



**FACULTAD DE INGENIERÍA, ARQUITECTURA  
Y URBANISMO**

**ESCUELA ACADÉMICO PROFESIONAL DE  
INGENIERÍA DE SISTEMAS**

**TESIS**

**ANÁLISIS COMPARATIVO DE FRAMEWORKS PHP  
PARA MEDIR EL RENDIMIENTO.**

**PARA OPTAR EL TÍTULO PROFESIONAL DE  
INGENIERO DE SISTEMAS**

**Autores:**

**Bach. Llatas Yovera, Luis Humberto**

**Bach. Rojas Herrera, Alex Jhampier**

**Asesor:**

**Mg.Samillan Ayala, Alberto Enrique**

**Línea de Investigación:**

**Tecnologías de la Información**

**Pimentel – Perú**

**2017**

## ANÁLISIS COMPARATIVO DE FRAMEWORKS PHP PARA MEDIR EL RENDIMIENTO.

Aprobación de la Tesis

---

**Bach. Llatas Yovera Luis Humberto**  
Autor

---

**Bach. Rojas Herrera Alex Jhampier**  
Autor

---

**Mg. Samillan Ayala Alberto Enrique**  
Asesor

---

**Ing. Ocampo Merino Rosa Liliana**  
Presidente del Jurado

---

**Mg. Tuesta Monteza Víctor Alexci**

**Secretario del Jurado**

---

**Mg. Samillan Ayala Alberto Enrique**

**Vocal del Jurado**

## DEDICATORIA

Dirigido a nuestros padres y a los docentes que nos guiaron.

**Alex Jhampier-Luis Llatas**

## AGRADECIMIENTO

Agradecemos a Dios, a nuestros padres, compañeros de aula y a esta prestigiosa universidad por permitirnos alcanzar nuestras metas profesionales.

**Alex Jhampier-Luis Llatas**

## CONTENIDO

<b>CAPITULO I: PROBLEMA DE INVESTIGACION .....</b>	<b>16</b>
1.1 Situación Problemática .....	16
1.2 Formulación del problema.....	17
1.3 Delimitación de la Investigación.....	17
1.4 Justificación e Importancia.....	17
1.5 Limitación de la Investigación .....	18
1.6 Objetivos .....	19
1.6.1 Objetivo general .....	19
1.6.2 Objetivos específicos.....	19
<b>CAPÍTULO II: MARCO TEÓRICO .....</b>	<b>20</b>
2.1 Antecedentes de la Investigación .....	20
2.2 Estado del arte.....	23
2.3 Base Teórica Científica .....	26
2.3.1 Framework web .....	26
2.3.2 Lenguaje de Programación PHP .....	28
2.3.3 Frameworks de PHP .....	31
2.3.3.1 FRAMEWORK YII .....	31
2.3.3.2 FRAMEWORK LARAVEL .....	33
2.3.4 Modelo Vista Controlador (MVC).....	34
2.3.6 Base de Datos MySQL .....	37
2.3.7 Rendimiento .....	39
2.4 Definición de Términos Básicos.....	40
<b>CAPÍTULO III: MARCO METODOLÓGICO .....</b>	<b>41</b>
3.1 Tipo y diseño de investigación .....	41
3.1.1 Tipo de Investigación.....	41
3.1.2 Diseño de Investigación.....	41
3.2 Población y muestra.....	41
3.2.1 Población.....	41
3.2.2 Muestra.....	41
3.3 Hipótesis .....	41
3.10 Principios de rigor científico .....	41
3.4 Variables .....	42



