 2015). *APR*. Recuperado el 20 de 02 de 2016, de APR: http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=435:i-que-es-y-para-que-sirve-html-el-lenguaje-mas-importante-para-crear-paginas-webs-html-tags-cu00704b&catid=69:tutorial-basico-programador-web-html-desde-cero&Itemid=192
- Gutiérrez, J. J. (07 de 05 de 2012). *investigacion_ficheros*. Obtenido de investigacion_ficheros: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
- Hazaël-Massieux, D. (20 de 02 de 2016). *Lenguaje de Programacion*. Obtenido de Desarrollo web: <https://www.w3.org/standards/webdesign/script>
- Larrea, S. (2 de 11 de 2015). *Tesis*. Obtenido de Tesis: <dspace.esPOCH.edu.ec/bitstream/123456789/4376/1/18T00601.docx>
- Lerdorf. (24 de 03 de 2016). *Características de CodeIgniter*. Recuperado el 04 de 06 de 2016, de Características de CodeIgniter: https://codeigniter.com/user_guide/overview/features.html
- Lerdorf, R. (24 de 03 de 2016). *Tutorial de CodeIgniter*. Recuperado el 4 de 07 de 2016, de Tutorial de CodeIgniter: http://www.codeigniter.com/user_guide/tutorial/index.html
- Lessin, J. (24 de 05 de 2014). *Ventajas y desventajas de usar Bootstrap*. Obtenido de Ventajas y desventajas de usar Bootstrap: <http://jorgelessin.com/ventajas-y-desventajas-de-usar-bootstrap/>
- Martin, P. R. (15 de 02 de 2016). *asptutor.com*. Obtenido de asptutor.com: <http://www.asptutor.com/zip/cbhtml.pdf>
- McDade, J. (22 de 02 de 2016). *laravel.com*. Obtenido de laravel.com: <https://laravel.com/docs/5.2/quickstart#introduction>

- Mehdi Achour & Jakub Vrana. (28 de 01 de 2016). *Manual de PHP*. Obtenido de Manual de PHP: <http://php.net/manual/es/index.php>
- Melgoza, J. (07 de 08 de 2013). *Criterios para Elegir el Framework Correcto*. Obtenido de Criterios para Elegir el Framework Correcto: <http://jonathanmelgoza.com/blog/10-criterios-para-elegir-el-framework-correcto/>
- Otwell, C. ©. (31 de 05 de 2016). *Laravel*. Obtenido de Rutas: <https://laravel.com/docs/5.2/routing>
- Patricio. (15 de 04 de 2013). *Desarrollando Webs Dinámicas*. Obtenido de Desarrollando Webs Dinámicas: <http://desarrollandowebdinamicas.blogspot.com/2013/04/laravel-un-framework-php-facil-de-usar.html>
- Phillips, M. (2013). *The NDSA Levels of Digital Preservation: An Explanation and* . Obtenido de http://www.digitalpreservation.gov/ndsa/working_groups/documents/NDSA_Levels_Archiving_2013.pdf
- Preservica. (July de 2013). *Digital Preservation Maturity Model*. Obtenido de Tessella: <http://preservica.com/download/852>
- rafaelma. (20 de 03 de 2013). *PostgreSQL*. Obtenido de PostgreSQL: http://www.postgresql.org.es/sobre_postgresql
- Rubio, . O. (2013). medición del software. En F. O. Rubio, *medición del software* (pág. 117). https://www.google.com.ec/?gfe_rd=cr&ei=Mdp-V7aMNeWw8weF9LHQDg#q=importancia+de+medir++el+software&tbm=bks.
- SÁNCHEZ & VERA . (2011). *Tesis*. Obtenido de Tesis: <http://dspace.esPOCH.edu.ec/bitstream/123456789/1461/1/18T00481.pdf>
- Sánchez, J. (01 de 01 de 2012). *Introducción a PHP*. Obtenido de Introducción a PHP: <http://www.jorgesanchez.net/web/iaw/iaw2.pdf>
- Solutions, K. (31 de 01 de 2014). *Criterios para Elegir el Framework Correcto*. Obtenido de Criterios para Elegir el Framework Correcto: <http://www.kipoint.com.mx/es/blog/10-criterios-para-elegir-el-framework-correcto#.V3wNzVThDIV>
- standard-ieee-std-1045-ieee-standard-software-productivity-metrics. (4 de 07 de 2016). *standard-ieee-std-1045-ieee*. Obtenido de standard-ieee-std-1045-ieee: <http://segoldmine.ppi-int.com/content/standard-ieee-std-1045-ieee-standard-software-productivity-metrics>
- Tertre, X. d. (6 de 03 de 2014). *Prestashop*. Obtenido de Prestashop: <https://www.prestashop.com/blog/es/que-es-bootstrap-la-historia-y-el-bombo-parte-1-de-2/>

The Consultative Committee for Space Data Systems. (January de 2002). *REFERENCE MODEL FOR AN OPEN ARCHIVAL INFORMATION SYSTEM (OAIS)*. Obtenido de <http://public.ccsds.org/publications/archive/650x0b1s.pdf>

UNESCO. (2003). *DIRECTRICES DE PRESERVACIÓN DIGITAL*. Obtenido de Preparado por la Biblioteca Nacional de Australia : <http://unesdoc.unesco.org/images/0013/001300/130071s.pdf>

Vargas, A. (3 de 05 de 2016). *Metrica de software*. Obtenido de Metrica de software: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo2.pdf

Vilcaguano Zumba, M. I. (12 de 10 de 2015). *Analisis coparativo de los frameworks*. Obtenido de Analisis coparativo de los frameworks: <http://dspace.espace.edu.ec/handle/123456789/3777>

Vinconti, D. M. (22 de 12 de 2015). *Ingenieria de Software Avanzado*. Obtenido de Ingenieria de Software Avanzado: <https://www.inf.utfsm.cl/~visconti/iswav/documentos/ProdyEstEsfuerzo.pdf>

Vschart. (05 de 07 de 2016). *Laravel vs. CodeIgniter*. Obtenido de Laravel vs. CodeIgniter: <http://vschart.com/compare/laravel/vs/codeigniter>

8 APENDICES O ANEXOS

8.1 GLOSARIO

CTE	Centro de Tecnologías Educativas
URLS	Uniform Resource Locator
MIT	Massachusetts Institute of Technology
ASP	Active Server Pages
API	Interfaz de Programación de Aplicaciones.
ACL	Access Control List
BD	Base de Datos
CGI	Common Gateway Interface (Interfaz de entrada común)
CSS	Cascading Style Sheets – Hoja de Estilo en Cascada
CRUD	Create, Read, Update Delete (crear, ver, editar, borrar)
UNACH	Universidad Nacional De Chimborazo
HTML	Lenguaje de Marcado de Hipertexto
HTTP	Protocolo de Transferencia de Hipertexto
IDE	Integrated Development Environment
JDBC	Java database Connectivity.
MVC	Modelo Vista Controlador
MVC+S	Model View Controller and Services
ODBC	Open Database Connectivity
OOP	Programación Orientada a Objetos
ORM	Object Relational Mapping
PDF	Portable Document Format
PHP	Lenguaje de Programación Interpretado
RAD	Desarrollo Rápido De Aplicaciones
RIA	Aplicaciones de Internet Enriquecidas

SQL Structured Query Language

XP Programación Extrema.

WEB Abreviatura de World Wide Web

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.

POO siglas programación orientada a objetos.

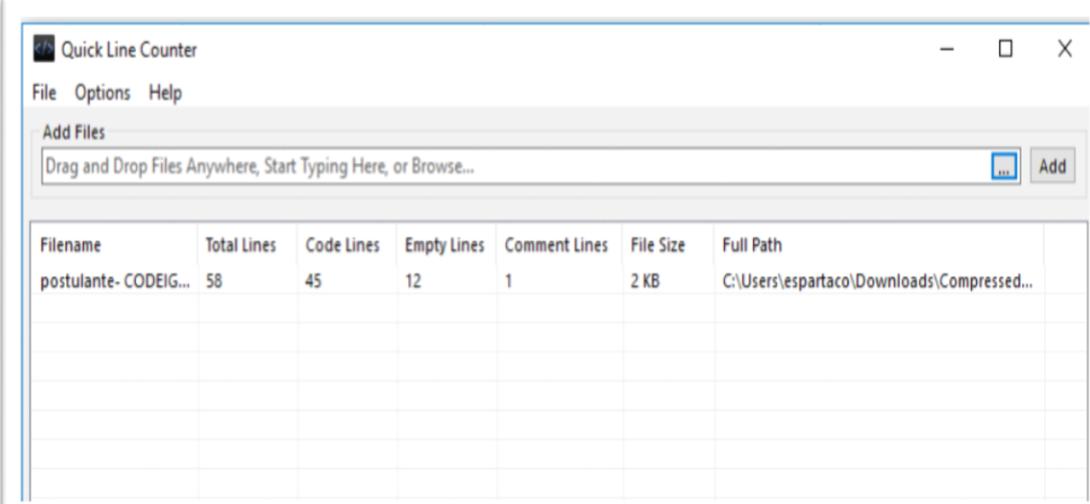
LDC líneas de código

UNACH Universidad Nacional de Chimborazo.

8.2 ANEXO

ANEXO A

Líneas de códigos que fueron Programadas, para la implementación de los sistemas con los Frameworks PHP: Laravel y CodeIgniter, siguiendo la arquitectura Modelo, Vista, Controlador.



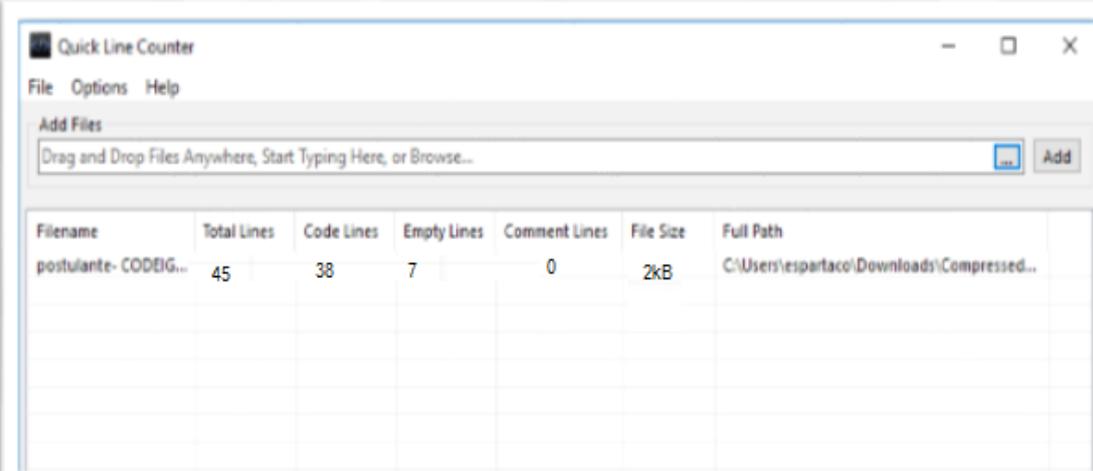
Quick Line Counter

File Options Help

Add Files

Drag and Drop Files Anywhere, Start Typing Here, or Browse...

Filename	Total Lines	Code Lines	Empty Lines	Comment Lines	File Size	Full Path
postulante- CODEIG...	58	45	12	1	2 KB	C:\Users\espartaco\Downloads\Compressed...



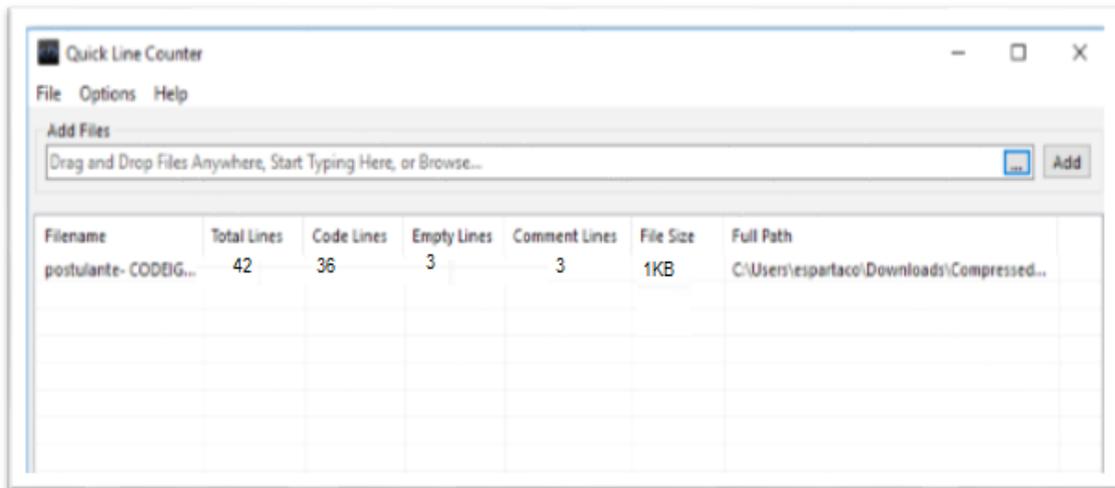
Quick Line Counter

File Options Help

Add Files

Drag and Drop Files Anywhere, Start Typing Here, or Browse...

Filename	Total Lines	Code Lines	Empty Lines	Comment Lines	File Size	Full Path
postulante- CODEIG...	45	38	7	0	2kB	C:\Users\espartaco\Downloads\Compressed...



```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
```

```
class Postulante extends CI_Controller {
```

```
    function __construct()
    {
        parent::__construct();

        $this->load->model('mfiles');
        $this->load->model('menrolls');
        $this->load->library('session');
        $this->load->helper('url');
    }
```

```
    function subirArchivo()
    {
        session_start();

        if(!isset($_SESSION['con_postu']))
        { //si no se hay sesion abierta
            redirect('/clogin/index','refresh');
        }
        else
```

```

{
    if ( isset($_POST['subir']) && isset($_GET['viewid']))
    {
        $iduser= $_SESSION['con_postu_id'];
        $idcon=$_GET['viewid'];

        // $nombre = $_FILES['archivoPDF']['name']."-
c".$idcon."u".$iduser;

        $trozos= explode(".", $_FILES['archivoPDF']['name']);
        $extension= $trozos[1];
        $nombre= $trozos[0];

        $tipo = $_FILES['archivoPDF']['type'];
        $nombreTemporal = $_FILES['archivoPDF']['tmp_name'];
        $ruta = "C:/xampp/dataConcursos/" . $nombre."-
c".$idcon."u".$iduser.".".$extension;;

        $existente=$this->mfiles->comprobarArchivoExistente($ruta);

        if ($nombre != "")
        {
            if (copy($nombreTemporal, $ruta))
            {
                if(empty($existente))
                {
                    $subirArchivo=$this->mfiles-
>AgregarPDF($nombre, $ruta, $iduser, $idcon);
                }

                redirect('/postulante/verConcurso?viewid='.$_GET['viewid'].'&action=subirArch
ivos', 'refresh');
            }
}

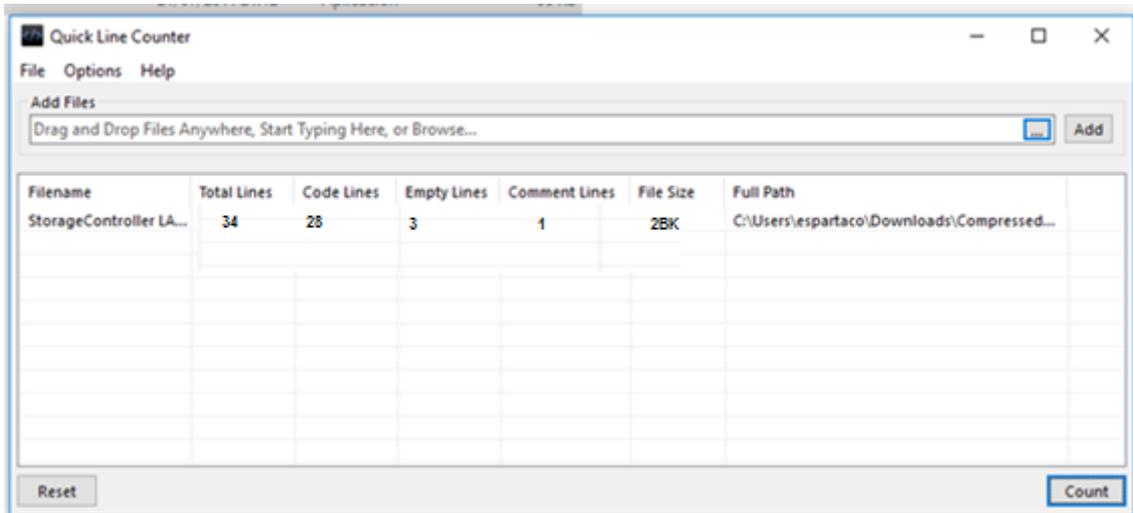
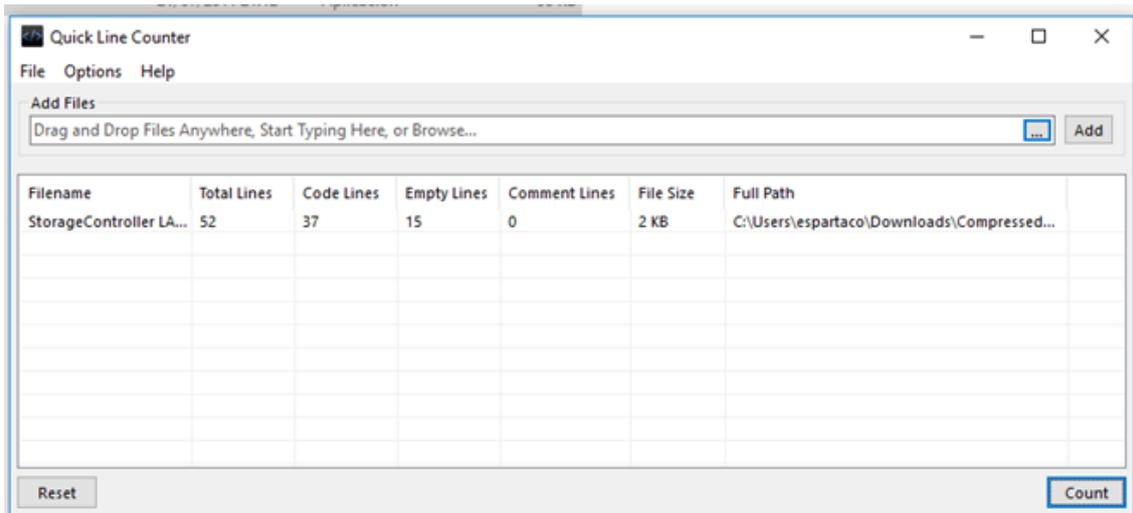
```

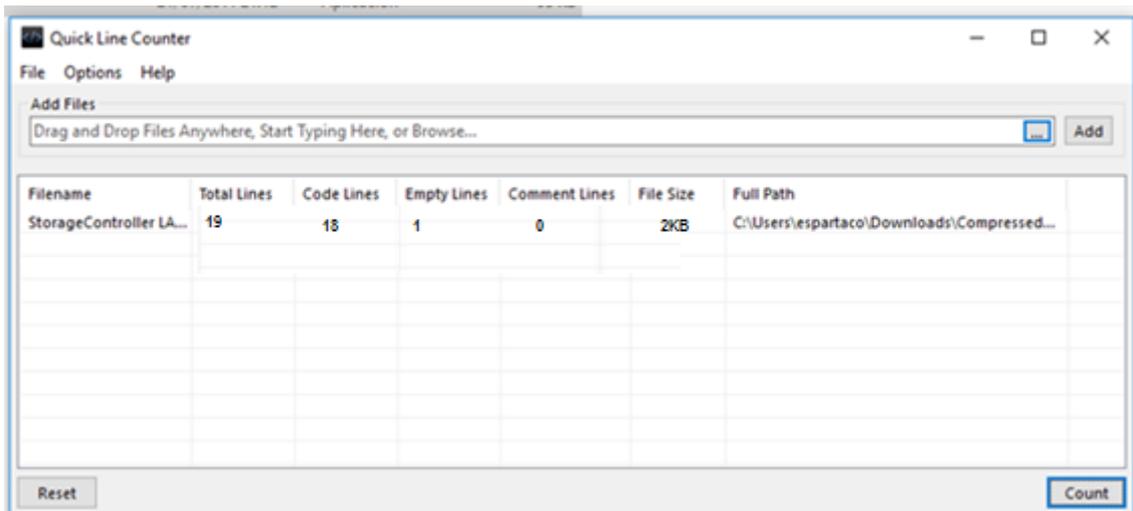
```

    }
    }
}
}
}

```

Laravel





```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use DB;
use App\Http\Requests\FileRequest;
use Laracasts\Flash\Flash;
use App\Http\Requests;

class StorageController extends Controller
{
    public function saveRequisito(FileRequest $request, $userid)
    {
        $file = $request->file('file');
        if( $file->getClientMimeType() == "application/pdf" )
        {
```

```

$nombre = $file->getClientOriginalName();

$partesNombreArchivo = explode('.', $nombre);
$nombreArchivo= $partesNombreArchivo[0];
$extension= $partesNombreArchivo[1];

$renombrado=$nombreArchivo.time();
$encriptado = hash('sha256', $renombrado);
$encriptado= $encriptado.".".$extension;

\Storage::disk('local')->put($encriptado, \File::get($file));

$archivo = new File();
$archivo-> nombre = $file->getClientOriginalName();
$archivo-> filename = $encriptado;
$archivo-> iduser = $userid;

$archivo->save();

Flash::success("Su archivo para el requisito ".$nombreLiteral.") ha sido subido
exitosamente!");
return redirect()->route('user.concursos.view',$requisito-> idenrollpostu);

}
else
{
Flash::warning("Solo se permiten subir archivos <b>pdf</b>");
return redirect()->route('user.concursos.view',$request-> idenrollpostu);
}
}
}

```

ANEXO B

Indicador 2: El tiempo de respuesta del sistema.

Prueba 4.1: Tiempo de respuesta en transacciones de inserción

Framework Laravel

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Etiqueta	# Muestras	Media	Min	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de By
http://localhost/...	55	39275	8895	46272	5719.11	0.00%	1.2/sec	10.01	8757
http://localhost/...	55	2757	1133	4418	841.25	0.00%	2.6/sec	21.17	8411
http://localhost/...	110	1742	131	4812	900.64	0.00%	4.9/sec	20.07	4221
http://localhost/...	55	962	195	2976	610.84	0.00%	2.4/sec	11.89	5026
http://localhost/...	55	839	110	2442	596.71	0.00%	2.4/sec	21.95	9238
http://localhost/...	55	1723	122	4875	1170.63	0.00%	2.5/sec	23.62	9784
Total	385	5.16	110	6272	13386.35	0.00%	7.1/sec	49.34	7094

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Etiqueta	# Muestras	Media	Min	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de By
http://localhost/...	55	39275	28895	46272	5719.11	0.00%	1.2/sec	10.01	8757
http://localhost/...	55	2757	1133	4418	841.25	0.00%	2.6/sec	21.17	8411
http://localhost/...	110	1742	131	4812	900.64	0.00%	4.9/sec	20.07	4221
http://localhost/...	55	962	195	2976	610.84	0.00%	2.4/sec	11.89	5026
http://localhost/...	55	839	110	2442	596.71	0.00%	2.4/sec	21.95	9238
http://localhost/...	55	1723	122	4875	1170.63	0.00%	2.5/sec	23.62	9784
Total	385	7.006	110	46272	13386.35	0.00%	7.1/sec	49.34	7094

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos Configurar

Etiqueta	# Muestras	Media	Min	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de By
http://localhost/...	55	5152	4732	8199	571.68	0.00%	6.5/sec	182.07	28463.5
http://localhost/...	55	5009	4764	5336	109.12	0.00%	6.3/sec	187.22	30297.2
http://localhost/...	55	6266	4280	7009	703.72	0.00%	6.3/sec	957.29	142370.8
http://localhost/...	55	4960	2904	5647	613.83	0.00%	6.1/sec	904.94	164435.3
http://localhost/...	110	4765	2399	7263	780.99	0.00%	6.6/sec	99.47	15492.0
http://localhost/...	55	4742	3390	5590	549.26	0.00%	5.2/sec	147.80	29322.8
Total	385	4.993	2.022	8.624	799.50	0.00%	0.6213/sec	609.20	60839.1

¿Incluir el nombre del grupo en la etiqueta? Guardar la tabla de datos Guardar la cabecera de la tabla

Archivo Editar Search Lanzar Opciones Ayuda

0 0 / 55

Test Plan

- Estado
- HTTP Cookie Manager
- User Defined Variables
- HTTP Header Manager
- Step 1
 - http://localhost/
 - http://localhost/admin/login/index
 - http://localhost/admin/login/auten
 - http://localhost/admin/login/logou
- Informe Agregado
- Reporte resumen
- Banco de Trabajo

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de Byt.
http://localho...	55	5152	732	8199	571.68	0.00%	6.5/sec	182.07	28463.5
http://localho...	55	5009	4764	5336	109.12	0.00%	6.3/sec	187.22	30297.2
http://localho...	55	6266	4280	009	703.72	0.00%	6.9/sec	957.29	142370.8
http://localho...	55	4960	904	5647	613.83	0.00%	6.1/sec	984.94	164435.3
http://localho...	110	4765	399	263	780.99	0.00%	6.6/sec	99.47	15492.0
http://localho...	55	4742	3390	590	549.26	0.00%	5.2/sec	147.80	29322.8
Total	385	6.507	0.068	53.971	799.50	0.00%	14.2672/sec	609.20	60839.1

¿Incluir el nombre del grupo en la etiqueta? Guardar la cabecera de la tabla

Archivo Editar Search Lanzar Opciones Ayuda

0 0 / 55

Test Plan

- Estado
- HTTP Cookie Manager
- User Defined Variables
- HTTP Header Manager
- Step 1
 - http://localhost/
 - http://localhost/admin/login/index
 - http://localhost/admin/login/auten
 - http://localhost/admin/login/logou
- Informe Agregado
- Reporte resumen
- Banco de Trabajo

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de Byt.
http://localho...	55	5152	732	8199	571.68	0.00%	6.5/sec	182.07	28463.5
http://localho...	55	5009	4764	5336	109.12	0.00%	6.3/sec	187.22	30297.2
http://localho...	55	6266	4280	009	703.72	0.00%	6.9/sec	957.29	142370.8
http://localho...	55	4960	904	5647	613.83	0.00%	6.1/sec	984.94	164435.3
http://localho...	110	4765	399	263	780.99	0.00%	6.6/sec	99.47	15492.0
http://localho...	55	4742	3390	590	549.26	0.00%	5.2/sec	147.80	29322.8
Total	385	5.094	2.399	8.199	799.50	0.00%	0.7995/sec	609.20	60839.1

Archivo Editar Search Lanzar Opciones Ayuda

0 0 / 55

Test Plan

- Estado
- HTTP Cookie Manager
- User Defined Variables
- HTTP Header Manager
- Step 1
 - http://localhost/
 - http://localhost/admin/login/index
 - http://localhost/admin/login/auten
 - http://localhost/admin/login/logou
- Informe Agregado
- Reporte resumen
- Banco de Trabajo

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de Byt.
http://localho...	55	5152	732	8199	571.68	0.00%	6.5/sec	182.07	28463.5
http://localho...	55	5009	4764	5336	109.12	0.00%	6.3/sec	187.22	30297.2
http://localho...	55	6266	4280	009	703.72	0.00%	6.9/sec	957.29	142370.8
http://localho...	55	4960	904	5647	613.83	0.00%	6.1/sec	984.94	164435.3
http://localho...	110	4765	399	263	780.99	0.00%	6.6/sec	99.47	15492.0
http://localho...	55	4742	3390	590	549.26	0.00%	5.2/sec	147.80	29322.8
Total	385	7.416	0.127	35.829	799.50	0.00%	11.4347/sec	609.20	60839.1

Archivo Editar Search Lanzar Opciones Ayuda

0 0 / 55

Test Plan

- Estado
- HTTP Cookie Manager
- User Defined Variables
- HTTP Header Manager
- Step 1
 - http://localhost/
 - http://localhost/admin/login/index
 - http://localhost/admin/login/auten
 - http://localhost/admin/login/logou
- Informe Agregado
- Reporte resumen
- Banco de Trabajo

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar... Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están.	% Error	Rendimiento	Kb/sec	Media de Byt.
http://localho...	55	5152	732	8199	571.68	0.00%	6.5/sec	182.07	28463.5
http://localho...	55	5009	4764	5336	109.12	0.00%	6.3/sec	187.22	30297.2
http://localho...	55	6266	4280	009	703.72	0.00%	6.9/sec	957.29	142370.8
http://localho...	55	4960	904	5647	613.83	0.00%	6.1/sec	984.94	164435.3
http://localho...	110	4765	399	263	780.99	0.00%	6.6/sec	99.47	15492.0
http://localho...	55	4742	3390	590	549.26	0.00%	5.2/sec	147.80	29322.8
Total	385	4.806	1.966	8.343	799.50	0.00%	1.5237/sec	609.20	60839.1

Archivo Editar Search Lanzar Opciones Ayuda

0 / 55

Test Plan

- Estado
- HTTP Cookie Manager
- User Defined Variables
- HTTP Header Manager
- Step 1
 - http://localhost/
 - http://localhost/admin/login/index
 - http://localhost/admin/login/auten
 - http://localhost/admin/login/logou
- Informe Agregado
- Reporte resumen
- Banco de Trabajo

Reporte resumen

Nombre: Reporte resumen

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Navegar...

Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Están...	% Error	Rendimiento	Kb/sec	Media de Byt...
http://localho...	55	5152	732	8199	571.68	0.00%	6.5/sec	182.07	28463.5
http://localho...	55	5009	4764	5336	109.12	0.00%	6.3/sec	187.22	30297.2
http://localho...	55	6266	4280	009	703.72	0.00%	6.9/sec	957.29	142370.8
http://localho...	55	4960	904	5647	613.83	0.00%	6.1/sec	984.94	164435.3
http://localho...	110	4765	399	263	780.99	0.00%	6.6/sec	99.47	15492.0
http://localho...	55	4742	3390	590	549.26	0.00%	5.2/sec	147.80	29322.8
Total	385	7.109	0.085	40.267	799.50	0.00%	12.4804/sec	609.20	60839.1

¿Incluir el nombre del grupo en la etiqueta? Guardar la cabecera de la tabla

ANEXO C

Indicador 3: El manejo de seguridad.

Prueba 3.1: Manejo de autenticación

El sistema de autenticación en laravel es realmente sencillo, laravel permite proporcionar el método Auth con el que simplemente pasando el usuario y el password ya nos hace todo el trabajo, genial. Eso sí, como todo buen framework, para poder hacer uso de Auth, deberá hacer algunas cosas antes entre ellas modificar el archivo auth.php alojado en config, yaqué devuelve un array, simplemente debe pasar el driver que quiera utilizar, el fluent o el eloquent, se puede utilizar cualquiera de los dos drivers. Y además para las bases de datos utiliza bcrypt para tener mayor seguridad con los datos.

Prueba 3.2: Uso de ACL

En Laravel permite configurar las restricciones de acceso de las aplicaciones de una forma mucho más conveniente, porque se puede definir todas las reglas necesarias y posteriormente utilizarlas desde los controladores, modelos, middleware e inclusive las plantillas de blade. Este componente es muy fácil de usar, pero cuenta con una serie de características bastante amplias, y se pueden adaptar al tamaño de cualquier proyecto.

Prueba 5.3: Uso de sesiones

El uso de secciones de las aplicaciones de Laravel se realiza bajo HTTP, las sesiones proveen un modo de almacenar información acerca del usuario a través de las peticiones. Laravel viene con gran variedad de soluciones de sesión disponibles para su uso a través de una clara y unificada API. Soporta soluciones populares de fábrica tales como Memcached, Redis, y Bases de Datos que son incluidas de fábrica.

La configuración de las sesiones se almacena en app/config/session.php. Yaqué por defecto, Laravel está configurado para usar el gestor de sesión file, que funciona bien para la mayoría de las aplicaciones. Antes de usar una sesión Redis con Laravel, necesitará instalar el paquete predis/predis (~ 1.0) a través de Composer. Si necesita que todos los datos almacenados de la sesión sean encriptados, establezca la opción de configuración encrypt en true.

ANEXO D

Línea de código del sistema.

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Administrator extends CI_Controller {

    function __construct()
    {
        parent::__construct();

        $this->load->model('usuarios');
        $this->load->model('mcategorias');
        $this->load->model('mconcursos');
        $this->load->model('mcontext');
        $this->load->model('menrolls');
            $this->load->library('session');
            $this->load->helper('url');

    }

    function index()
    {
        session_start();

        if(!isset($_SESSION['con_admin']))
            { //si no se hay sesion abierta
                redirect('/clogin/index','refresh');
            }
        else
            {
                $data['titulo']= 'Página Principal';
                $this->load->view('administracion/plantilla/Header',$data);
                $this->load->view('administracion/index',$_SESSION['con_admin']);
                $this->load->view('administracion/plantilla/Footer');
            }
    }

    ///////////////////////////////////////////////////////////////////
    //CONTROLADORES PARA ADMINISTRACION DE USUARIOS//
    ///////////////////////////////////////////////////////////////////

    function VerCedulasUsuariosAJAX()
    {
```

```

session_start();

if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
      redirect('/clogin/index','refresh');
    }

else
    {
        $cedulas=$this->usuarios->VerCedulasConcursos();
        foreach ($cedulas as $cedula) {
            if ( ($cedula-> username == trim($_GET['ci'])) && ($cedula->
id != $_GET['id']) ){
                echo "<h4 class='bg-danger'> Este nombre de usuario
o cedula ya existe. Por favor, seleccione otro</h4>";
            }
        }
    }
}

function NuevoUsuario()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
        { //si no se hay sesion abierta
          redirect('/clogin/index','refresh');
        }

    $data['titulo']= 'Ingresar Nuevo';
    $this->load->view('administracion/plantilla/Header',$data);
    $this->load->view('administracion/usuario/formnuevo',$data);

    if(isset($_POST['submit']))
        {
            $username= $_POST['cedula'];
            $nombres=$_POST['nombres'];
            $apellidos=$_POST['apellidos'];
            $password=$_POST['password'];
            $email=$_POST['email'];
            $role=$_POST['role'];

```

```

        $validacion=$this->usuarios->UsuarioRegistrado($username);
//Se comprueba que el Usuario no este registrado

        $existente=$validacion['0'] -> total;

        if ($existente == '1' )
        {

                redirect('/administrator/NuevoUsuario?doble=si&username='.$username.'&nombres='.$nombres.'&apellidos='.$apellidos.'&email='.$email.','refresh');

        }

        else
        {

                $nuevo=$this->usuarios->InsertarUsuario($username, $nombres, $apellidos, $password, $email, $role);

                // TODOS LOS DATOS DEL NUEVO USUARIO EN LA
                TABLA FUERON INGRESADOS POR TRIGGER

                redirect('/administrator/AdministrarUsuario','refresh');

        }

        $this->load->view('administracion/plantilla/Footer');
}

function AdministrarUsuario()
{
    session_start();
    $data['titulo']= 'Administración: Usuarios: Cuentas: Examinar lista de usuarios';
    $this->load->view('administracion/plantilla/Header',$data);
    if(isset($_POST['submit1']))
    {
        $ci= $_POST['ci'];

        $this->load->model('usuarios');
        $data['usuarios_obtenidos']=$this->usuarios->ObtenerUsuarioCI($ci); // comprobamos
        que el usuario exista en la base de datos y la password ingresada sea correcta
        $this->load->view('administracion/usuario/buscar',$data);
    }

    else

```

```

        {
            if(isset($_POST['submit2']))
            {
                $apellido= $_POST['apellido'];
                $this->load->model('usuarios');
                $data['usuarios_obtenidos']=$this->usuarios-
>ObtenerUsuarioApellido($apellido); // comprobamos que el usuario exista en la base de datos y la
password ingresada sea correcta
                $this->load->view('administracion/usuario/buscar',$data);
            }
            else
            {
                $this->load-
>view('administracion/usuario/buscar',$_SESSION['con_admin']);
            }
        }
        $data['lista_usuarios']=$this->usuarios->ObtenerUsuarios(); // comprobamos
que el usuario exista en la base de datos y la password ingresada sea correcta
        $this->load->view('administracion/usuario/usuarios',$data);
        $this->load->view('administracion/plantilla/Footer');
    }

```

```

function BorrarUsuario()
{
    session_start();

    if(!isset($_SESSION['con_admin']) )
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }

    else
    {
        $idUser= $_GET['iduser'];

        $borrado=$this->usuarios->EliminarUsuario($idUser); //Se
comprueba que el Usuario no este registrado
        redirect('/administrator/AdministrarUsuario','refresh');
    }
}

```

// ACTUALIZAR - PARTE 1: VerDatosUsuario - PARTE 2: ModificarUsuario

```
function VerDatosUsuario() //
{
    session_start();

    if(!isset($_SESSION['con_admin']) )
        { //si no se hay sesion abierta
            redirect('/clogin/index','refresh');
        }
    else
        {
            $idUserio= $_GET['id'];

            $this->load->model('usuarios');
            $data['usuarios_obtenidos']=$this->usuarios-
>ObtenerUsuariold($idUserio); // comprobamos que el usuario exista en la base de datos y la
password ingresada sea correcta

            $data['titulo']= 'Actualizar Usuario';
            $this->load-
>view('administracion/plantilla/Header',$data);
            $this->load-
>view('administracion/usuario/actualizar',$data);
            $this->load->view('administracion/plantilla/Footer');
        }
}

function ModificarUsuario() // FUNCION PARA ANTES DE ACTUALIZAR
{
    session_start();

    if(!isset($_SESSION['con_admin']) )
        { //si no se hay sesion abierta
            redirect('/clogin/index','refresh');
        }
    else
        {
            $idUserio= $_POST['id'];
            $username= $_POST['cedula'];
            $nombres=$_POST['nombres'];
            $apellidos=$_POST['apellidos'];
            $email=$_POST['email'];
            $password=$_POST['password'];
            $role=$_POST['role'];

```

```

                $actualizacion=$this->usuarios-
>ActualizarUsuario($idUsuario, $username, $nombres, $apellidos, $email, $password, $role);

                redirect('/administrator/AdministrarUsuario','refresh');

            }
        }
    }
}

```

```

////////////////////////////////////
//////CONTROADORES PARA ADMINISTRACION DE CATEGORIAS //////////////////////////////////
////////////////////////////////////

```

```

function Management()
{
    session_start();
    $data['titulo']= 'Gesti&oacute;n de concursos y categor&iacute;as';
    $this->load->view('administracion/plantilla/Header',$data);
    $data['escuelas_listado']=$this->mcategorias->ObtenerSoloEscuelas(); // Para
ventana modal de ACTUALIZAR concurso

    $data['categorias_listado']=$this->mcategorias->ObtenerCategoriasListado();
//Listado de todas las categorias

    $data['lista_categorias1']=$this->mcategorias->ObtenerCategoriasAnio();

    if(isset($_GET['idanio']))
    {
        $idanio=$_GET['idanio'];
        $data['lista_categorias2']=$this->mcategorias-
>ObtenerCategoriasFacultad($idanio);
        $data['hijos_concursos']=$this->mcategorias-
>ContarCategoriasHijas($idanio);

        if(isset($_GET['idfacultad']))
        {
            $idfacultad=$_GET['idfacultad'];
            $data['lista_categorias3']=$this->mcategorias-
>ObtenerCategoriasEscuela($idfacultad);
            $data['hijos_concursos']=$this->mcategorias-
>ContarCategoriasHijas($idfacultad);

            if(isset($_GET['idescuela']))

```

```

        {
            $idescuela=$_GET['idescuela'];
            $data['lista_concursos']=$this->
            >mconcursos->ObtenerConcursosEscuela($idescuela);
            $data['hijos_concursos']=$this->
            >mcategorias->ContarCategoriasHijas($idescuela);
        }
    }
    $this->load->view('administracion/categoria/management',$data);
    $this->load->view('administracion/plantilla/Footer');
}

function NuevaCategoria()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }

    $data['titulo']= 'Agregar nueva categoria';
    $this->load->view('administracion/plantilla/Header',$data);
    $data['categorias_listado']=$this->mcategorias->ObtenerCategoriasListado();
    $this->load->view('administracion/categoria/formnuevo',$data);

    if(isset($_POST['submit']))
    {
        $name= $_POST['nombreCategoria'];
        $padreANDprofundidad= $_POST['parentdepth'];

        $partes = explode("/", $padreANDprofundidad);

        $parent= $partes[0]; // padre
        $profundidad= $partes[1]; // profundidad

        $crearCategoria=$this->mcategorias->InsertarCategoria($name,
        $parent, $profundidad); //Creo la nueva facultad en la tabla mdl_course_categories

        $CatCreada=$this->mcategorias->
        >BuscarUltimaCategoriaCreada($name); //Busca el id de la ultima facultad creada
        $idCatCreada= $CatCreada -> id ;
    }
}

```

```

        if($profundidad == '0')
        {
            $path='/';
        }
        else
        {
            $catPadre=$this->mcategorias-
>BuscarPathCategoriaPadre($idCatCreada); //Busca el id de la ultima facultad creada
            $path= $catPadre -> pathpadre."/";
        }

        $nuevoPathCategoria=$this->mcategorias-
>ActualizarPathCategoria($idCatCreada, $path);//Actualizo el path de la facultad creada

        $nuevoContextCategoria=$this->mcontext-
>InsertarCategoriaContext($idCatCreada, $profundidad); //Insertar el id de la facultad en la tabla
mdl_context

        $datosPadreContext=$this->mcontext-
>ConsultarPathCategoriaPadre($parent);

        if(isset($datosPadreContext -> path))
        {
            $pathContextPadre=$datosPadreContext -> path;
        }
        else
        {
            $pathContextPadre='/1';
        }

        $pathCPadre= $pathContextPadre.'/';

        $nuevoContextPathCategoria=$this->mcontext-
>ActualizarContextPathCategoria($idCatCreada,$pathCPadre); //Insertar el id del periodo en la tabla
mdl_context

//http://localhost/concursos/administrator/Management?idanio=3&idfacultad=4&idescuela
=6

        $PathCatCreada=$this->mcategorias->BuscarPathById($idCatCreada);
//Busca el id de la ultima facultad creada
        $pathNuevaCategoria= $PathCatCreada -> path ;

```

```

$partes = explode("/", $pathNuevaCategoria);

$npartes= count($partes);

if ($npartes == 2)
{

redirect('/administrator/Management?idanio='.$partes["1"],'refresh');
}
else
{
    if ($npartes == 3)
    {

redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"],'refr
esh');

    }
    else
    {

redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"].'&id
escuela='.$partes["3"],'refresh');

    }
}

$this->load->view('administracion/plantilla/Footer');
}

function BorrarCategoria()
{
    session_start();

    if(!isset($_SESSION['con_admin']) )
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }

    else
    {
        if(isset($_GET['idborrar']))
        {

```

```

        $idCategoria= $_GET['idborrar'];

        $PathIdborrar=$this->mcategorias-
>BuscarPathById($idCategoria); //Busca el id de la ultima facultad creada
        $pathCategoriaBorrada= $PathIdborrar -> path ;

        $partes = explode("/", $pathCategoriaBorrada);

        $npartes= count($partes);
        echo $npartes;

        $borrado=$this->mcategorias-
>EliminarCategoria($idCategoria); //Se comprueba que el Usuario no este registrado

        if (($npartes == 2) && ($borrado == false))
        {

                redirect('/administrator/Management?borrado=si','refresh');
                }
                else
                {
                        if (($npartes == 3) && ($borrado == false))
                        {

                                redirect('/administrator/Management?borrado=si&idanio='.$partes["1"],'refresh');
                                }
                                else
                                {

                                        redirect('/administrator/Management?borrado=si&idanio='.$partes["1"].'&idfacultad='.$partes["2"],'refresh');
                                        }
                                }
                        }
                }
        }

function ModificarCategoria() // FUNCION PARA ANTES DE ACTUALIZAR
{
        session_start();

        if(!isset($_SESSION['con_admin']))
        { //si no se hay sesion abierta

```

```

        redirect('/clogin/index','refresh');
    }
else
    {
        $idCategoria = $_POST['id'];
        $name= $_POST['nombreCategoria'];

        $ActCategoria=$this->mcategorias-
>ActualizarCategoria($idCategoria, $name); //Creo la nueva facultad en la tabla
mdl_course_categories

        $PathCatActualizada=$this->mcategorias-
>BuscarPathById($idCategoria); //Busca el id de la ultima facultad creada
        $pathCategoriaActualizada= $PathCatActualizada -> path ;

        $partes = explode("/", $pathCategoriaActualizada);

        $npartes= count($partes);

        if ($npartes == 2)
            {

                redirect('/administrator/Management?idanio='.$partes["1"],'refresh');
            }
            else
            {
                if ($npartes == 3)
                {

                    redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"],'refr
esh');

                }
                else
                {

                    redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"].'&id
escuela='.$partes["3"],'refresh');

                }
            }
        }
    }
}

```

////////////////////////////////////

```

////////CONTROLOADORES PARA ADMINISTRACION DE CONCURSOS////////
////////

```

```

function VerShortNamesConcursosAJAX()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }

    else
    {
        $shortNames=$this->mconcursos->VerShortNamesConcursos();
        foreach ($shortNames as $sname) {
            if ( ($sname->shortname == trim($_GET['u'])) ) && ($sname-
>id != $_GET['id']) ){
                echo "<h4 class='bg-danger'>Este nombre corto ya
est&acute; en uso en otro concurso</h4>";
            }
        }
    }
}

```