3.4.1 Independiente.....	42
3.4.2 Dependiente .....	42
3.5 Operacionalización.....	43
3.6 Abordaje metodológico, técnicas e instrumentos de recolección de datos .....	46
3.6.1 Abordaje Metodológico .....	46
3.6.2 Método y técnica de recolección de datos.....	46
3.6.3 Instrumentos de recolección de datos .....	46
3.7 Procedimiento para la recolección de datos .....	46
3.8 Análisis estadístico de datos .....	47
3.9 Principios éticos .....	48
<b>CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS.....</b>	<b>49</b>
4.1. Resultados en tablas y gráficos .....	49
<b>CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN .....</b>	<b>52</b>
5.1. Analizar los framework PHP para su Selección .....	52
5.2 Seleccionar los indicadores de Rendimiento para la comparación entre frameworks .....	65
5.3. Desarrollar un sistema con cada framework seleccionado .....	66
5.3.1 Metodología SCRUM.....	67
5.3.1.1 Descripción del Sistema Propuesto.....	67
5.3.1.2 Personas y Roles del Proyecto .....	67
5.3.1.3 Historias de Usuario .....	68
5.3.1.4 Product Backlog .....	69
5.3.1.5 Planificación SCRUM .....	70
5.3.1.6 Reuniones SCRUM .....	70
5.3.1.7 Desarrollo de Sprint del Proyecto.....	70
5.3.1.8 Sprints Backlog .....	71
5.4 Evaluar los framework en base a los indicadores de rendimiento seleccionados. ....	110
5.4.1 PLAN DE PRUEBAS PARA MEDIR EL RENDIMIENTO.....	110
5.4.1.1 Hoja resumen de Proyecto y de modificaciones.....	110
5.4.1.2 Objetivo del Plan de Pruebas .....	111
5.4.1.3 Alcance de Las Pruebas .....	111
5.4.1.4 Tipo de Requerimiento a medir .....	113
5.4.1.5 Criterios de Aprobación / Rechazo.....	113
5.4.1.6 Casos de Pruebas.....	114



5.4.1.7 Casos De Pruebas .....	116
5.4.1.8 Realizar las conclusiones obtenidas y recomendaciones .....	125
5.4.1 Evaluación de Framework Laravel en Tiempo de Respuesta.....	126
5.4.2 Evaluación de Framework Laravel en Consumo de Memoria RAM	134
5.4.3 Evaluación de Framework Laravel en Tiempo de Carga .....	142
5.4.4 Evaluación de Framework Yii en Tiempo de Respuesta .....	149
5.4.5 Evaluación de Framework Yii en Consumo de Memoria RAM .....	157
5.4.6 Evaluación de Framework Yii en Tiempo de Carga.....	165
5.5 Identificar según cada indicador medido, que framework resulto mejor..	173
5.5.1 Indicador Tiempo de Respuesta.....	173
5.5.2 Indicador Consumo de Memoria RAM.....	175
5.5.3 Indicador Tiempo de Carga .....	177
<b>CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>178</b>
<b>Referencias.....</b>	<b>180</b>
<b>Anexos .....</b>	<b>183</b>



## INDICE DE FIGURAS

Figura 1. Uso de lenguajes de programación.....	24
Figura 2. Uso de lenguajes de programación del lado del servidor para sitios web. .....	29
Figura 3. Lenguajes de Programación de Lado del Servidor .....	29
Figura 4. Funcionamiento del Lenguaje PHP .....	31
Figura 5. Funcionamiento del patrón Modelo -Vista-Controlador .....	35
Figura 6. Archivos para implementar Modelo, Vista y Controlador en PHP .....	36
Figura 7. Esquema del uso de MVC en PHP .....	37
Figura 8. Arquitectura de integración con un SGBDR .....	38
Figura 9. Comparación de Tiempo de Consulta .....	65
Figura 10. Consumo de memoria del marco .....	66
Figura 11. Número de archivos cargados .....	66
Figura 12. Modelo Conceptual de la Base de Datos .....	74
Figura 13. Modelo Físico en MySQL .....	75
Figura 14. Pantalla de Logueo .....	84
Figura 15. Pantalla Categorías.....	84
Figura 16. Pantalla Agregar una Nueva Categoría.....	85
Figura 17. Pantalla Editar Categorías .....	85
Figura 18. Pantalla Eliminar Categorías.....	86
Figura 19. Pantalla Marcas .....	87
Figura 20. Pantalla Listar Marca.....	87
Figura 21. Pantalla Productos .....	88
Figura 22. Pantalla Agregar Producto .....	88
Figura 23. Pantalla Clientes y Proveedores .....	89
Figura 24. Pantalla Listar Clientes.....	89
Figura 25. Pantalla Agregar Proveedor .....	90
Figura 26. Pantalla Trabajadores .....	90
Figura 27. Pantalla Agregar Nuevo Trabajador .....	91
Figura 28. Pantalla Editar Trabajador.....	91
Figura 29. Pantalla Ingreso-Compras.....	92
Figura 30. Pantalla Agregar Ingreso-Compra.....	92
Figura 31. Pantalla Listado de Ingresos .....	93
Figura 32. Pantalla Ingreso-Producto.....	93
Figura 33. Pantalla de Productos .....	94
Figura 34. Pantalla Ventas .....	94
Figura 35. Pantalla Agregar una Venta .....	95
Figura 36. Pantalla Confirmar Venta .....	95
Figura 37. Pantalla Listado de Ventas.....	96
Figura 38. Pantalla de Logueo .....	97
Figura 39. Pantalla Categorías.....	98
Figura 40. Pantalla Agregar Nueva Categoría.....	98
Figura 41. Pantalla Editar Categorías .....	99
Figura 42. Pantalla de Eliminar Categoría.....	99
Figura 43. Pantalla Marcas .....	100
Figura 44. Pantalla Listar Marcas.....	100