```

function NuevoConcurso()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }
    else
    {
        $data['titulo']= 'Nuevo Concurso';
        $this->load->view('administracion/plantilla/Header',$data);
        $data['escuelas_listado']=$this->mcategorias->ObtenerSoloEscuelas();
        $this->load->view('administracion/concurso/formnuevo',$data);

        if(isset($_POST['submit']))
        {

```

```

$fullname= $_POST['fullname'];
$shortname=$_POST['shortname'];
$idcategoria=$_POST['category'];
$visible=$_POST['visible'];
$fdia=$_POST['dia'];
$fmes=$_POST['mes'];
$fanio=$_POST['anio'];

if (strlen($fdia)!=2){
    $fdia= "0".$fdia;
}

if (strlen($fmes)!=2){
    $fmes= "0".$fmes;
}

$finicio = $fanio."-".$fmes."-".$fdia;
$validarfecha = checkdate($fmes, $fdia, $fanio); // devuelve

```

TRUE si la fecha es valida

```

$validacion=$this->mconcursos-
>ConcursoRegistrado($shortname); //Se comprueba que el Usuario no este registrado

```

```

$existente=$validacion['0'] -> total;

```

```

if ($existente == '1' && $validarfecha == false )
{

```

```

    redirect('/administrator/NuevoConcurso?doble=si&fecha=no&fullname='.$fullname.'&shortn
ame='.$shortname.'&category='.$categoria.'&visible='.$visible.'&dia='.$fdia.'&mes='.$fmes.'&anio='.
$fanio.', 'refresh');

```

```

}
else
{

```

```

    if ($existente == '1' && $validarfecha == true )
    {

```

```

        redirect('/administrator/NuevoConcurso?doble=si&fecha=si&fullname='.$fullname.'&shortn
ame='.$shortname.'&category='.$categoria.'&visible='.$visible.'&dia='.$fdia.'&mes='.$fmes.'&anio='.
$fanio.', 'refresh');

```

```

    }
else
{

```

```

                                $nuevo=$this->mconcursos-
>InsertarConcurso($fullname, $shortname, $idcategoria, $visible, $finicio);
                                $idConcurso = $this->mconcursos-
>ObtenerIdConcurso($shortname); //Obtengo el id del Curso de la tabla mdl_course con el nombre
Corto
                                $idCon = $idConcurso -> id ;

                                $nuevoContextConcurso=$this->mcontext-
>InsertarConcursoContext($idCon); //Insertar el id del curso en la tabla mdl_context

                                $pathCategoria=$this->mcontext-
>ConsultarPathUltimaCategoria($idcategoria); //Obtengo el Path de la categoria que contiene el
curso
                                $path = $pathCategoria -> path;

                                $idContextConcurso = $this->mcontext-
>ObtenerIdContextConcurso($idCon); //Obtengo el id del Curso de la tabla mdl_context con el id del
Curso creado
                                $idContextCon = $idContextConcurso -> id ;

                                $pathContextConcurso = $path.'/'.$idCon;
                                $nuevoContextPathConcurso=$this-
>mcontext->ActualizarContextPathConcurso($pathContextConcurso,$idContextCon); //Actualizo el
path del curso en la tabla contexto

                                $PathCat=$this->mcategorias-
>BuscarPathById($idcategoria); //Busca el id de la ultima facultad creada
                                $pathCategoria= $PathCat -> path ;
                                $partes = explode("/", $pathCategoria);

                                $npartes= count($partes);

                                if ($npartes == 2)
                                {

```

```

redirect('/administrator/Management?idanio='.$partes["1"],'refresh');
    }
    else
    {
        if ($npartes == 3)
        {
            redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"],'refr
esh');
        }
        else
        {
            redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"].'&id
escuela='.$partes["3"],'refresh');
        }
    }
}
}
}

$this->load->view('administracion/plantilla/Footer');
}
}

function BorrarConcurso()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }

    else
    {
        if(isset($_GET['conborrar']))
        {
            $idConcurso= $_GET['conborrar'];

            $concurso=$this->mconcurso->
>BuscarCategoriaPorIdConcurso($idConcurso); //Busca el id de la ultima facultad creada
            $idCategoriaConcurso= $concurso -> category ;

```

```

        $borrarCon=$this->mconcursos-
>EliminarConcurso($idConcurso); //Se comprueba que el Usuario no este registrado

        $PathCat=$this->mcategorias-
>BuscarPathById($idCategoriaConcurso); //Busca el id de la ultima facultad creada
        $pathCategoria= $PathCat -> path ;
        $partes = explode("/", $pathCategoria);

        $npartes= count($partes);

        if ($npartes == 2)
        {

            redirect('/administrator/Management?idanio='.$partes["1"],'refresh');
        }
        else
        {
            if ($npartes == 3)
            {

                redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"],'refr
esh');
            }
            else
            {

                redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"].'&id
escuela='.$partes["3"],'refresh');
            }
        }
    }
}

function ModificarConcurso()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }
}

```

```

else
{
    $idConcurso = $_POST['id'];
    $fullname= $_POST['fullname'];
    $shortname= $_POST['shortname'];
    $idcategory = $_POST['category'];
    $visible = $_POST['visible'];
    $fdia=$_POST['dia'];
    $fmes=$_POST['mes'];
    $fanio=$_POST['anio'];

    $escena = $_POST['vista'];

    if (strlen($fdia)!=2){
        $fdia= "0".$fdia;
    }

    if (strlen($fmes)!=2){
        $fmes= "0".$fmes;
    }

    $finicio = $fanio."-".$fmes."-".$fdia;

    $ActConcurso=$this->mconcursos->ActualizarConcurso($idConcurso,
    $fullname, $shortname, $idcategory, $visible, $finicio); //Creo la nueva facultad en la tabla
    mdl_course_categories

    $PathCatActualizada=$this->mcategorias-
    >BuscarPathById($idcategory); //Busca el id de la última facultad creada
    $pathCategoriaActualizada= $PathCatActualizada -> path ;

    $pathCategoria=$this->mcontext-
    >ConsultarPathUltimaCategoria($idcategory); //Obtengo el Path de la categoria que contiene el
    curso

    $path = $pathCategoria -> path;

    $idContextConcurso = $this->mcontext-
    >ObtenerIdContextConcurso($idConcurso); //Obtengo el id del Curso de la tabla mdl_context con el
    id del Curso creado

```

```
$idContextCon = $idContextConcurso -> id ;
```

```
$pathContextConcurso = $path.'/'. $idConcurso;
```

```
$nuevoContextPathConcurso=$this->mcontext-
```

```
>ActualizarContextPathConcurso($pathContextConcurso,$idContextCon); //Actualizo el path del  
curso en la tabla contexto
```

```
if($escena == 'categoria')
```

```
{
```

```
    $partes = explode("/", $pathCategoriaActualizada);
```

```
    $npartes= count($partes);
```

```
    if ($npartes == 2)
```

```
        {
```

```
            redirect('/administrator/Management?idanio='.$partes["1"],'refresh');
```

```
        }
```

```
        else
```

```
        {
```

```
            if ($npartes == 3)
```

```
            {
```

```
                redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"],'refr  
esh');
```

```
            }
```

```
            else
```

```
            {
```

```
                redirect('/administrator/Management?idanio='.$partes["1"].'&idfacultad='.$partes["2"].'&id  
escuela='.$partes["3"],'refresh');
```

```
            }
```

```
        }
```

```
    }
```

```
    else
```

```
    {
```

```
        redirect('/administrator/verConcurso?viewid='.$idConcurso,'refresh');
```

```
    }
```

```
}
```

```

}

function verConcurso()
{
    session_start();

    if(!isset($_SESSION['con_admin']) )
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }
    else
    {
        if(isset($_GET['viewid']))
        {
            $idConcurso = $_GET['viewid'];

            $data['concurso']= $this->mconcursos-
>ObtenerDetalleConcursoPorID($idConcurso); //Busca el id del concurso
            $nombreConcurso= $data["concurso"] -> fullname;

            $data['titulo']= $nombreConcurso;
            $this->load->view('administracion/plantilla/Header',$data);

            $data['escuelas_listado']=$this->mcategorias-
>ObtenerSoloEscuelas(); // Para ventana modal de ACTUALIZAR concurso
            $this->load->view('administracion/concurso/verconcurso',$data);
            $this->load->view('administracion/plantilla/Footer');
        }
    }
}

```

```

////////////////////////////////////
/////CONTROADORES PARA MATRICULAR USUARIOS EN CONCURSOS////////////////////////////////
////////////////////////////////////

```

```

function verEnrolUsers()
{
    session_start();

    if(!isset($_SESSION['con_admin']) )
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }
    else
    {

```

```

        $idConcurso = $_GET['id']; // id del CONCURSO

        $data['concurso'] = $this->mconcursos-
>ObtenerConcursoPorId($idConcurso);
        if( !empty($data['concurso']) )
        {
            $nombreConcurso = $data["concurso"] -> fullname;

            $data['titulo'] = $nombreConcurso;

            $data['escuelas_listado'] = $this->mcategorias-
>ObtenerSoloEscuelas();
            $data['listado_evaluadores'] = $this->menrolls-
>ObtenerEvaluadoresConcurso($idConcurso);
            $data['listado_postulantes'] = $this->menrolls-
>ObtenerPostulantesConcurso($idConcurso);

        }
        else
        {
            $data['concurso'] = is_null($data['concurso']);
            $data['concurso'] = $this->mconcursos-
>ObtenerDetalleConcursoPorID($idConcurso); //Busca el id del concurso
            $data['concurso']-> numusuarios = 0;
            $nombreConcurso = $data["concurso"] -> fullname;
        }

        $this->load->view('administracion/plantilla/Header',$data);
        $this->load->view('administracion/enroll/enrolamiento',$data);
        $this->load->view('administracion/plantilla/Footer');

    }
}

function VerUsuariosEnrollAJAX()
{
    session_start();

    if(!isset($_SESSION['con_admin']))
    { //si no se hay sesion abierta
        redirect('/clogin/index','refresh');
    }
}

```

```

else
{
    $cedula = trim($_GET['cedula']);
    $idconcurso = trim($_GET['idconcurso']);

    $matriculado=$this->menrolls->VerUsuarioMatriculadoAJAX($cedula,
$idconcurso);

    if ( empty($matriculado) ) // si es el usuario esta matriculado
    {
        $usuarios=$this->usuarios->VerUsuariosParaAJAX($cedula);
        if (!empty($usuarios)) {
            if ($usuarios-> numberrole == 2){
                $rol= "Postulante";
                echo
                "<strong>".$rol."</strong><br>".$usuarios-> nombres." ".$usuarios-> apellidos."<br><span
                class='pull-left text-muted small'><em>".$usuarios-> email."</em></span>";
                echo "<input type='hidden' id='userid'
                name='userid' value='".$usuarios-> id.'">";
            }
            else{
                if ($usuarios-> numberrole == 1)
                {
                    $rol= "Evaluador";
                    echo
                    "<strong>".$rol."</strong><br>".$usuarios-> nombres." ".$usuarios-> apellidos."<br><span
                    class='pull-left text-muted small'><em>".$usuarios-> email."</em></span>";
                    echo "<input type='hidden' id='userid'
                    name='userid' value='".$usuarios-> id.'">";
                }
            }
            else
            {
                echo "No se encuentran usuarios";
            }
        }
    }
}
else
{
    echo "No se encuentran usuarios";
}
}
else
{

```

```
        echo "<p class='text-danger'>Este usuario ya se encuentra  
matriculado en este concurso</p>";
```

```
    }  
  }  
}  
  
function MatricularUsuarioConcurso()  
{  
    session_start();  
  
    if(!isset($_SESSION['con_admin']))  
    { //si no se hay sesion abierta  
        redirect('/clogin/index','refresh');  
    }  
    else  
    {  
        if(isset($_POST['submit']))  
        {  
            $userid= $_POST['userid'];  
            $concursoid=$_POST['concursoid'];  
  
            $contextConcurso=$this->mcontext-  
>ObtenerIdContextConcurso($concursoid);  
            $idContextoConcurso= $contextConcurso-> id;  
  
            $nuevoEnroll=$this->menrolls->MatricularUsuario($userid,  
$idContextoConcurso); //Se comprueba que el Usuario no este registrado  
  
            redirect('/administrator/verEnrolUsers?id='.$concursoid,'refresh');  
        }  
    }  
}
```

```
function DesmatricularUsuario()  
{  
    session_start();  
  
    if(!isset($_SESSION['con_admin']))  
    { //si no se hay sesion abierta  
        redirect('/clogin/index','refresh');  
    }  
    else  
    {  
        if(isset($_GET['userid']) && isset($_GET['concursoid']))
```

```

        {
            $userid= $_GET['userid'];
            $concursoid=$_GET['concursoid'];

            $contextConcurso=$this->mcontext-
>ObtenerIdContextConcurso($concursoid);
            $idContextoConcurso= $contextConcurso-> id;

            $borrarEnroll=$this->menrolls->DesMatricularUsuario($userid,
            $idContextoConcurso); //Se comprueba que el Usuario no este registrado

            redirect('/administrator/verEnrolUsers?id='.$concursoid,'refresh');
        }
    }
}

```

```

////////////////////////////////////
/////CONTROLADOR PARA CERRAR SESSION EN ESTE CONTROLADOR/////
////////////////////////////////////
function CerrarSesion()
{
    session_start();
    session_destroy();
    redirect( base_url(),'refresh');
}

```

ANEXO E

Manual técnico y manual de usuario



SIGCOM Manual Técnico

Versión: 1.0
Fecha: 10/08/2016

INTRODUCCIÓN

La finalidad de todo manual técnico es la de proporcionar al lector la lógica con la que se ha desarrollado una aplicación, la cual se sabe que es propia de cada programador, por lo que se considera necesario se documentada.

En su contenido podremos observar los aspectos más esenciales para poder utilizar de manera fácil esta aplicación.

OBJETIVO

Proporcionar y explicar al lector los recursos que se utilizaron para la creación del sistema el cual está realizado como una página web dinámica.

AMBITO

Este proyecto forma parte de una serie de proyectos orientados al desarrollo de aplicaciones de 3 capas.

Este componente software se desarrolló con la finalidad de lograr facilitar las tareas que se desarrollan en la Universidad.

ALCANCE

El proyecto intenta dar a conocer la habilidad que posee el desarrollador de este proyecto en la solución de este tipo de problemas, está destinado tanto a aquellos que deseen obtener información sobre la programación en PHP, para que sirva de guía para futuros cambios y actualizaciones en la aplicación “SIGCOM”.

REQUERIMIENTOS

Instalación y ejecución de los componentes necesarios para el desarrollo de “SIGCOM” para la automatización del proceso de concursos de méritos y oposición.

Configuración del Proyecto

INSTALACIÓN DE HERRAMIENTAS

INSTALACIÓN DE HTTP APACHE SERVER

Para poder emular un servidor en Windows, hacemos uso de XAMPP, el cual nos proporciona la instalación de un servidor web, así como PHP y como gestor de base de datos MySQL.

Descargar del sitio oficial de Apache Friends:

<https://www.apachefriends.org/xampp-files/5.6.23/xampp-win32-5.6.23-0-VC11-installer.exe>

Al ejecutar el instalador XAMPP nos muestra dos avisos:

El primero aparece si en el ordenador hay instalado un antivirus: (Ver Figura No. 1)

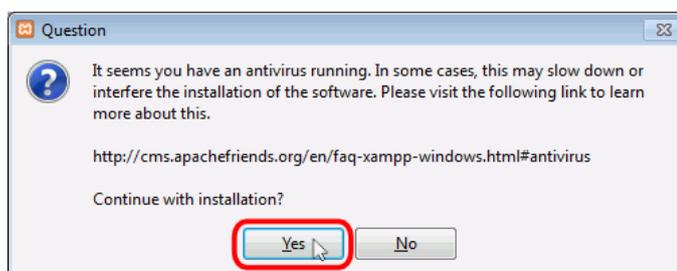


Figura No. 1 - Aviso1

El segundo aparece si está activado el Control de Cuentas de Usuario y recuerda que algunos directorios tienen permisos restringidos: (Ver Figura No. 2)

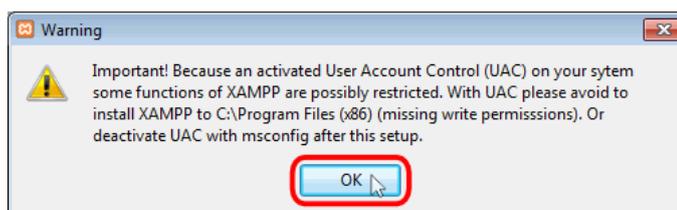


Figura No. 2 - Aviso 2

A continuación se inicia el asistente de instalación. Para continuar, hay que hacer clic en el botón "Next". (Ver Figura No. 3)

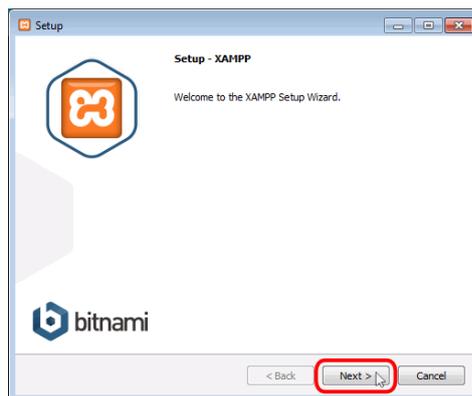


Figura No. 3- Ventana de bienvenida

En la pantalla de selección de componentes puede elegirse la instalación o no de estos componentes. para el sistema solo se necesita solo PHP. Para continuar, hay que hacer clic en el botón "Next". (Ver Figura No. 4)

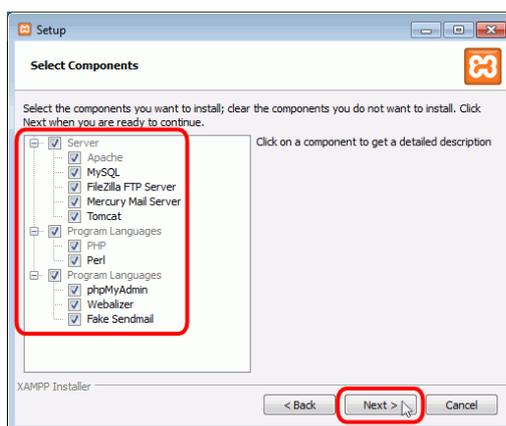


Figura No. 4- Ventana de selección de componentes

En la siguiente pantalla se puede elegir la carpeta de instalación de XAMPP. La carpeta de instalación predeterminada es C:\xampp. Para continuar la configuración de la instalación, hay que hacer clic en el botón "Next".(Ver Figura No. 5)

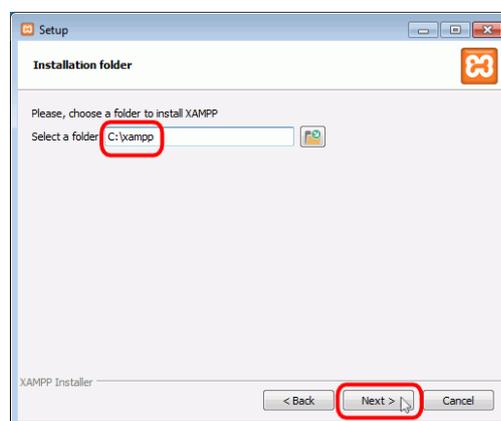


Figura No. 5- Ventana para seleccionar la ruta de instalación de XAMPP

Para empezar la instalación de XAMPP, hay que hacer clic en el botón "Next" en la pantalla siguiente. (Ver Figura No. 6)

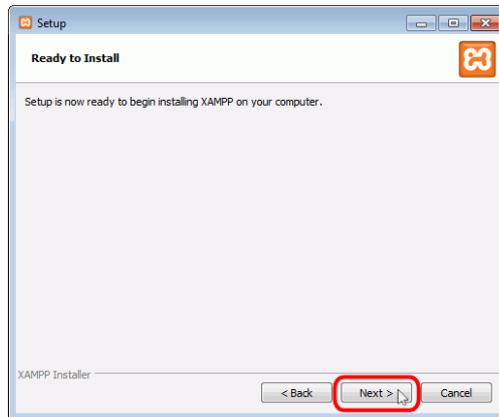


Figura No. 6- Ventana para iniciar instalación

A continuación, se inicia el proceso de copia de archivos, que puede durar unos minutos. (Ver Figura No. 7)

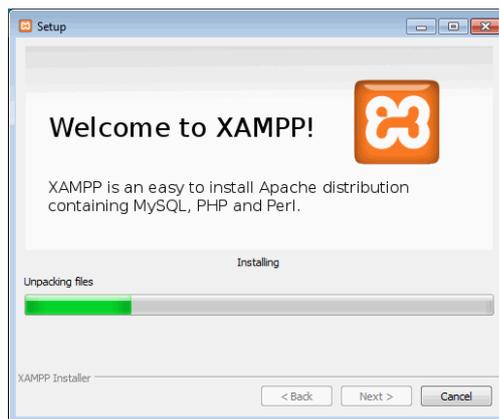


Figura No. 7- Proceso de instalación de XAMPP

A continuación se muestra la pantalla que confirma que XAMPP ha sido instalado. Hay que hacer clic en el botón "Finish". (Ver Figura No. 8)



Figura No. 8- Ventana de **instalación finalizada** XAMPP

CONFIGURACIÓN DE PHP

El sistema SIGCOM usa como gestor de Base de datos PostgreSQL, por lo que es necesario habilitar la extensión de postgresql en nuestro archivo de configuración de PHP ubicado en: C:\xampp\php\php.ini (Ver Figura No. 9)

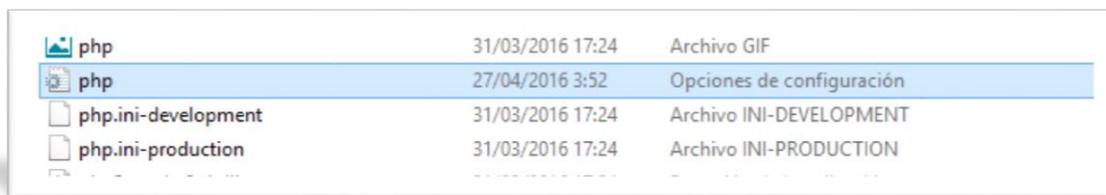


Figura No. 9- Archivo de configuración de PHP

Abrimos el archivo con un editor de texto (Ver Figura No. 10)

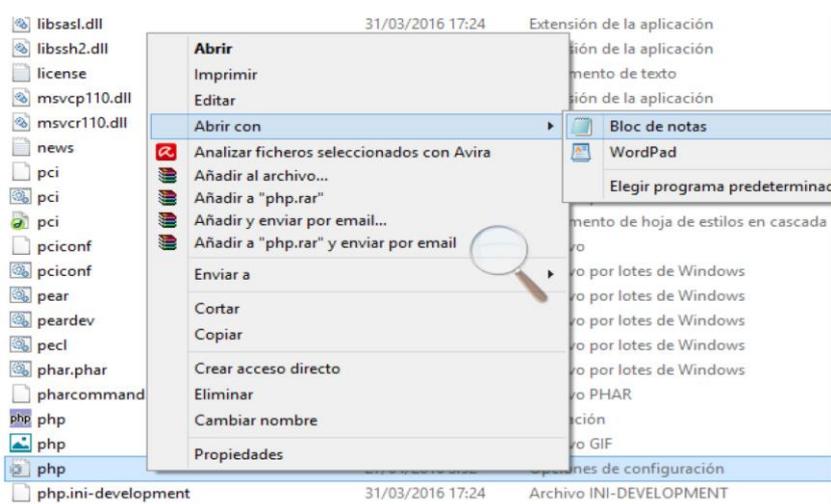


Figura No. 10- Abrir archivo con editor de texto

Se abrirá el archivo y a continuación presionamos la combinación de teclas CONTROL + B, si se abrirá una ventana para buscar. En esta ventana ingresaremos “extensión=php_pdo_pgsql.dll”. (Ver Figura No. 11).

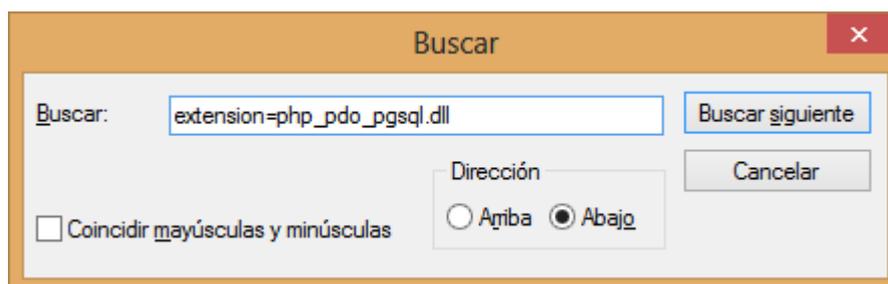
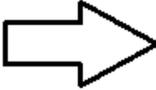


Figura No. 11- Ventana de búsqueda

Borramos el “;” que se encuentra al inicio para habilitar esta extensión. Repetimos el procedimiento 3 con la extensión “extensión=php_pgsql.dll” e **igualmente borramos** el “;” inicial. (Ver Figura No. 12).