Figura 45.Pantalla Productos .....	101
Figura 46.Pantalla Agregar Nuevo Producto .....	101
Figura 47.Pantalla Persona .....	102
Figura 48.Pantallazo Proveedor .....	102
Figura 49.Pantallazo Editar Trabajador .....	103
Figura 50.Pantalla Lista de Ingresos .....	103
Figura 51.Pantalla Agregar Nuevo Ingreso .....	104
Figura 52.Pantalla Productos .....	104
Figura 53.Pantalla Ventas .....	105
Figura 54.Pantalla Listar Venta .....	105
Figura 55.Pantalla Agregar Nuevo Trabajador .....	106
Figura 56.Apache en PC Local.....	126
Figura 57.Peticiones al Servidor.....	128
Figura 58.Transcurriendo las Peticiones .....	128
Figura 59.Resultados Obtenidos con 1000 peticiones .....	129
Figura 60. Cantidad y Tiempo de Peticiones.....	131
Figura 61.Insertando Codigo de Prueba en Tabla Producto. ....	135
Figura 62 Prueba de Carga a la tabla Producto .....	145
Figura 63.Prueba de Carga a la tabla Persona .....	146
Figura 64.Prueba de Carga a la tabla Venta .....	147
Figura 65.Ruta de Apache.....	150
Figura 66.URL a Testear .....	151
Figura 67.Prueba con Peticiones al Servidor .....	151
Figura 68.Prueba 1000 Peticiones .....	152
Figura 69.Tiempo de Respuesta y Peticiones .....	154
Figura 70.Memoria RAM de tabla Categoría .....	158
Figura 71.Prueba de memoria RAM a la tabla Marca .....	159
Figura 72.Prueba de memoria RAM a la tabla Producto .....	160
Figura 73.Prueba de memoria RAM a la tabla Persona .....	161
Figura 74.Prueba de memoria RAM a la tabla Venta .....	162
Figura 75.Prueba de Carga a la tabla Categoría.....	166
Figura 76.Prueba de Carga a la tabla Marca.....	167
Figura 77.Prueba de Carga a la tabla Producto .....	168
Figura 78.Prueba de Carga a la tabla Persona .....	169
Figura 79.Prueba de Carga a la tabla Venta .....	170



## INDICE DE GRAFICAS

Gráfico 1.PHP usado en los sitios Web.....	22
Gráfico 2. Resultados de la encuesta de framework PHP.....	23
Gráfico 3.Tiempo que tarda que tarda cada versión de PHP .....	39
Gráfico 4.Comparacion de Ambos Framework Laravel Vs Yii.....	49
Gráfico 5.Comparación de ambos Framework Laravel Vs Yii .....	50
Gráfico 6.Diagrama de Flujo que muestra como es el Proceso de la Propuesta Planteado .....	52
Gráfico 7.Promedio de Tiempo de Respuesta cada 1000 peticiones.....	133
Gráfico 8. Promedio en Tiempo de Respuesta en Milisegundos.....	133
Gráfico 9.Promedio de Tiempo de Respuesta en Segundos .....	134
Gráfico 10.Total de Memoria RAM en tablas hechas con Laravel.....	140
Gráfico 11.Promedio de Memoria RAM total consumida por el Framework Laravel (bytes) .....	141
Gráfico 12.Promedio de Memoria RAM en Laravel .....	141
Gráfico 13.Total de Tiempo de Carga en tablas hechas con Laravel.....	148
Gráfico 14.Promedio de Tiempo de Carga total consumida por el Framework Laravel (microsegundos).....	149
Gráfico 15.Promedio de Tiempo de Respuesta cada 100 peticiones.....	156
Gráfico 16.Promedio de Tiempo de Respuesta.....	156
Gráfico 17.Promedio Uso de Memoria RAM en tablas hechas con Yii.....	163
Gráfico 18.Promedio Memoria RAM.....	164
Gráfico 19.Promedio Memoria RAM.....	164
Gráfico 20.Total de Tiempo de Carga en tablas hechas con Yii.....	171
Gráfico 21.Promedio de Tiempo de Carga total consumida por el Framework Yii (microsegundos).....	172
Gráfico 22.Resultado en Tiempo de Respuesta en milisegundos .....	173
Gráfico 23.Resultado en Tiempo de Respuesta en segundos .....	174
Gráfico 24.Resultado en Consumo de Memoria RAM en bytes .....	175
Gráfico 25.Resultado en Consumo de Memoria RAM en megabytes .....	176