```
;  
;extension=php_pdo_odbc.dll  
;extension=php_pdo_pgsql.dll  
extension=php_pdo_sqlite.dll  
;extension=php_pgsql.dll  
extension=php_shmop.dll
```



```
;  
extension=php_pdo_odbc.dll  
extension=php_pdo_pgsql.dll  
extension=php_pdo_sqlite.dll  
extension=php_pgsql.dll  
extension=php_shmop.dll
```

Figura No. 12- Configuración Final de PHP

Cambiamos el tiempo de ejecución de PHP de 30 a 600

“max_execution_time=600”

Presionamos la combinación de teclas CONTROL + G para guardar la configuración.

INSTALACIÓN DE COMPOSER

Composer es una herramienta para gestionar las dependencias en PHP. Te permite declarar las librerías de las cuales tu proyecto depende o necesita y las instala en el proyecto por ti.

Descargar de su sitio oficial:

<https://getcomposer.org/Composer-Setup.exe>

Una vez que la descarga finalice, ejecuta el instalador y haz click en Next. (Ver Figura No. 11).

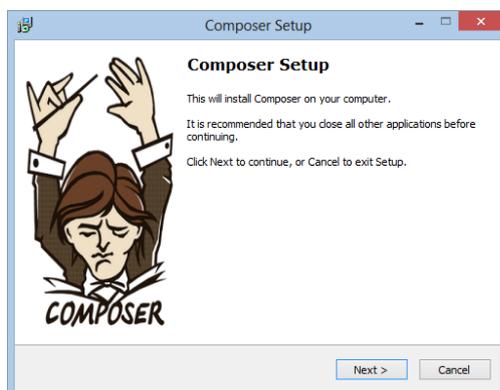


Figura No. 13- Pantalla de bienvenida de instalación de composer

A continuación nos pide que indiquemos la ruta del ejecutable de PHP, en este proyecto se trabaja con XAMPP, el ejecutable de PHP se encuentra en la ruta “C:\xampp\php\” y seleccionamos “php.exe”, luego click en Next. (Ver Figura No. 14).

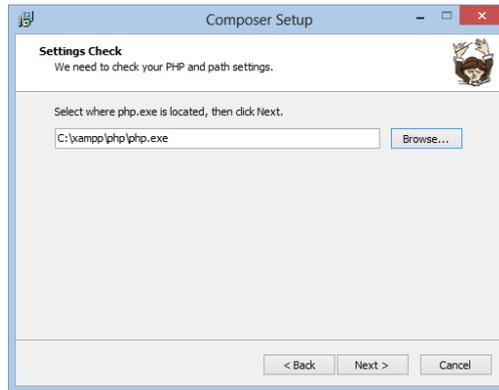


Figura No. 14- Ventana para seleccionar la ruta de instalación de Composer
En este punto el instalador de Composer nos muestra la configuración de la instalación, simplemente le damos click a Install.

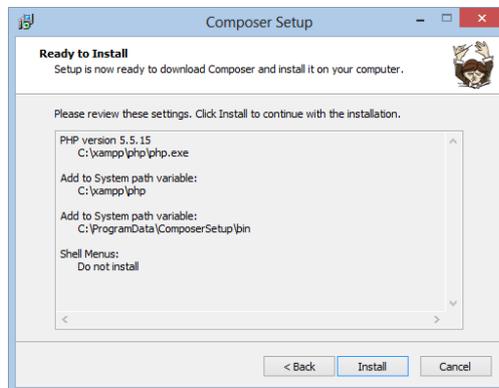


Figura No. 15- Ventana de **configuración de composer**
A continuación se muestra la pantalla que confirma que COMPOSER ha sido instalado. Hay que hacer clic en el botón "Finish". (Ver Figura No. 16)



Figura No. 16- Ventana de **instalación finalizada COMPOSER**

INSTALACIÓN DE POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

Descargar de su sitio oficial:

www.postgresql.org/download/windows

Luego aparecerá la siguiente ventana y presionamos sobre siguiente. (Ver Figura No. 17)



Figura No. 17 - Ventana de bienvenida instalación PostreSQL

Se mostrara una ventana en donde pondremos la dirección o el directorio de instalación en donde vamos a guardar el programa. (Ver Figura No. 18)



Figura No. 18- Directorio de instalación de PostgreSQI

Se mostrará una nueva ventana, aquí pondremos la dirección de donde vamos a guardar los datos. (Ver Figura No. 19)



Figura No. 19- Directorio de datos de PostgreSQL

Pulsamos nuevamente siguiente, ahí te aparecerá una ventana en la que nos pedirá una contraseña de usuario de PostgreSQL, ingresaremos nuestra contraseña. (Ver Figura No. 20)

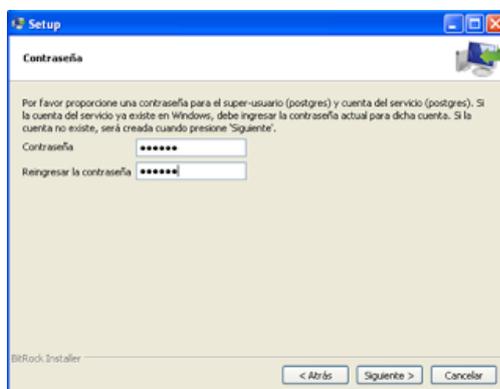


Figura No. 20- Ventana de solicitud de contraseña

Aparecerá una ventana en la que pedirá el puerto por donde se comunicara el programa, este aparecerá por default, es el puerto 5432, dejamos el mismo número de puerto: (Ver Figura No. 21)

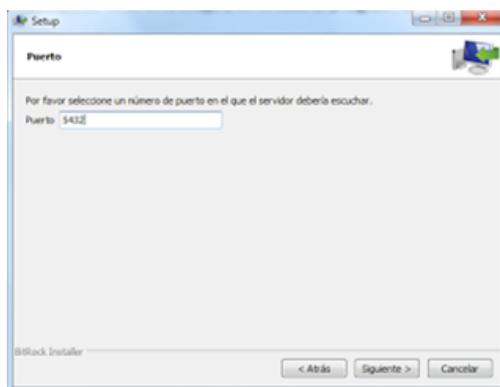


Figura No. 21- Ventana de solicitud de puerto

En la configuración regional, dejaremos la que da por default y haremos clic en siguiente. Aparecerá una ventana en la cual indicará que el programa está listo para instalarse. posterior a esto haremos clic en siguiente. (Ver Figura No. 22)

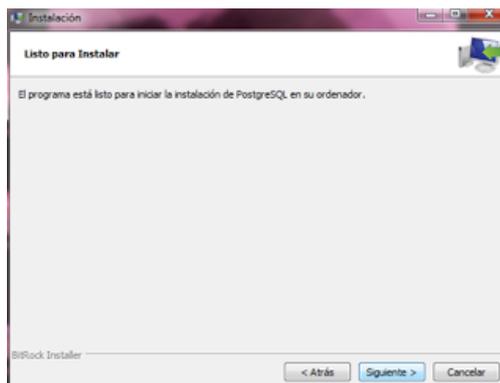


Figura No. 22- Ventana para iniciar instalación de PostgrSQL

Aparecerá una ventana en la cual observaremos que se está instalando PostreSQL. (Ver Figura No. 23)

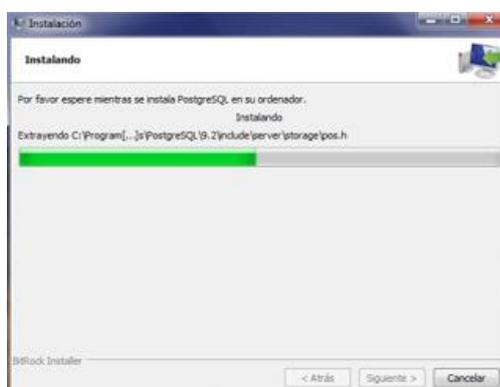


Figura No. 23 - Proceso de instalación de PostgrSQL

Observaremos que se está instalando PostreSQL. (Ver Figura No. 24)



Figura No. 24- Ventana de instalación finalizada PostgrSQL

INSTALACIÓN DE LARAVEL 9.1

Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti". Fue creado en 2011 y tiene una gran influencia de frameworks como Ruby on Rails, Sinatra y ASP.NET MVC.

Descargamos laravel desde su repositorio de Github (Ver Figura No. 25)

<https://github.com/laravel/laravel/tree/5.1>

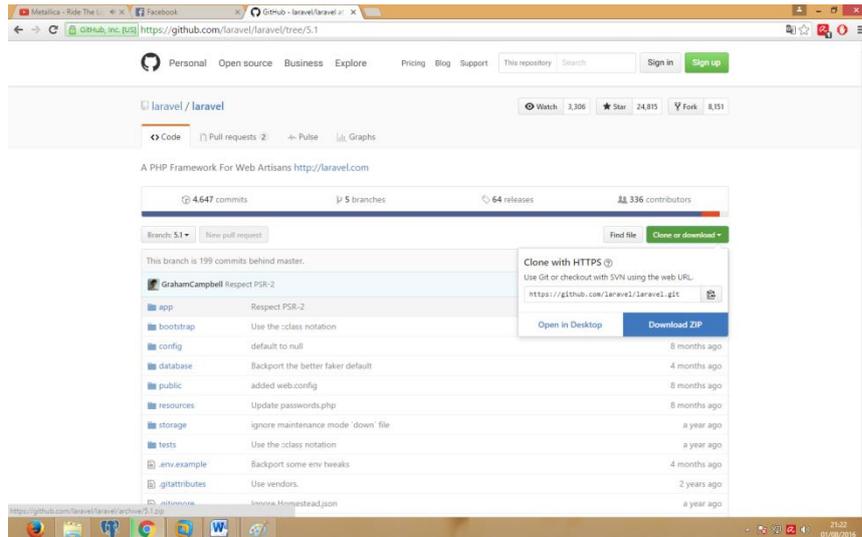


Figura No. 25 - Descarga de Laravel 5.1 en GitHub

Extraemos el contenido de la descarga en el directorio "C:\xampp\htdocs". (Ver Figura No. 26)

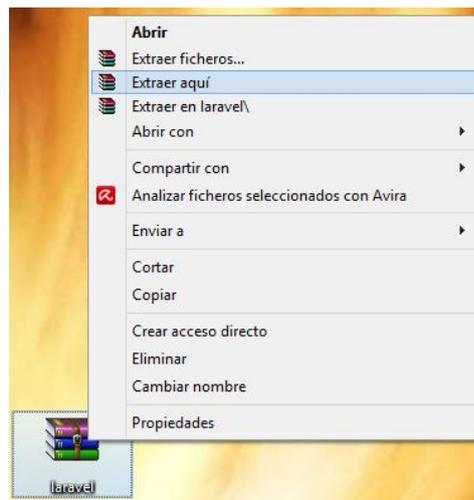


Figura No. 26- Descomprimir el zip descargado Laravel

Cambiamos de nombre a este archivo descomprimido con el nombre de nuestro proyecto, para este caso “sigcom”.

Abrimos XAMPP e iniciamos el Apache (Ver Figura No. 27)

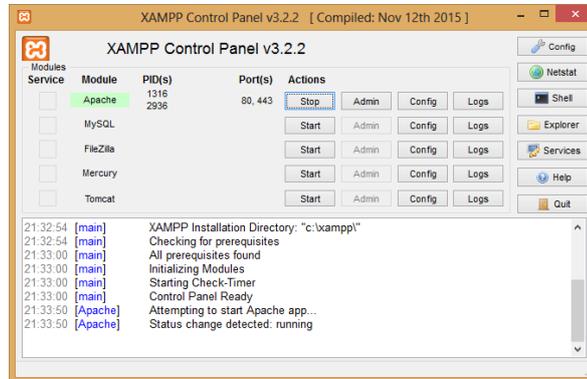


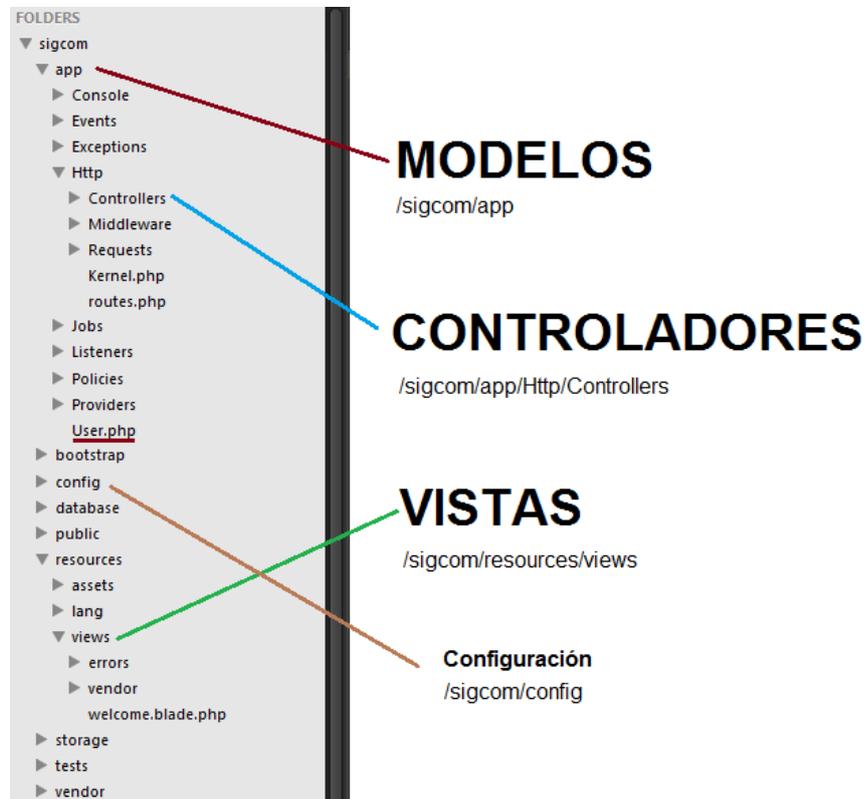
Figura No. 27- Iniciar servicio de Apache

Abrimos el navegador y en la barra de direcciones ingresamos: <http://localhost/sigcom/public/> y presionamos la tecla ENTER.

Nos aparece la siguiente ventana, esto nos indicara que Laravel ha sido instalado correctamente.

CONFIGURACIÓN DEL PROYECTO

Laravel es un framework MVC, el cual tiene la siguiente estructura. (Ver Figura No. 28)



CONEXIÓN A LA BASE DE DATOS

Abrimos nuestro proyecto con un editor de código como “Sublime Text 2”.

Abrimos el archivo de configuración de la base de datos localizado en: sigcom/config/database.php.

Nos dirigimos a la sección de conexiones e ingresamos la información de nuestro gestor de base de datos en este caso pgsq.

```
'pgsql' => [  
'driver' => 'pgsql',  
'host' => 'localhost',  
'port' => '5432',
```

```
'database' => 'concurso',
'username' => 'postgres',
'password' => '*****',
'charset' => 'utf8',
'prefix' => "",
'schema' => 'public',
],
```

CORREO ELECTRÓNICO

Abrimos nuestro proyecto con un editor de código como “Sublime Text 2”.

Abrimos el archivo de configuración de la base de datos localizado en:
sigcom/config/mail.php

Ingresamos la información de nuestro servidor de correo. En este ejemplo se usa una cuenta de office 365.

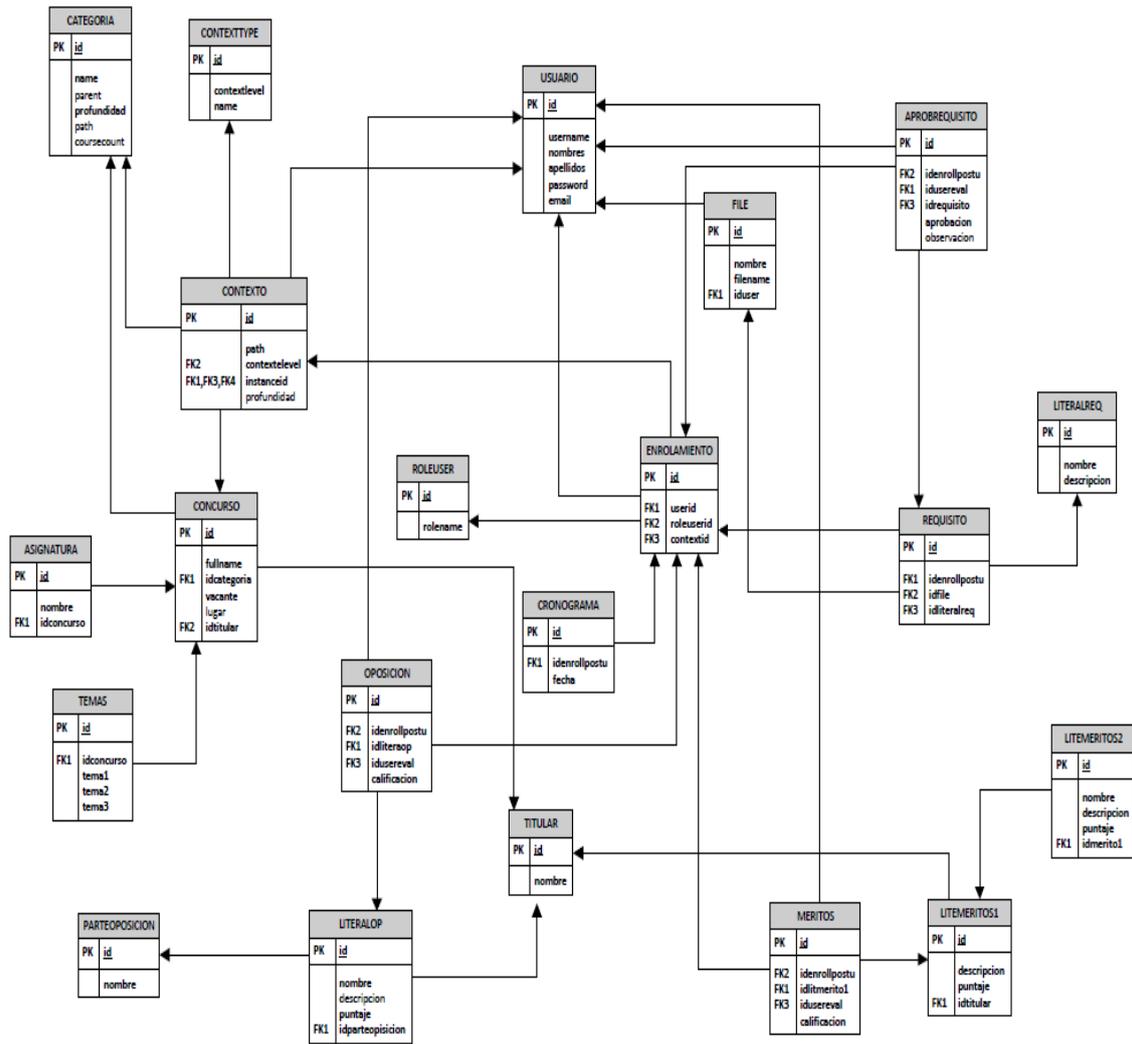
```
<?php
```

```
return [
```

```
    'driver' => 'smtp',
    'host' => 'smtp.office365.com',
    'port' => 587,
    'from' => ['address' => 'concursomeritos@unach.edu.ec', 'name' => 'Concurso de
Méritos y Oposición Unach'],
    'encryption' => 'tls',
    'username' => 'concursomeritos@unach.edu.ec',
    'password' => '*****',
    'sendmail' => '/usr/sbin/sendmail -bs',
];
```

BASE DATOS

DIAGRAMA RELACIONAL





Unach

UNIVERSIDAD NACIONAL DE CHIMBORAZO

Libres por la Ciencia y el Saber

SIGCOM **Manual de Usuario**

Versión: 1.0

Fecha: 10/08/2016

INTRODUCCION

En este documento se describirá los objetivos e información clara y concisa de cómo utilizar el Sistema de Concursos de Méritos y Oposición de la UNACH para el web y su funcionamiento.

El Sistema de Concursos de Méritos y Oposición de la UNACH con el objetivo de brindar facilidades a un postulante para consultar su situación en cada una de las fases de un concurso, sus datos de registros, entre otras opciones, y a los evaluadores para realizar el respectivo registro de calificaciones.

Es de mucha importancia consultar este manual antes y/o durante la visualización de las páginas, ya que lo guiará paso a paso en el manejo de las funciones en él. Con el fin de facilitar la comprensión del manual, se incluye gráficos explicativos.

OBJETIVO

El presente documento tiene como objetivo primordial de esta Manual es dar una guía al usuario a utilizar el **Sistema de Automatización de los concursos de Méritos y Oposición de la Universidad Nacional de Chimborazo “SIGCOM”**, obteniendo información de cada una de las fases de los concursos para informar a los postulantes de un concurso sobre situación actual; y comprende:

Guía para acceder al Sistema de Concursos de Méritos y Oposición de la Universidad Nacional de Chimborazo “SIGCOM”

Conocer cómo utilizar el sistema, detallando e ilustrando cada una de las funciones.

Conocer el alcance de toda la información por medio de una explicación detallada e ilustrada de cada una de las páginas que lo conforman.

DIRIGIDO A

Este manual está orientado a los Usuarios que se encuentran involucrados en la etapa de Operación del sitio web del Sistema de Concursos de Méritos y Oposición de la UNACH, es decir para los postulantes, evaluadores y administrador, que van a interactuar con el sitio web.

LO QUE DEBE CONOCER

Los conocimientos mínimos que deben tener los usuarios que operarán las páginas y deberán utilizar este manual son:

Conocimientos básicos de Navegación en Web.

Conocimientos básicos acerca de Programas con manejo de archivos.

Conocimiento básico de Internet.

Conocimiento básico de Windows.

CONVENCIONES Y ESTÁNDARES A UTILIZAR

Entre las Convenciones y estándares a utilizar tenemos las siguientes:

Convenciones de Formato de Texto

FORMATO	TIPO DE INFORMACIÓN
Viñeta Numérica	Numeración de procedimiento paso a paso
Negrita	Términos a resaltar su importancia
Menor que (<) Mayor que (>)	Nombres de opciones que puede elegir el usuario, ya sea haciendo uso del mouse o teclado.
Gráfico del botón	Palabra botón y a su lado irá la representación gráfica.

Convenciones del Uso del Mouse

TÉRMINO	SIGNIFICADO
Señalar	Colocar el extremo superior del Mouse sobre el elemento que se desea señalar.
Hacer Clic	Presionar el botón principal del Mouse (generalmente el botón izquierdo) y soltarlo inmediatamente.

ESPECIFICACIONES TÉCNICAS

Para la Implementación del sitio web Sistema de Concursos de Méritos y Oposición requerimos lo siguiente:

HARDWARE

Cliente Requerido

Procesador Core Duo 1+ GHz.

Memoria Ram 1 Gb.

Disco Duro 40 Mb de espacio libre

Equipo Servidor

Procesador Core i3 4 GHz

Memoria RAM 4 GB. La cantidad de memoria Ram varía según la cantidad de usuarios

Web, así como de la cantidad de tareas extras que ejecute el servidor.

Disco Duro DD con 80 Gb LIBRES para datos

SOFTWARE

Software del Cliente

El Software soporta Google Chrome, Mozilla Firefox, Safari, Opera entre otros.

Es de vital importancia utilizar navegadores que permitan utilizar JavaScript y Cookies.

Software del Servidor

Centos 6.5, Linux(kernel 2.6)

Apache HTTP Server

PostgreSQL 9.4

Composer 1.2.0

INGRESO Y REGISTRO AL SISTEMA

Como acceder al Sistema SIGCOM

Ubíquese en el ícono del navegador, y haga doble click. **(Ver Figura No. 1)**



Figura No. 1- Ubicación del ícono del navegador

A continuación aparecerá la pantalla del navegador, ubíquese en la barra de direcciones y escriba la dirección del sitio web concursos.unach.edu.ec y presionamos la tecla Enter. **(Ver Figura No. 2)**

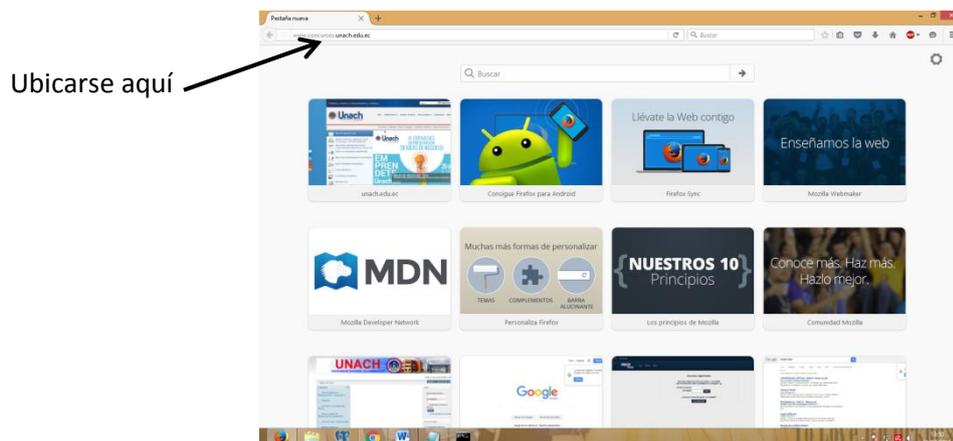


Figura No. 2- Ubicación de barra de direcciones

A continuación aparecerá la Pantalla Principal del Sistema de Concursos de Méritos y Oposición de la UNACH “SIGCOM”. (Ver Figura No. 3)

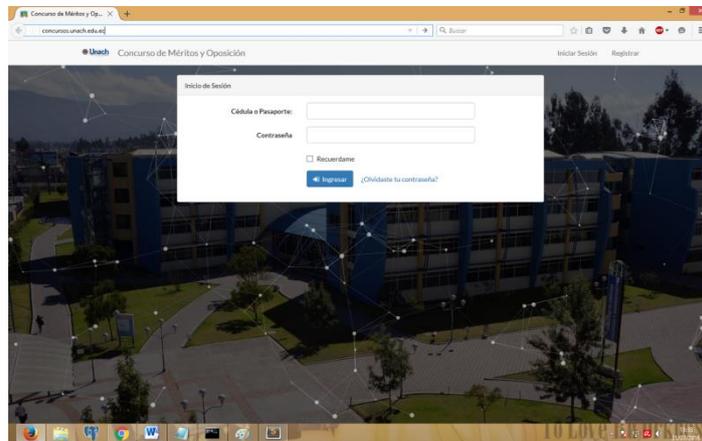
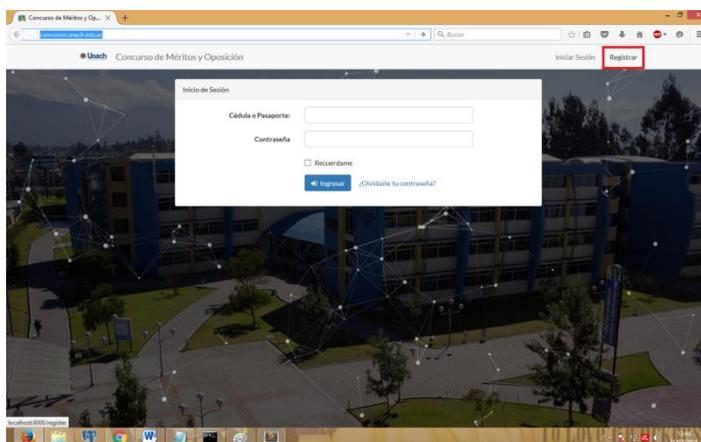


Figura No. 3- Pantalla Sistema SIGCOM

Como registrarse en el Sistema SIGCOM

Para registrarnos hacemos click en el botón “Registrar”. (Ver Figura No. 4)



← Clic aquí

Figura No. 4- Ubicación del enlace para el registro

A continuación aparecerá un formulario que el usuario deberá llenar con su información personal y hacemos click en “Registrarme”. (Ver Figura No. 5)

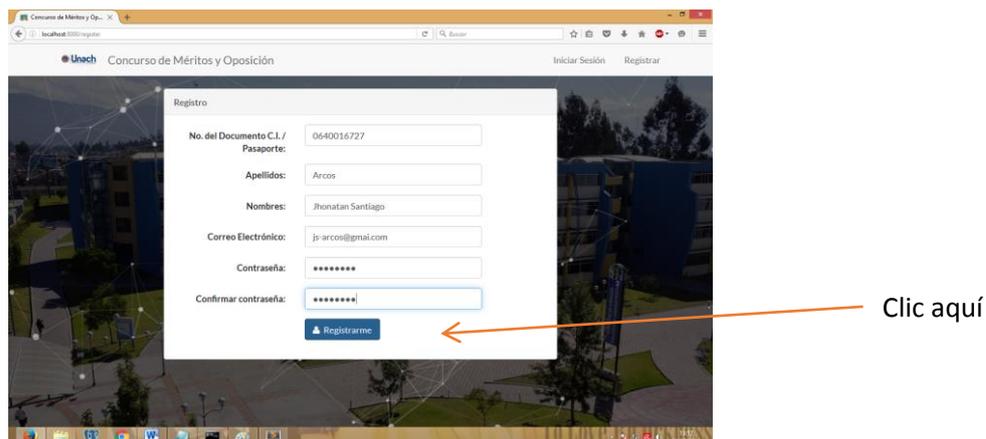


Figura No. 5- Formulario de registro

A continuación aparece la Pantalla Principal del Sistema SIGCOM con nuestra cuenta registrada. (Ver Figura No. 6)

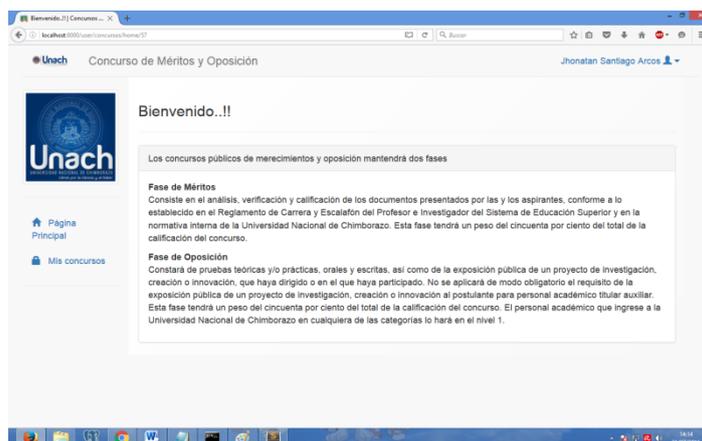


Figura No. 6- Página principal del sistema

Como ingresar con nuestras credenciales al Sistema SIGCOM

Escriba en la barra de direcciones: “concursos.unach.edu.ec” y presionamos la tecla Enter. A continuación en el cuadro de texto superior ingresamos nuestro número de cédula o Pasaporte, en el segundo cuadro de texto ingresamos la contraseña y hacemos click en Ingresar. (Ver Figura No. 7)

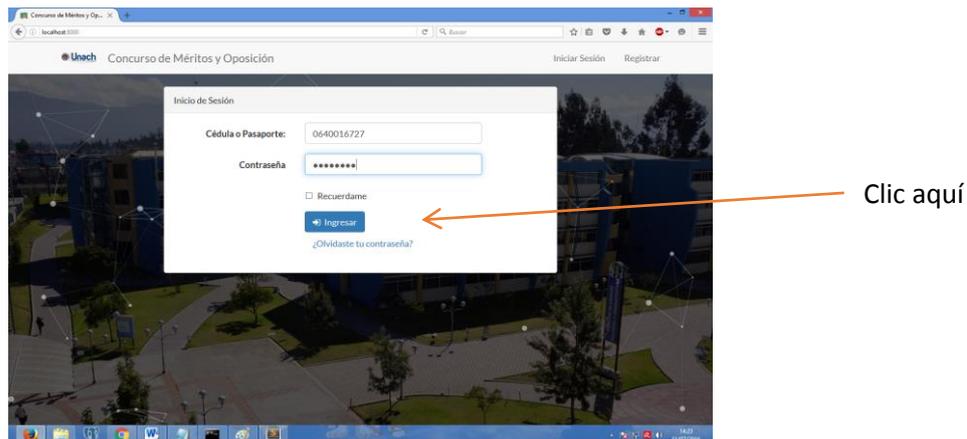


Figura No. 7- Ingreso de credenciales para acceder al sistema

A continuación aparece la Pantalla Principal del Sistema SIGCOM con nuestra cuenta registrada. (Ver Figura No. 8)

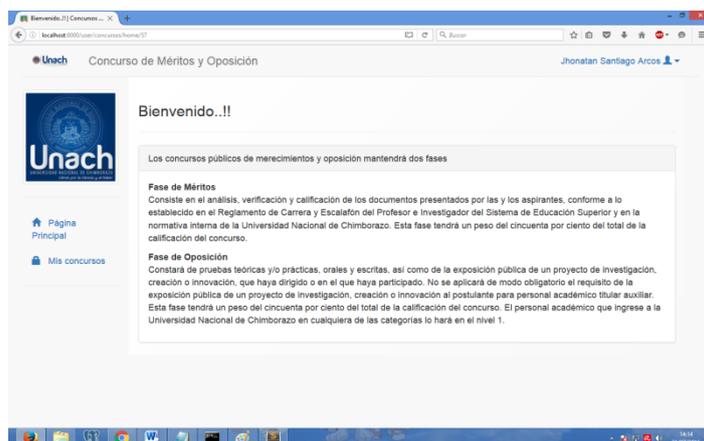


Figura No. 8- Página principal del sistema

OPERACIÓN DEL SISTEMA

El Sistema SIGCOM cuenta con 2 tipos de cuentas: Administrador y usuarios. Al acceder con el usuario y contraseña, tanto el administrador como los usuarios cuentan con distintas opciones que se detallan a continuación.

OPERACIONES DEL ADMINISTRADOR

El administrador cuenta con las siguientes funciones (Ver Figura No. 9)

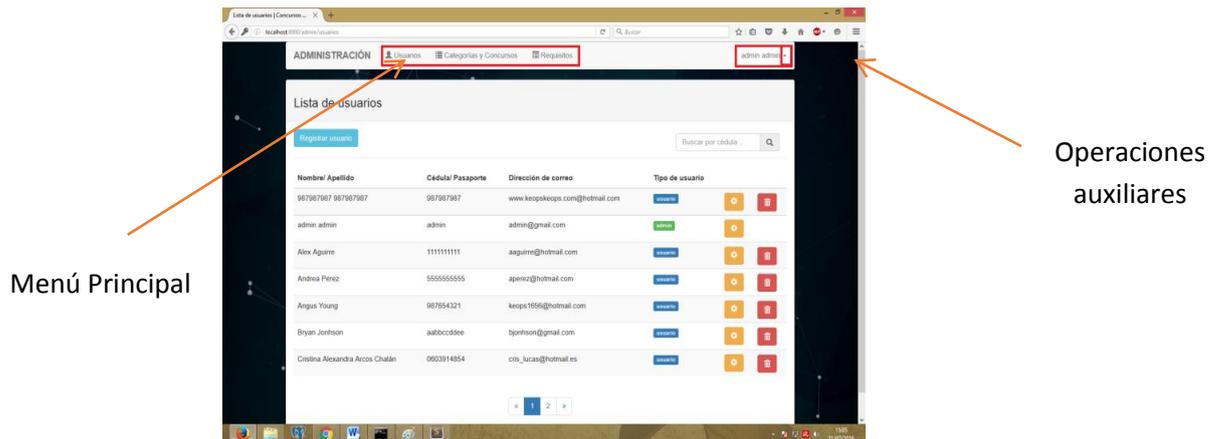


Figura No. 9- Opciones del Sistema - Administrador

Opciones Auxiliares

Entre estas tenemos:

Salir: Nos permite salir del Sistema.

Para salir del Sistema hacemos click en el link Salir. (Ver Figura No. 10)

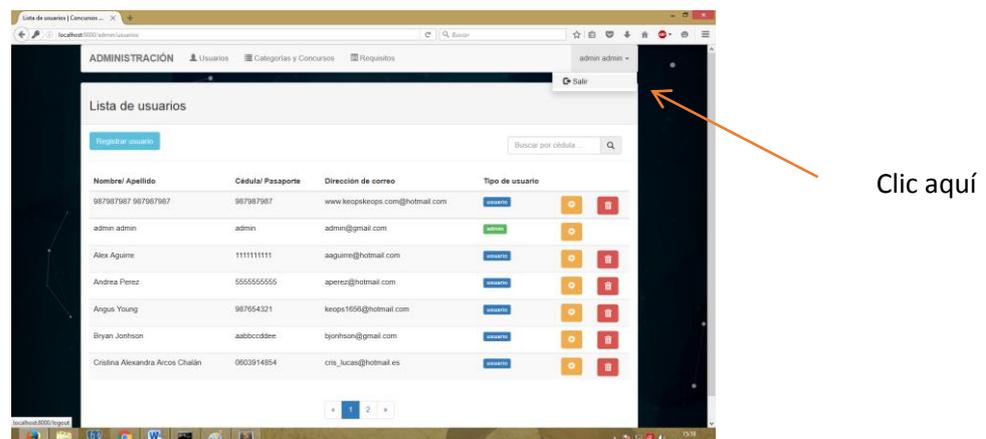


Figura No. 10- Link Salir

Menú principal

El Sistema Académico en Línea de ESPOL cuenta con un Menú Principal el cual se lo detalla a continuación. (Ver Figura No. 11)

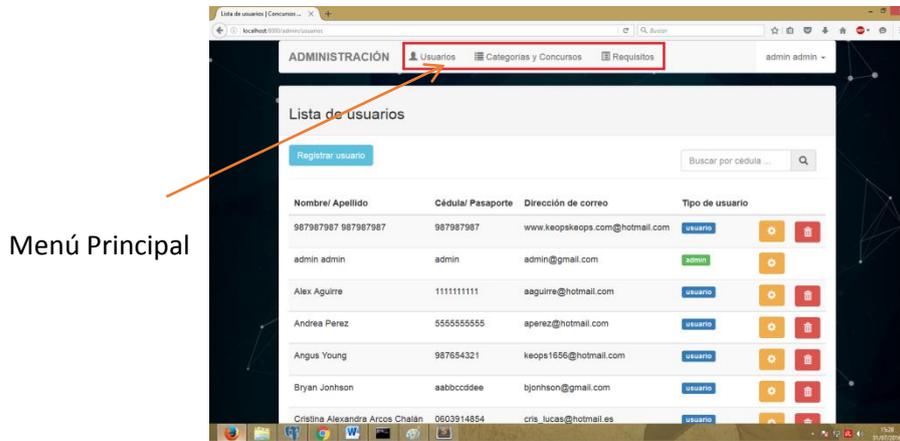


Figura No. 11- Menú Principal

Usuarios

Al hacer click en el enlace de usuarios. El sistema nos proporciona el listado de todos los usuarios existentes con información detallada de cada uno de ellos y varias operaciones que se pueden realizar en ellos. También nos muestra el botón para registrar un nuevo usuario. (Ver Figura No. 12)

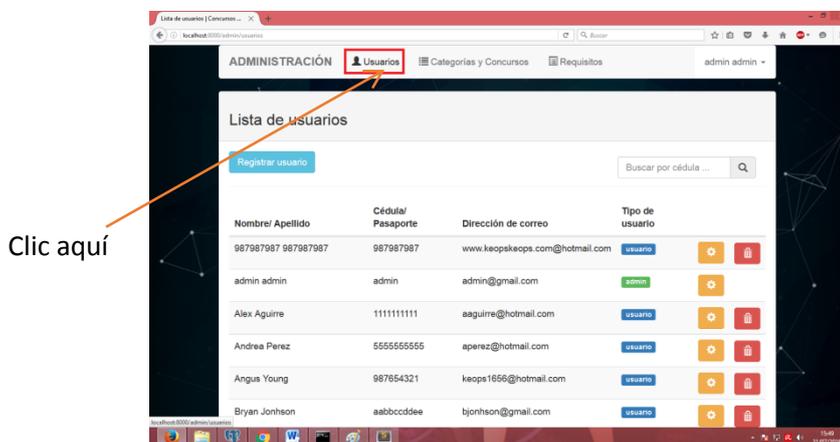


Figura No. 12- Link de Gestión de usuarios

Paginación de usuarios.

El sistema nos brinda un listado de todos los usuarios, sin embargo el sistema divide esta lista en páginas. En cada página se visualiza 8 usuarios. Para seguir avanzando en la lista, se ha dispuesto de una barra de paginación ubicada en la parte inferior de la lista de usuarios. Esta barra nos permite desplazarnos por cada una de las páginas de la lista de usuarios. (Ver Figura No. 13)

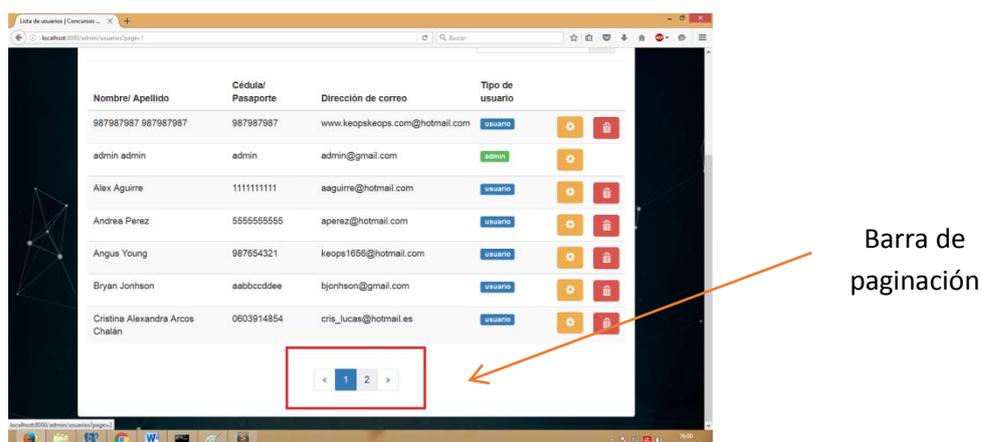


Figura No. 13- Barra de Paginación

Registro de usuarios.

Hacemos clic en el botón de Registrar usuario, el sistema nos presentara un formulario para agregar un nuevo usuario. (Ver Figura No. 14)

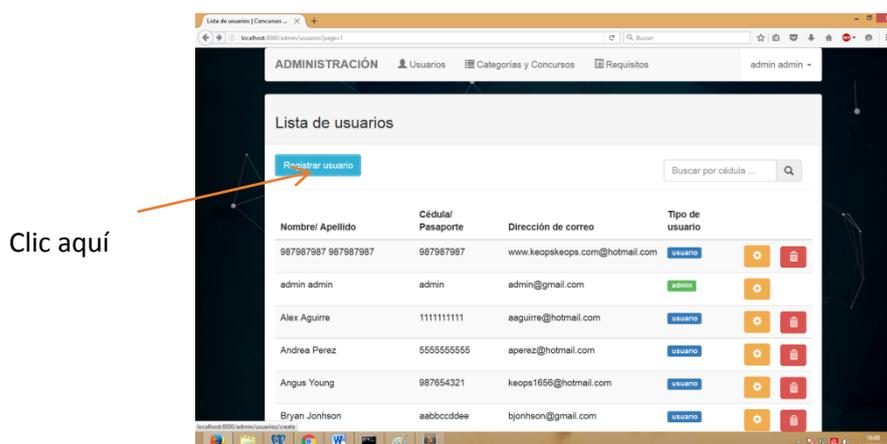


Figura No. 14 - Botón Registrar usuario

Completamos el formulario con información de usuario y seleccionamos el tipo de usuario: “Administrador” o “Usuario” y hacemos clic en el botón Registrar. En caso de querer anular el registro de usuario hacemos clic en el botón Cancelar. (Ver Figura No. 15)

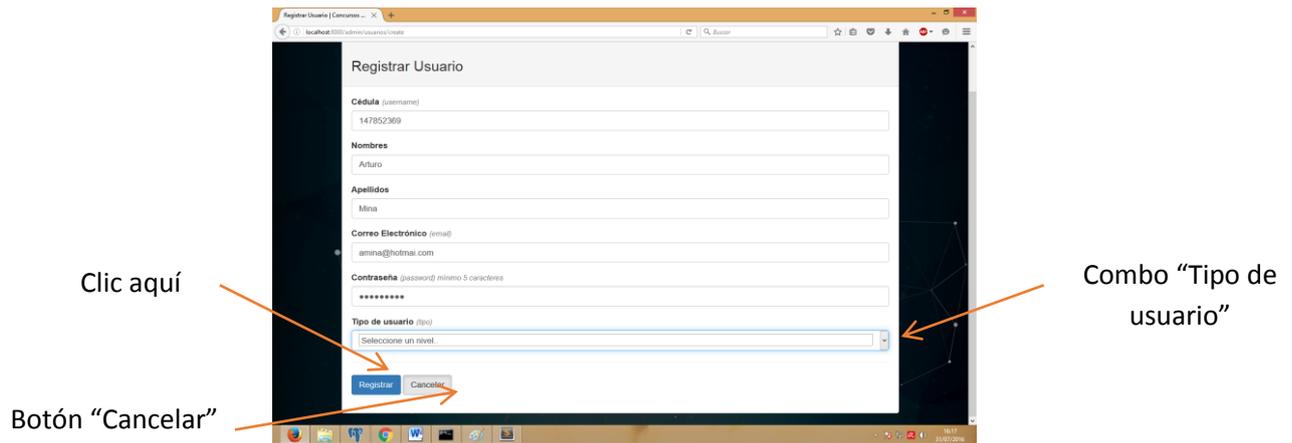


Figura No. 15- Formulario de registro de usuario

El sistema notificara que él usuario ha sido creado exitosamente. El nuevo usuario aparecerá en el listado de usuarios. (Ver Figura No. 16)

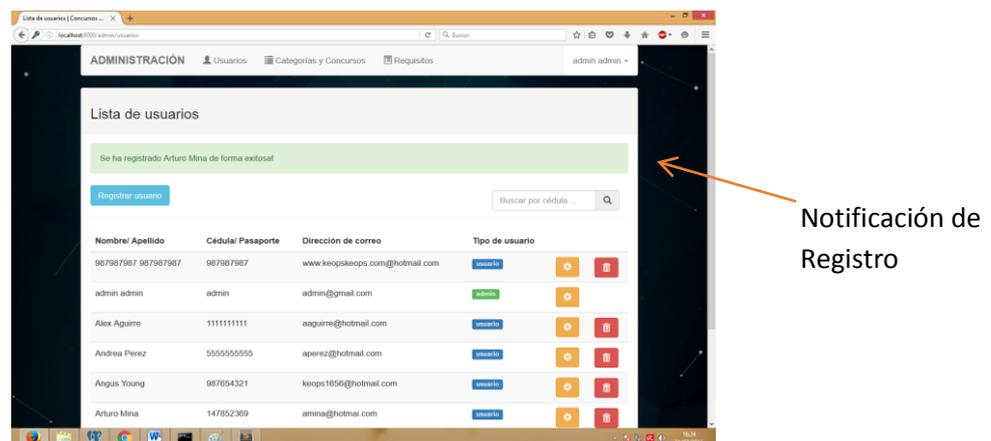


Figura No. 16 - Notificación de registro de usuario

Buscar usuario

El sistema nos permite buscar a un usuario pasándole como parámetro un número de cédula en la barra de búsqueda. (Ver Figura No. 17)

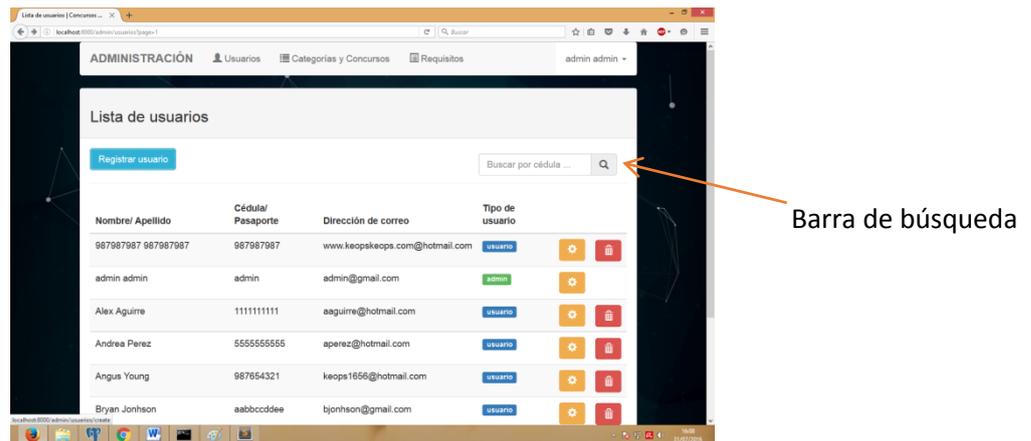


Figura No. 17- Barra de búsqueda de usuario

En el campo de texto de la barra de búsqueda ingresamos el número de cédula del usuario y presionamos la tecla Enter.

El sistema nos presenta el resultado de la búsqueda del usuario. En caso de no hallar ninguna coincidencia se presentara una lista vacía. **(Ver Figura No. 18)**

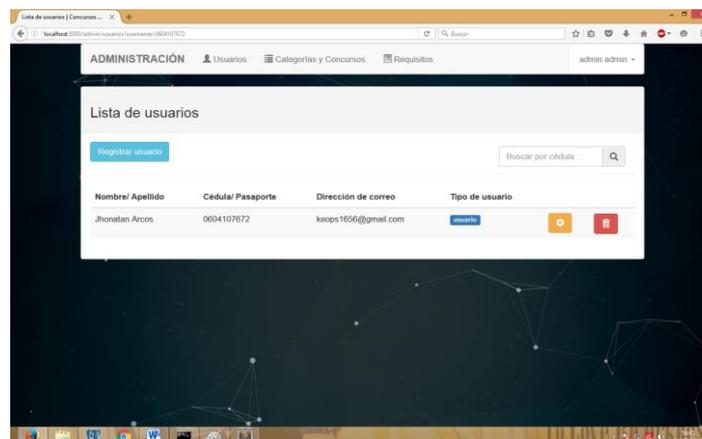


Figura No. 18 - Resultado de la búsqueda de usuario

Actualizar usuario

En el listado de usuarios, cada usuario dispone de un botón naranja con un ícono de un engrane que permite al administrador actualizar la información personal de un usuario, restablecer contraseña y cambiar el tipo de usuario. **(Ver Figura No. 19)**

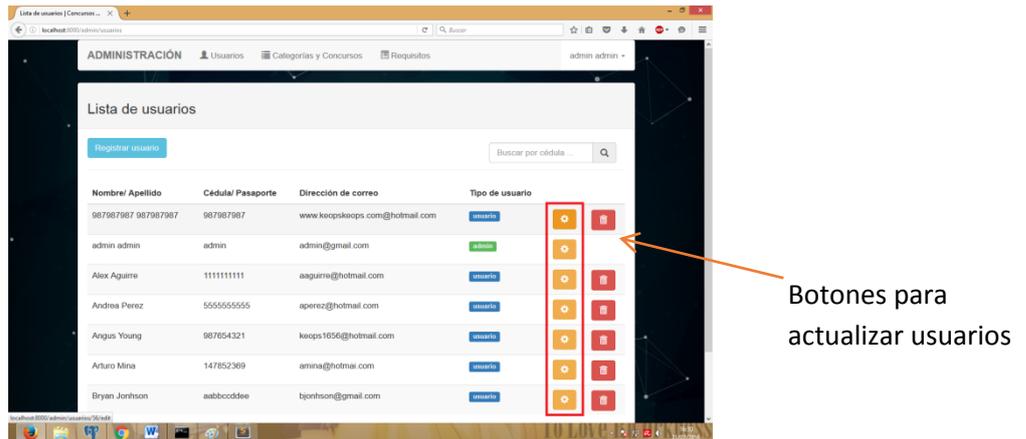


Figura No. 19 - Botón para actualizar usuario

Después de dar clic en el botón de actualizar de un usuario, se nos presentara un formulario con los campos asignados con la información actual del usuario. Editamos cada uno de los campos que se desea actualizar y a continuación presionamos el botón Actualizar usuarios. (Ver Figura No. 20)

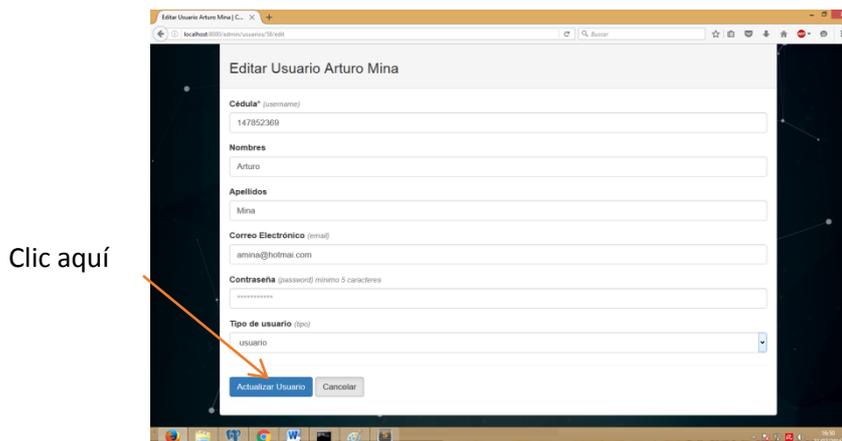


Figura No. 20- Formulario de actualización de usuario

El sistema notificara que él usuario ha sido actualizado exitosamente. (Ver Figura No. 21)

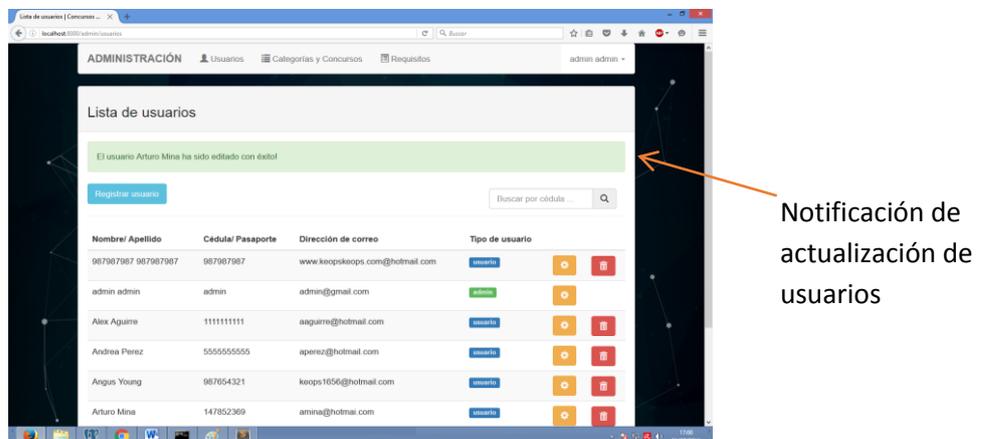
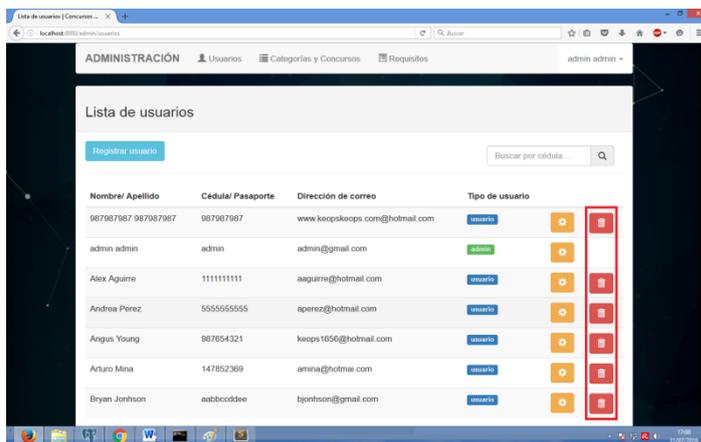


Figura No. 21 - Notificación de actualización de usuario

Eliminar usuario

En el listado de usuarios, cada usuario dispone de un botón rojo con un ícono de un tacho de basura que permite al administrador eliminar a un usuario y toda su información que generó en cada participación de los concursos que ha participado.

(Ver Figura No. 22)

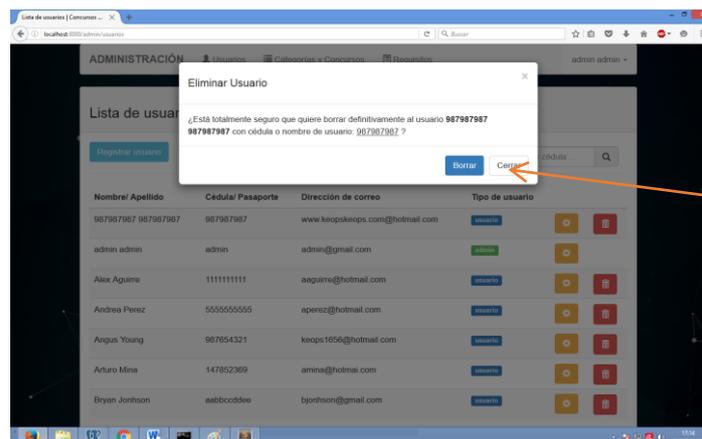


Botones para eliminar usuarios

Figura No. 22 - Botón para eliminar usuario

Presionamos el botón eliminar del usuario que el administrador requiera eliminar y se nos presentara una ventana modal solicitando la confirmación de la eliminación del usuario que se va a eliminar. Hacemos clic en el botón borrar.

(Ver Figura No. 23)



Clic aquí

Figura No. 23- Ventana modal para eliminar usuario

El sistema notificara que él usuario ha sido eliminado exitosamente.

(Ver Figura No. 24)

(Ver Figura No. 24)

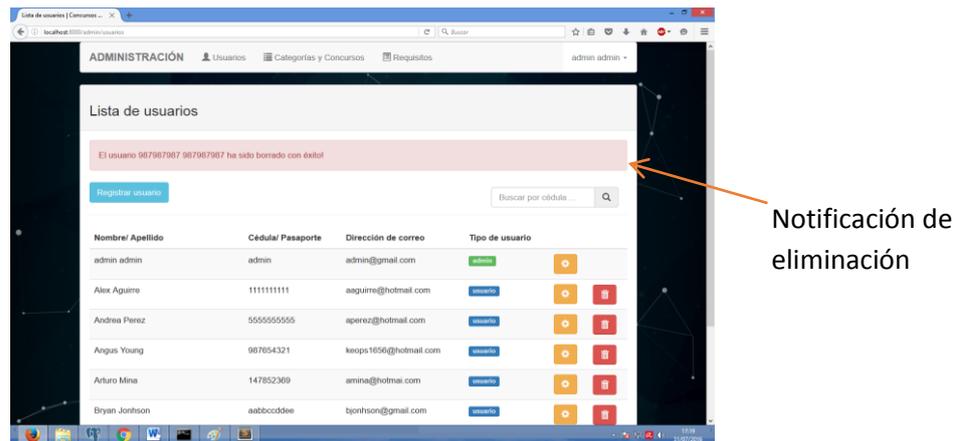


Figura No. 24 - Notificación de eliminación de usuario

Categorías y Concursos

El sistema SIGCOM implementa categorías con el fin de distribuir los concursos y así tener una distribución por año, facultad y escuelas.

Categorías

Al hacer clic en el enlace de Categorías y Concursos. El sistema nos proporciona el listado de todas las categorías existentes con información detallada de cada uno de ellas y varias operaciones que se pueden realizar en ellas. También nos muestra el botón para crear una nueva categoría. (Ver Figura No. 25)

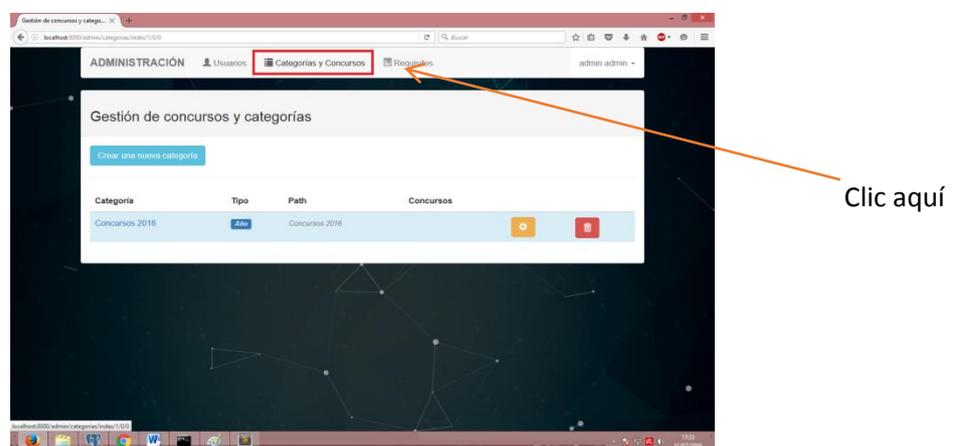


Figura No. 25 - Link de gestión de Categorías y concursos

SIGCOM dispone de 3 tipos de categorías con la siguiente jerarquía:

Año

Facultad

Escuela : Unicamente en esta categoría se pueden crear los concursos.

Crear categoría

Hacemos clic en el botón de crear nueva categoría, el sistema nos presentara un formulario para agregar la nueva categoría. (Ver Figura No. 26)

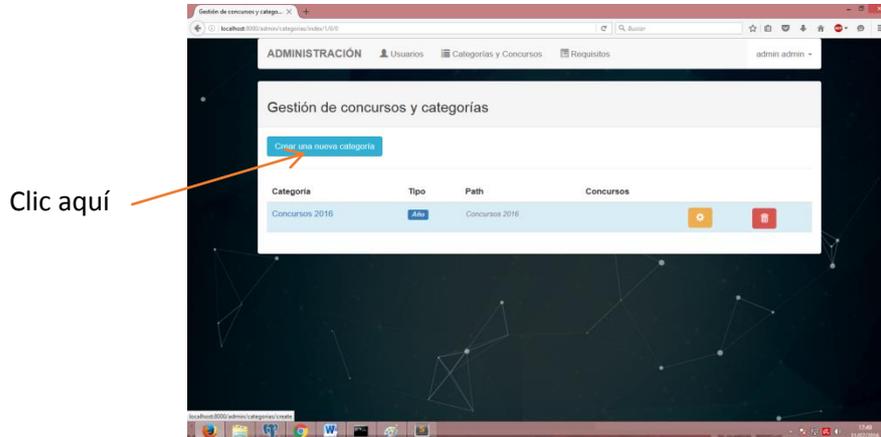


Figura No. 26 - Botón para crear nueva categoría

Seleccionamos la categoría padre que contendrá la categoría que se va crear. Ingresamos el nombre de la categoría y a continuación damos clic en el botón Crear Categoría. (Ver Figura No. 27)

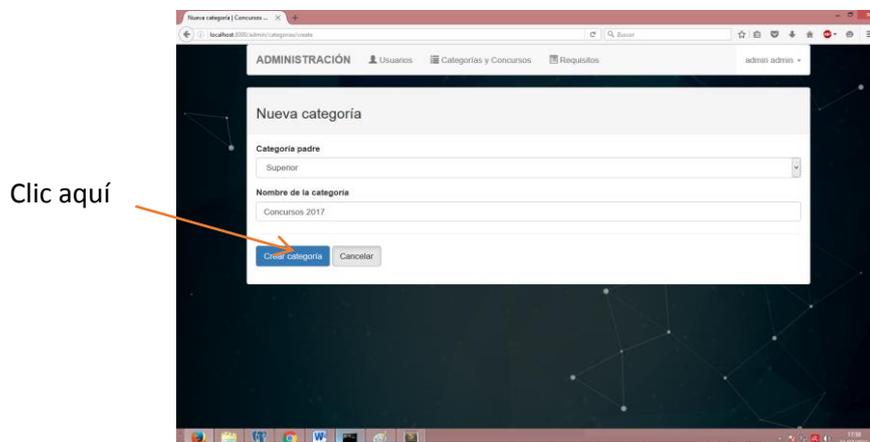
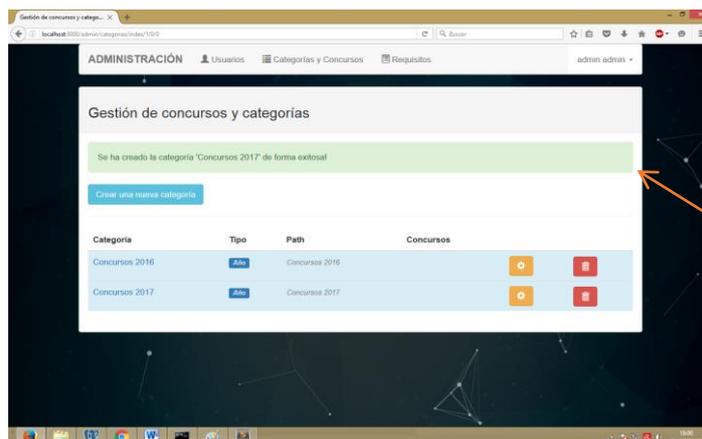


Figura No. 27 - Formulario de creación de categoría

El sistema notificara que la categoría ha sido creada exitosamente. La nueva categoría se mostrará en el listado de categorías. (Ver Figura No. 28)



Notificación de creación de categoría

Figura No. 28 - Notificación de creación de categoría

Navegación entre categorías

Cada una de las categorías de tipo “Año” y “Facultad” son enlaces que permite expandir el contenido de la categoría seleccionada. Las categorías tipo “Año” son de fondo azul, las categorías de tipo “Facultad” son de color amarillo y las categorías de tipo “Escuela” son de color verde.

Solo en las categorías de tipo “Escuela” se pueden crear nuevos concursos. En este caso las categorías de este tipo son un enlace para listar los concursos que tiene esa categoría. (Ver Figura No. 29)

Categoría “Año” o superior

Categoría “Facultad”

Categoría “Escuela”

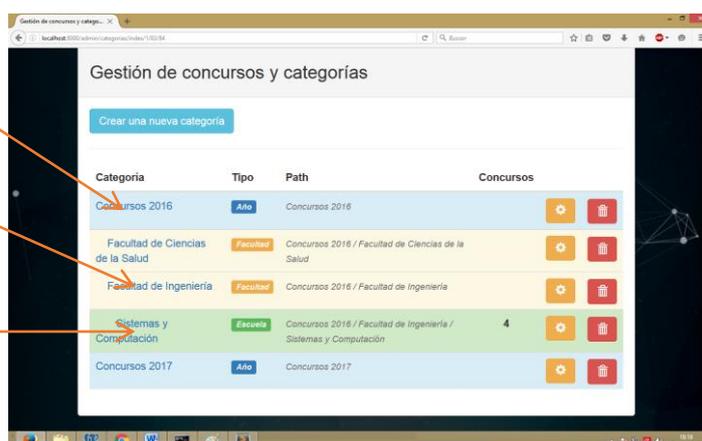


Figura No. 29 - Navegación entre categorías

Actualizar categoría.

En el listado de categorías, cada categoría dispone de un botón naranja con un ícono de un engrane que permite al administrador actualizar la información de una categoría (Ver Figura No. 30)

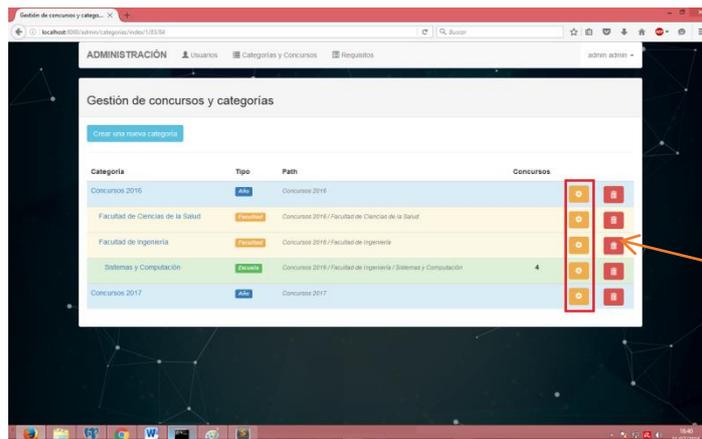


Figura No. 30 - Botones para actualizar categoría

Después de dar clic en el botón de actualizar categoría, se nos presentara un formulario con los campos asignados con la información actual de la categoría. Editamos cada uno de los campos que se desea actualizar y a continuación presionamos el botón Actualizar categoría. (Ver Figura No. 31)

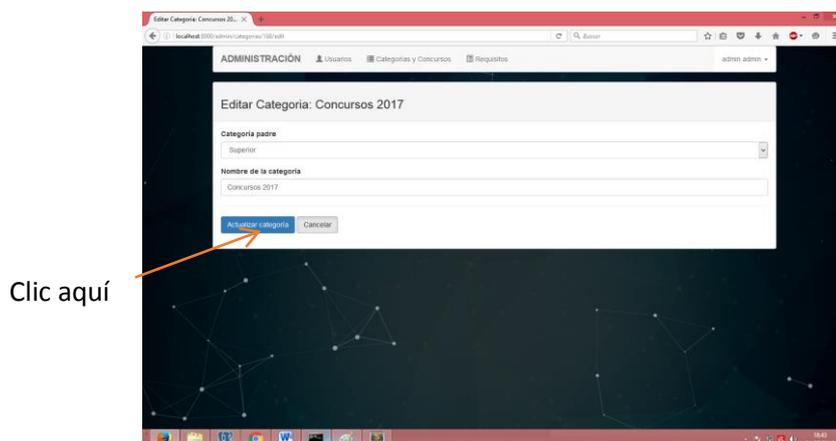


Figura No. 31 - Formulario de actualización de categoría

El sistema notificara que la categoría ha sido actualizada. (Ver Figura No. 32)

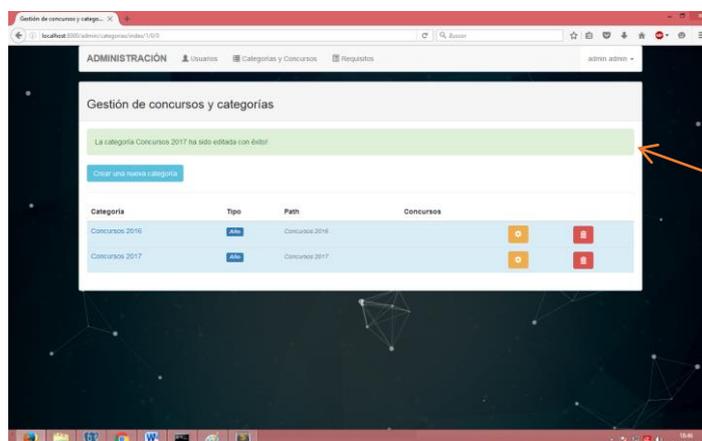


Figura No. 32 - Notificación de actualización de categoría

Eliminar categoría.

Las categorías disponen de un botón rojo para eliminar la categoría seleccionada y toda la información que contiene. (Ver Figura No. 33)

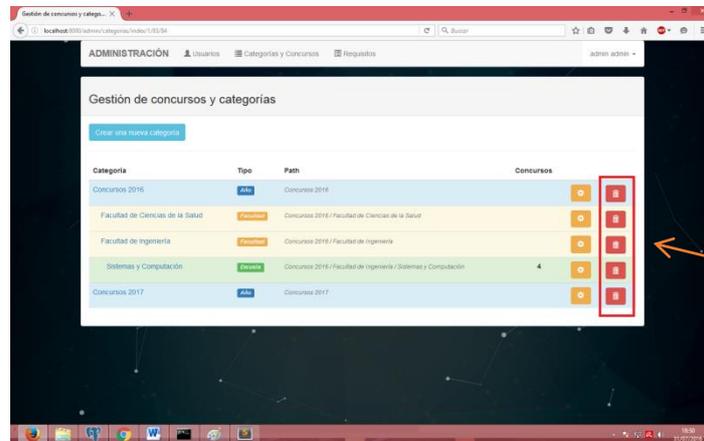


Figura No. 33 - Botón para eliminar categoría

Presionamos el botón de eliminar la categoría que el administrador requiera eliminar y se nos presentara una ventana modal solicitando la confirmación de la eliminación de la categoría que se va a eliminar. Hacemos clic en el botón borrar. (Ver Figura No. 34)

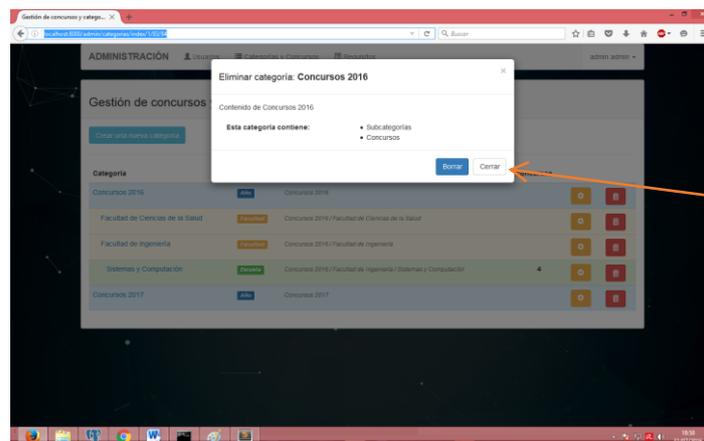
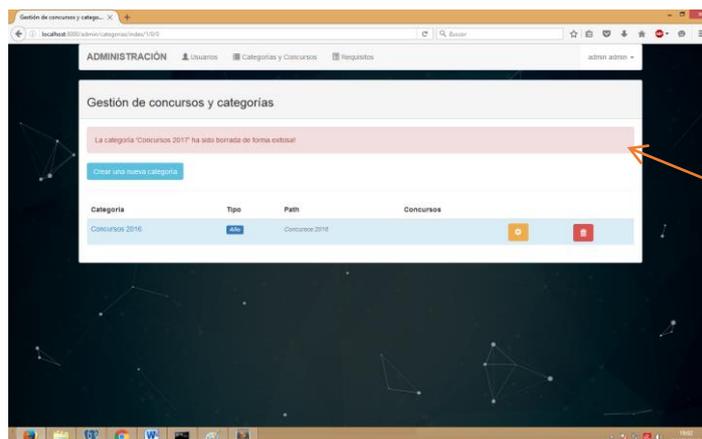


Figura No. 34 - Ventana modal para eliminar categoría

El sistema notificara que la categoría ha sido eliminada exitosamente. (Ver Figura No. 35)



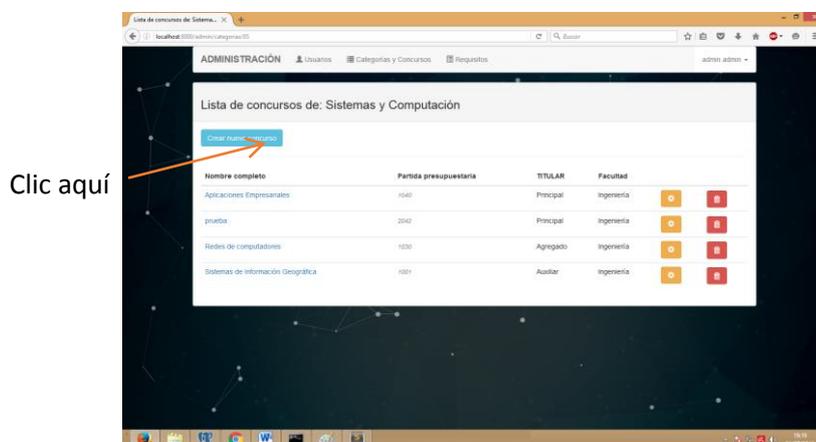
Notificación de eliminación de categoría

Figura No. 35 - Notificación de eliminación de categoría

Concursos

Cada concurso se encuentra alojado en una categoría tipo “Escuela”. Para obtener el listado de los concursos que tiene una categoría se debe navegar por cada categoría hasta entrar a una categoría “Escuela”. **(Ver Figura No. 29)**

Al seleccionar una categoría tipo “Escuela”, el sistema nos proporciona el listado de todos los concursos existentes en dicha seleccionada con información detallada de cada uno de los concursos y varias operaciones que se pueden realizar en ellos. También nos muestra el botón para crear un nuevo concurso. **(Ver Figura No. 36)**



Clic aquí

Figura No. 36 - Botón para crear concursos

Crear concurso

Hacemos clic en el botón de crear nuevo concurso, el sistema nos presentara un formulario para agregar un concurso en la categoría seleccionada. **(Ver Figura No. 37)**

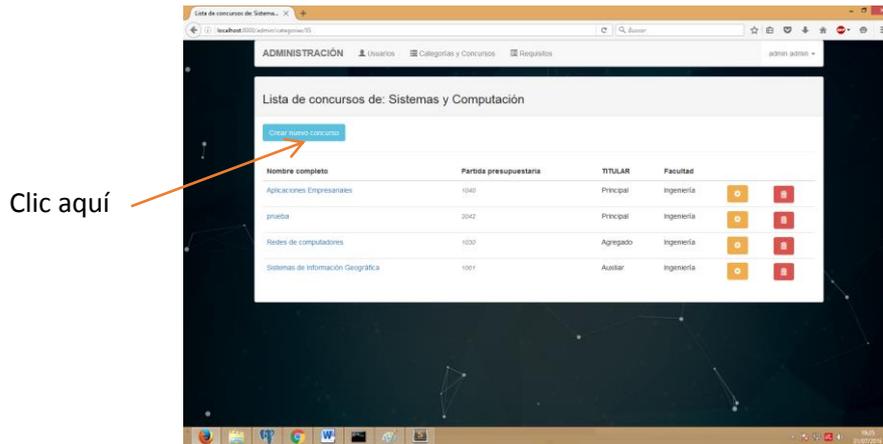


Figura No. 37 - Botón para crear nuevo concurso

Seleccionamos la titularidad del concurso y llenamos los demás requerimientos. (Ver **Figura No. 38**)

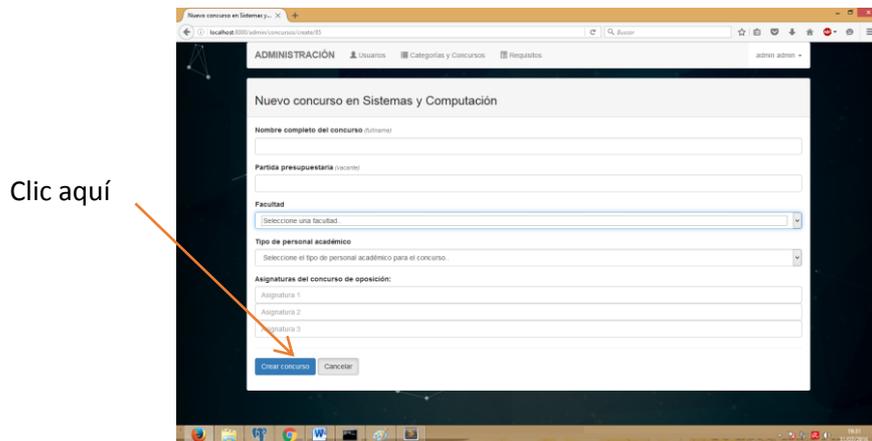


Figura No. 38 - Formulario de creación de un concurso

El sistema notificará que el concurso ha sido creado. El nuevo concurso se mostrará en el listado de la categoría seleccionada. (Ver **Figura No. 39**)

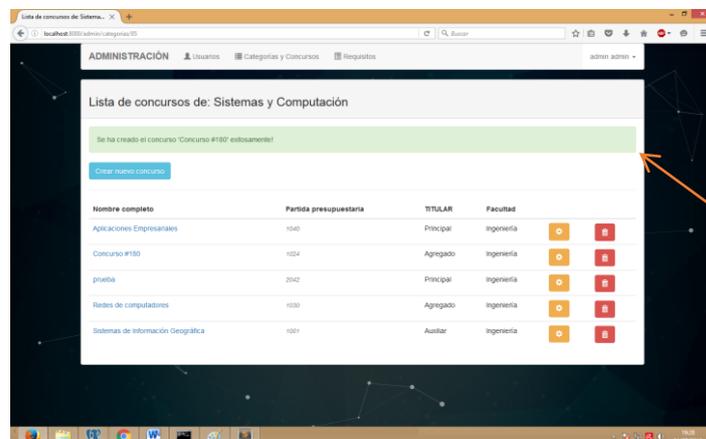
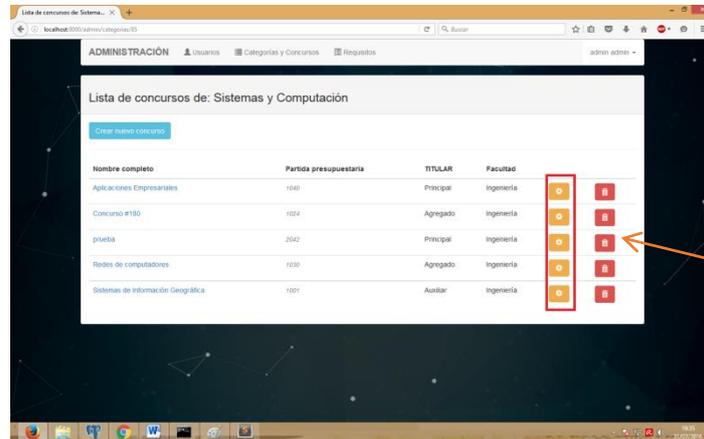


Figura No. 39 - Notificación de creación de concurso

Actualizar concurso.

Cada concurso dispone de un botón naranja con un ícono de un engrane que permite al administrador actualizar la información del concurso. (Ver Figura No.

40)



Botones para actualizar concursos

Figura No. 40 - Botones para actualizar concurso

Después de dar clic en el botón de actualizar concurso, se nos presentara un formulario con los campos asignados con la información actual del concurso. Editamos cada uno de los campos que se desea actualizar y a continuación presionamos el botón Actualizar concursos. (Ver Figura No. 41)

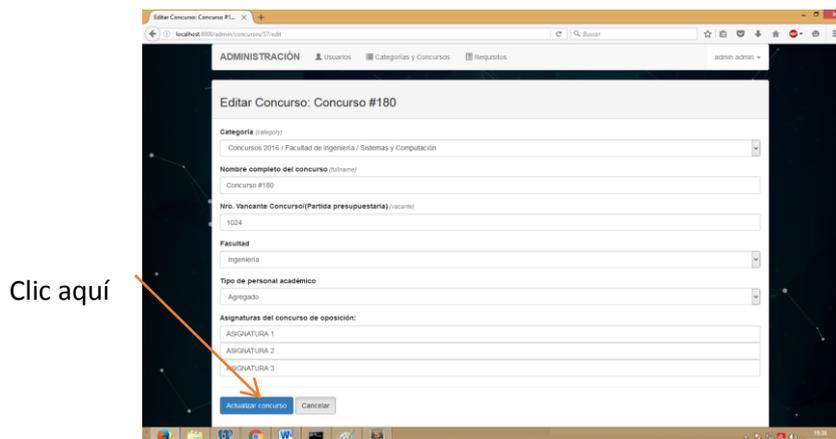
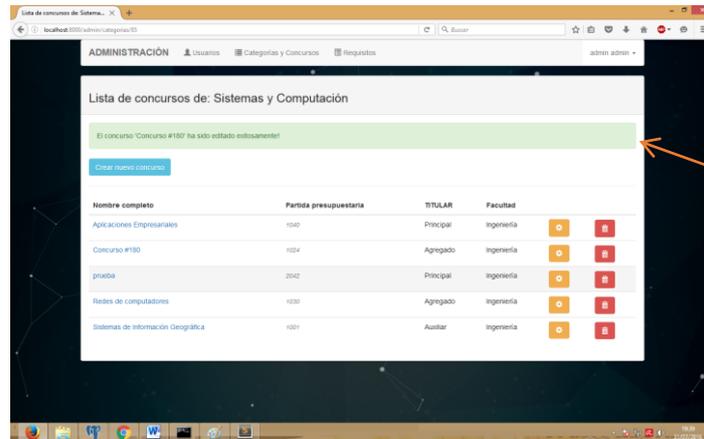


Figura No. 41 - Formulario de actualización de concurso

El sistema notificara que el concurso ha sido actualizado. (Ver Figura No. 42)

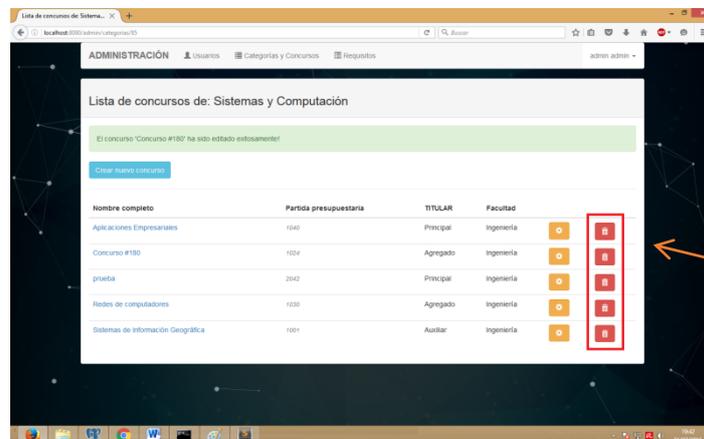


Notificación de actualización del concurso

Figura No. 42 - Notificación de actualización del concurso

Eliminar concurso.

Los concursos disponen de un botón rojo para eliminarlo con toda la información que contiene. (Ver Figura No. 43)



Botones para eliminar concursos

Figura No. 43 - Botón para eliminar concurso

Presionamos el botón de eliminar concurso y el sistema nos presentara una ventana modal solicitando la confirmación de la eliminación del concurso que se va a eliminar. Hacemos clic en el botón borrar. (Ver Figura No. 44)

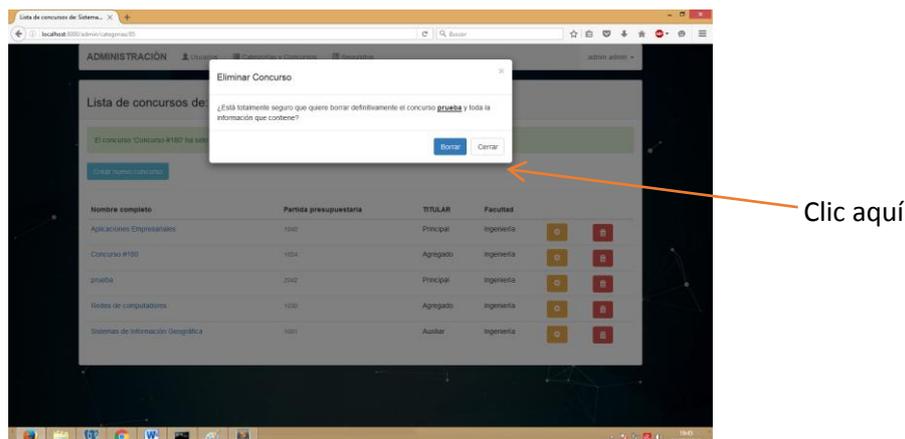


Figura No. 42 - Ventana modal para eliminar concurso

El sistema notificara que la categoría ha sido eliminada exitosamente. (Ver Figura No. 45)

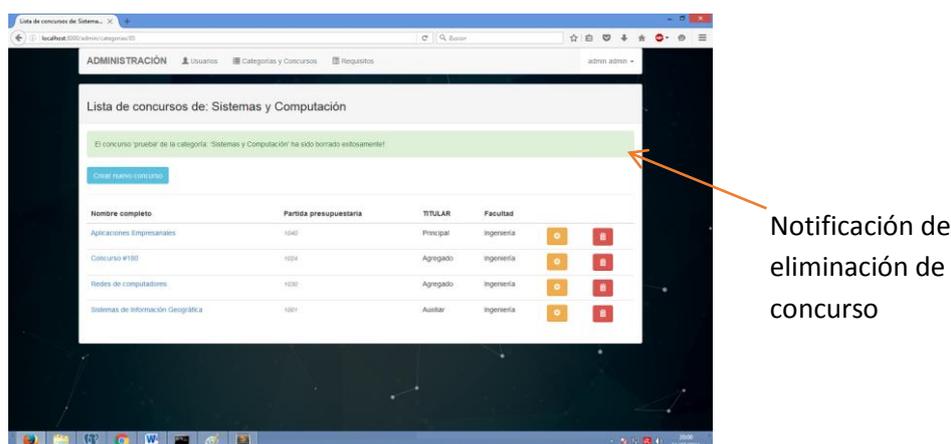


Figura No. 43 - Notificación de eliminación de concurso

Matrículas

Matriculación es el proceso de marcar a los usuarios como participantes en un concurso, que incluye tanto a postulante como a evaluadores. Al mismo tiempo se les da a los usuarios un rol en el curso que especifica lo que se les permite hacer.

SIGCOM dispone de 3 tipos de rol para una matrícula que se detallan a continuación:

POSTULANTE: Los postulantes pueden subir archivos, consultar información de cada una de las fases.

EVALUADOR: Pueden calificar a los postulantes en la fase de oposición, obtener informes.

EVALUADOR (PRESIDENTE): Pueden calificar a los postulantes en la fase de méritos y oposición, obtener informes, notificar resultados, etc.

Matricular usuario.

Ingresamos al concurso que se requiere gestionar y se nos presenta la pantalla principal del concurso (Ver Figura No. 44)

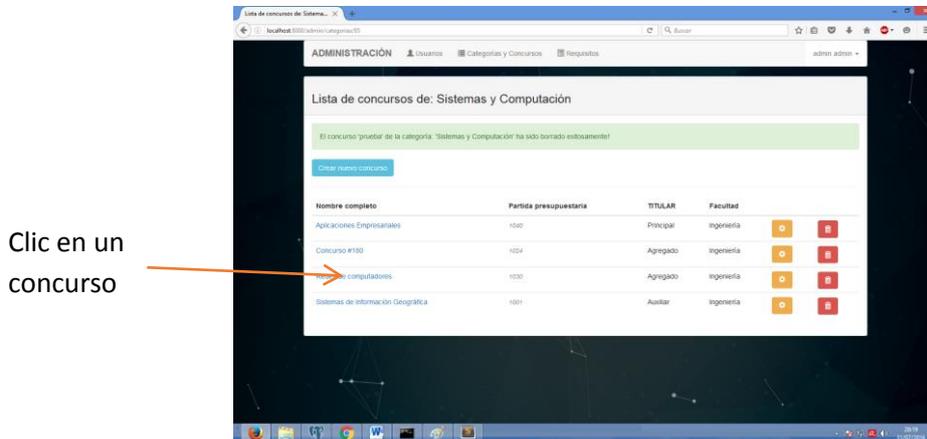


Figura No. 44 - Selección de un concurso

Hacemos clic en la opción de “Matricular usuarios” y se nos presenta dos paneles. (Ver Figura No. 45)

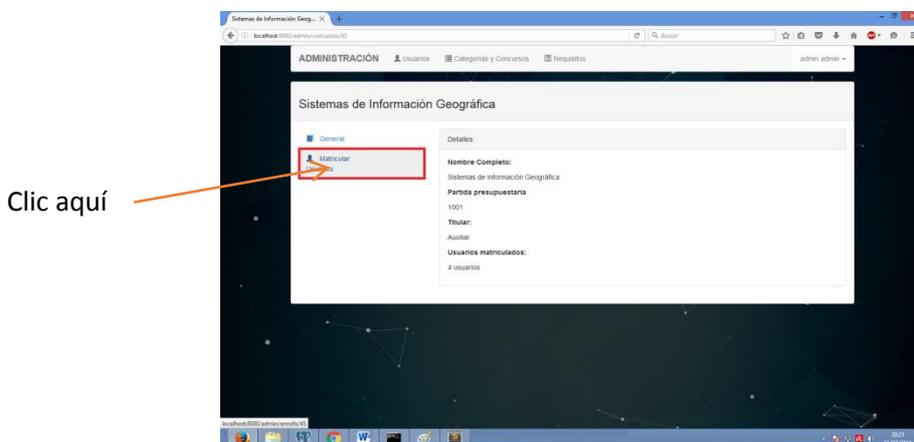


Figura No. 45 - Enlace para matricular usuarios

Se nos presenta dos paneles, el panel superior contiene un formulario para matricular a un usuario y el panel inferior es un listado de los usuarios matriculados en ese concurso. (Ver Figura No. 46)

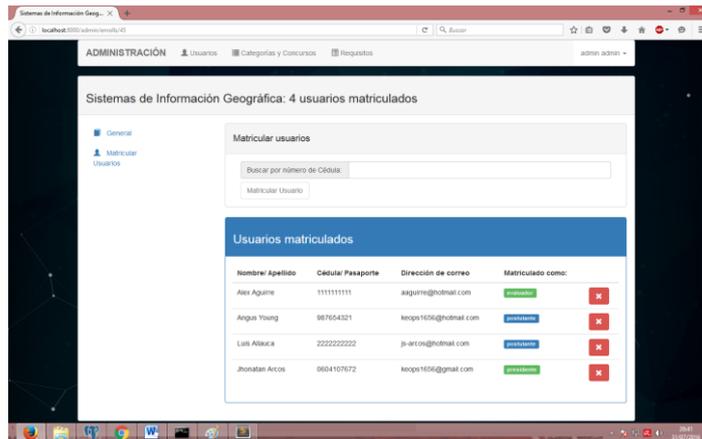


Figura No. 46 - Pantalla principal para la gestión de matrículas

Ingresamos el número de cédula del usuario que se desea matricular. El sistema muestra el resultado de la búsqueda. Si el usuario es encontrado y si aún no se encuentra en ese concurso, el sistema genera un formulario para asignar el rol que tendrá el usuario en el concurso y se habilita el botón “Maticular”. Hacemos clic en el botón “Maticular”. (Ver Figura No. 47)

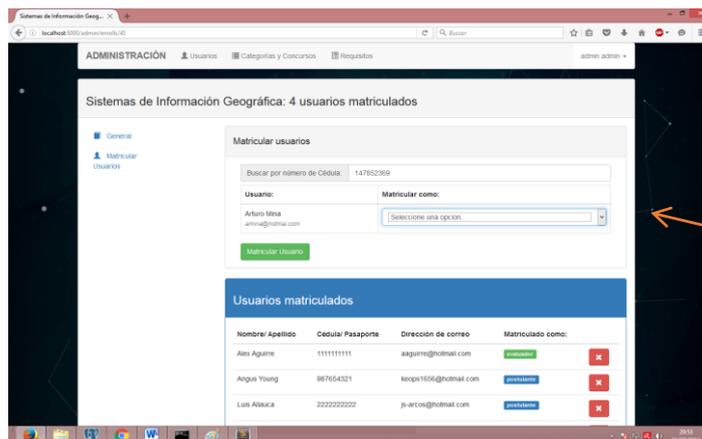


Figura No. 47 - Formulario para matricular usuarios

El sistema notifica que el usuario ha sido matriculado”. (Ver Figura No. 47)

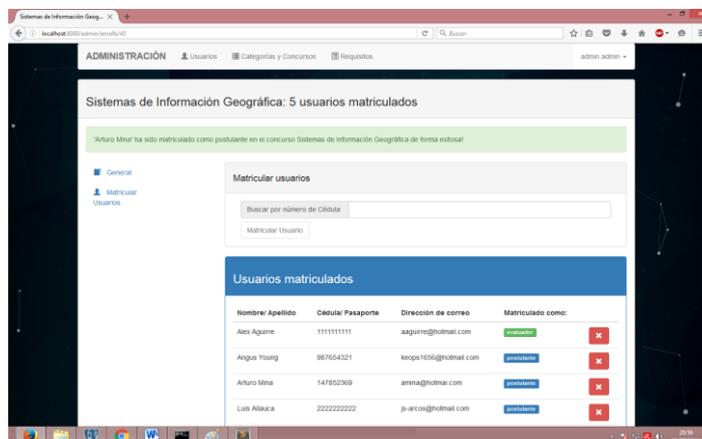


Figura No. 48 - Notificación de matrícula

Desmatricular usuario.

Ingresamos al concurso que se requiere gestionar y se nos presenta la pantalla principal del concurso Pantalla principal para la gestión de matrículas. Hacemos clic en la opción de “Matricular usuarios” y se nos presenta la pantalla principal para la gestión de matrículas. (Ver Figura No. 46)

En el listado de usuarios matriculados se dispone de un botón rojo cada uno de los participantes para desmatricularlo. (Ver Figura No. 49)

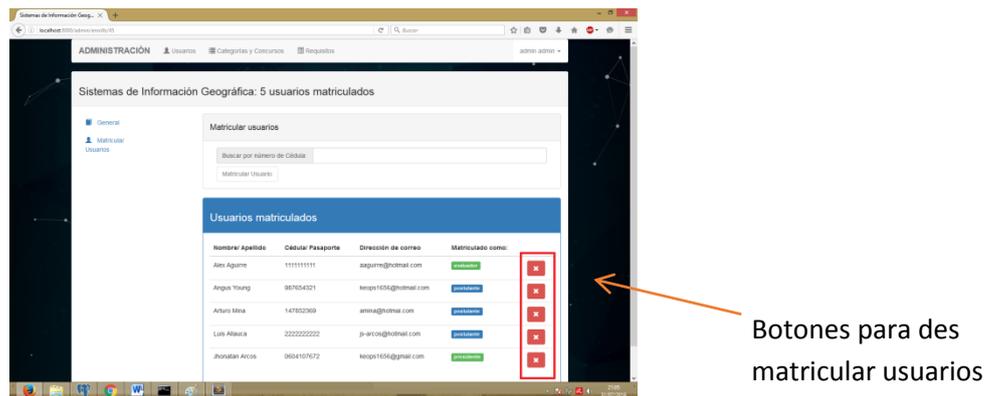


Figura No. 49 - Botones para desmatricular usuarios

Presionamos el botón de des matricular y el sistema nos presentara una ventana modal solicitando la confirmación de la des matriculación del usuario seleccionado en un concurso. Hacemos clic en el botón Sí. (Ver Figura No. 50)

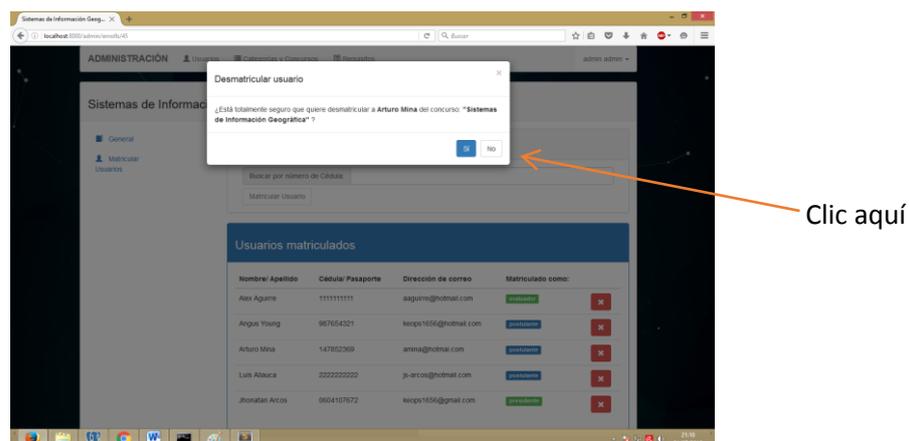


Figura No. 50 - Ventana modal para desmatricular usuario

El sistema notificara que se ha desmatriculado al usuario del concurso. (Ver Figura No. 51)

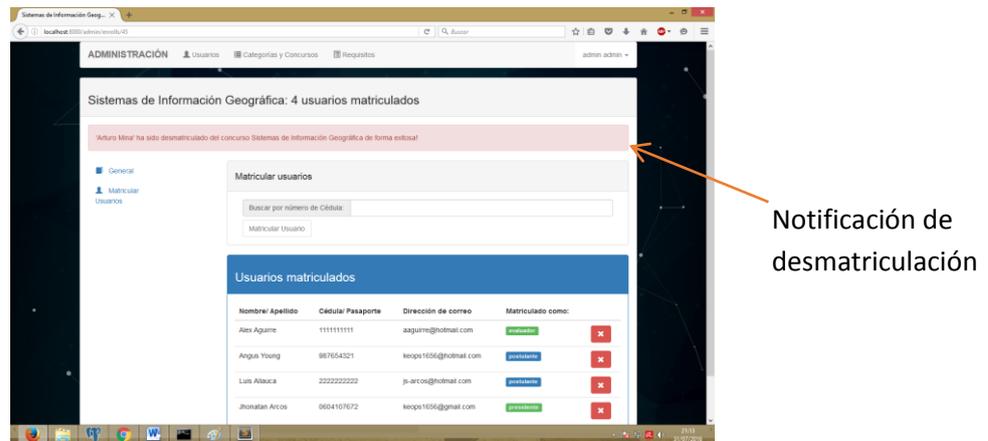


Figura No. 51 - Notificación de desmatriculación de usuario

Gestión de Requisitos Generales de postulación

Al hacer clic en el enlace de Requisitos. El sistema nos proporciona el listado de todos los requisitos existentes con información detallada de cada uno de ellos y varias operaciones que se pueden realizar en ellos. También nos muestra el botón para ingresar un nuevo requisito. (Ver Figura No. 52)

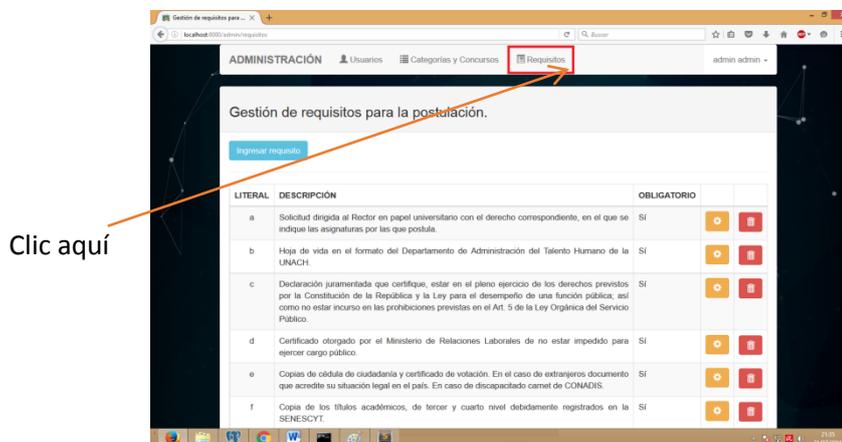


Figura No. 52- Link de Gestión de requisitos

Registro de requisitos.

Hacemos clic en el botón de Ingresar requisito, el sistema nos presentara un formulario para agregar un nuevo requisito. (Ver Figura No. 53)

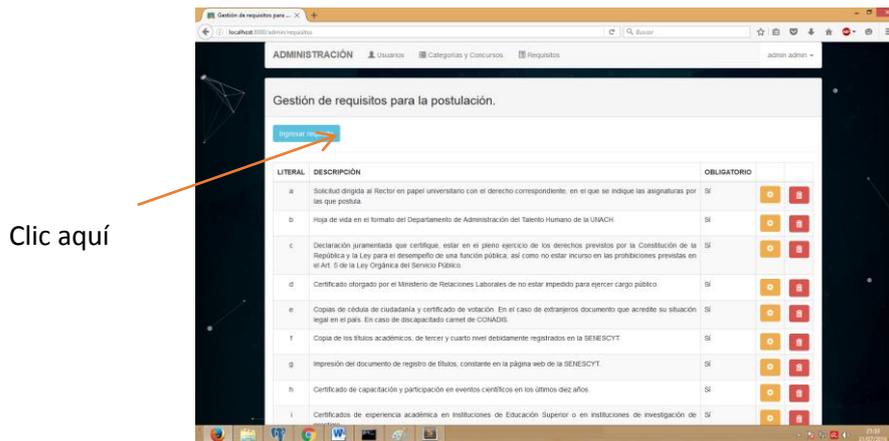


Figura No. 52 - Botón Ingresar requisito

Completamos el formulario con información del requisito, a continuación hacemos clic en el botón Crear requisito. (Ver Figura No. 53)

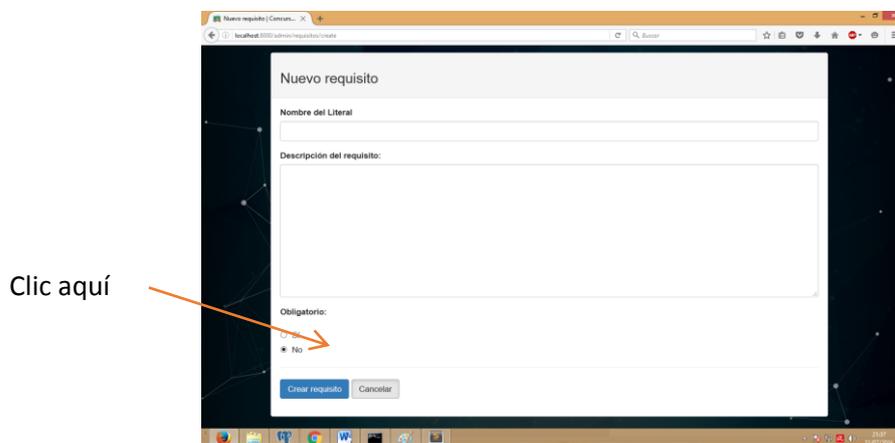


Figura No. 53 - Formulario de registro de requisitos

El sistema notificará que el requisito ha sido creado exitosamente. El nuevo requisito aparecerá en el listado de requisitos. (Ver Figura No. 55)

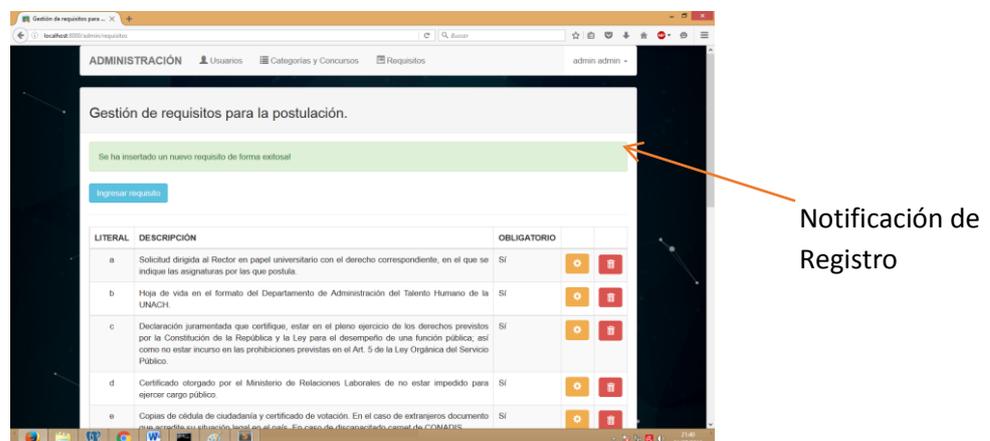


Figura No.55 - Notificación de registro de requisito

Actualizar requisito

Cada requisito dispone de un botón naranja con un ícono de un engrane que permite al administrador actualizar un requisito. (Ver Figura No. 56)

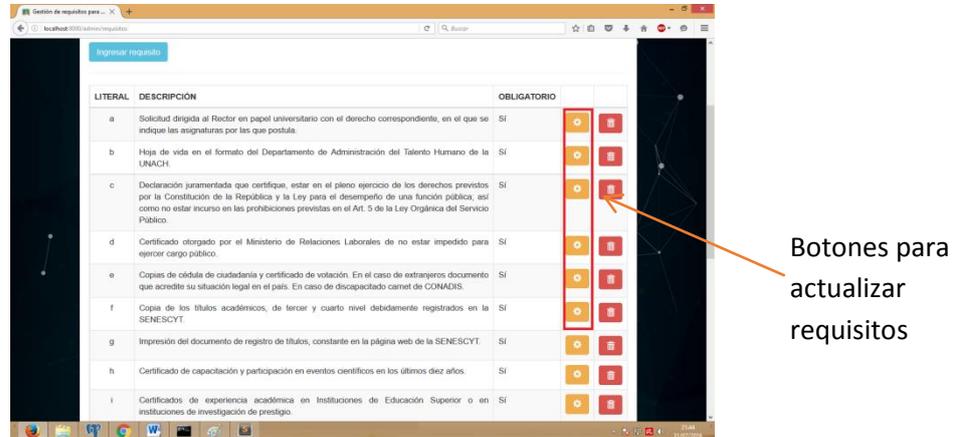


Figura No. 53 - Botón para actualizar requisito

Después de dar clic en el botón de actualizar requisito, se nos presentara un formulario con los campos asignados con la información actual del requisito. Editamos cada uno de los campos que se desea actualizar y a continuación presionamos el botón Actualizar requisito. (Ver Figura No. 57)

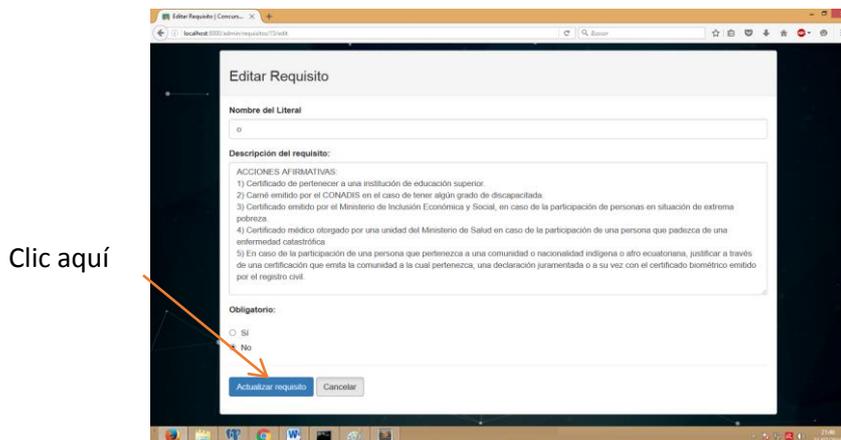


Figura No. 57- Formulario de actualización de requisito

El sistema notificara que el usuario ha sido actualizado exitosamente. (Ver Figura No. 58)



Notificación de actualización de requisito

Figura No. 5854 - Notificación de actualización de requisito

Eliminar requisito

En el listado de requisitos, cada uno dispone de un botón rojo con un ícono de un tachó de basura que permite al administrador eliminar a un requisito. (Ver Figura No. 59)



Botones para eliminar requisitos

Figura No. 59 - Botón para eliminar requisito

Presionamos el botón eliminar del requisito que el administrador requiera eliminar y se nos presentara una ventana modal solicitando la confirmación de la eliminación del requisito que se va a eliminar. Hacemos clic en el botón borrar. (Ver Figura No. 60)

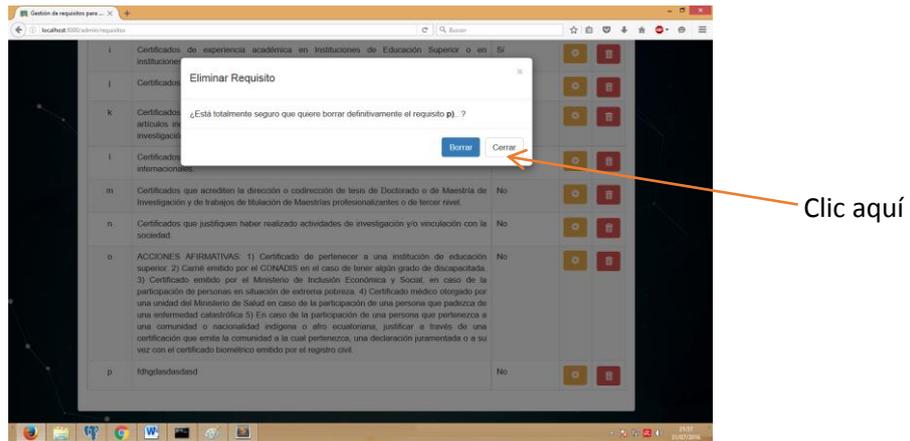


Figura No. 60- Ventana modal para eliminar requisito

El sistema notificara que él requerimiento ha sido eliminado exitosamente.(Ver **Figura No. 61)**

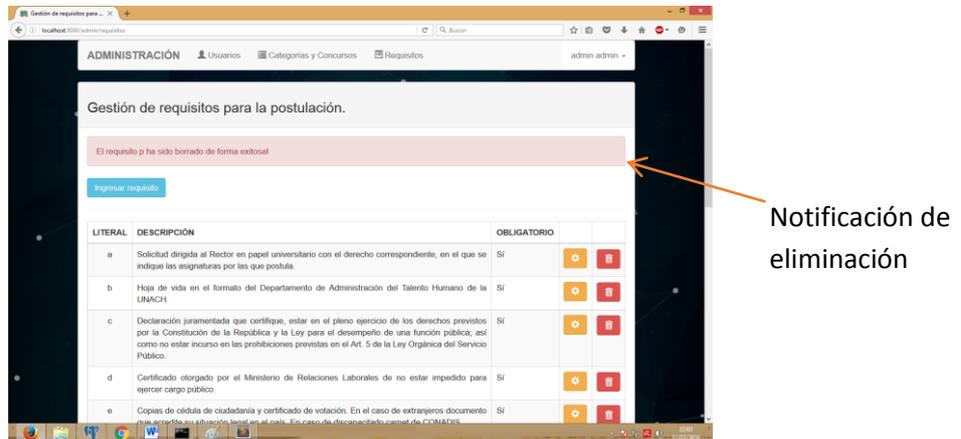


Figura No. 6155 - Notificación de eliminación de requisito