## INDICES DE TABLAS

Tabla 1.Tabla de Operacionalización .....	43
Tabla 2.Tabla de Niveles con Puntaje.....	62
Tabla 3.Puntaje a cada Framework según los criterios establecidos .....	63
Tabla 4.Personas y Roles Scrum .....	67
Tabla 5.Requerimientos del Software utilizado para la Programación .....	72
Tabla 6.Hardware utilizado para la Programación.....	73
Tabla 8.Cantidad de Peticiones y Tiempo de Respuesta .....	131
Tabla 9.Uso de Memoria RAM en Tablas del Sistema.....	139
Tabla 10.Total de memoria RAM asignado a formularios .....	140
Tabla 11.Tiempo de Carga en Tablas del Sistema .....	147
Tabla 12.Total de Tiempo de Carga asignado a formularios.....	148
Tabla 13. Promedio de Tiempo de Respuesta .....	155
Tabla 14. Consumo de Memoria RAM por cada Tabla .....	162
Tabla 15.Total de Memoria RAM.....	163
Tabla 16.Tiempo de Carga en Tablas del Sistema .....	170
Tabla 17.Total de Tiempo de Carga asignado a formularios.....	171



## RESUMEN

El desarrollo web es un factor principal que utilizan las organizaciones actualmente para mostrar sus servicios hacia los usuarios mediante internet. (Sánchez Osejo & Vera Cárdenas, 2011)

En investigaciones anteriores hemos recopilado información la cual hace referencia a una problemática antes de seleccionar un framework para posteriormente utilizarlo en la programación, ya que depende de diversas situaciones, las mismas que se deben tomar en cuenta como la gran diversidad de framework PHP, el tiempo de aprendizaje de un framework, el rendimiento, la mantenibilidad e incluso la adaptabilidad ya que muchos de ellos varían de versión cada cierto tiempo.

Por tales motivos se ha propuesto realizar un análisis comparativo de dos framework PHP que han sido elegidos según un top de framework PHP encontrados en portales web confiables, además de usar el criterio de popularidad tomado de estudios previos y como regla de medición hemos considerado al factor rendimiento el cual ha contenido indicadores que han sido probados mediante el desarrollo de un sistema realizado con cada framework, para posteriormente obtener los resultados que permitan determinar cuál es el mejor framework a utilizar. De tal manera esta investigación ha permitido ofrecer aportes a programadores en PHP, estudiantes, docentes en cada uno de los proyectos web que realicen.

### KEYWORDS

Framework PHP, MVC, MYSQL, Rendimiento, Eficiencia.]



## ABSTRACT

Web development is a major factor that organizations currently use to show their services to users via the internet. (Sánchez Osejo & Vera Cárdenas, 2011)

In previous research we have compiled information which refers to a problem before selecting a framework for later use in programming, as it depends on various situations, the same ones that must be taken into account as the learning time of a framework, performance , Maintainability and even adaptability since many of them vary in version from time to time.

For these reasons it has been proposed to perform a comparative analysis of two PHP frameworks that have been chosen according to previous studies and as a measurement rule we have considered the performance factor which has contained indicators that have been tested by the development of a system performed with each framework, To later obtain the results that allow to determine which is the best framework to use. In this way this research has allowed to offer contributions to programmers in PHP, students, teachers in each of the web projects that they perform.

### Keywords

Framework PHP, MVC, MYSQL, Performance, Efficiency.



## INTRODUCCION

Existen lenguajes de programación para desarrollo web que surgen con 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, uno de los principales es PHP, que se define como un lenguaje de código abierto, actualmente posee gran popularidad y es fácil de adecuar al desarrollo web ya que puede ser insertado en HTML. (Samaniego Larrea, 2015)

Además del uso de este lenguaje los programadores siempre han buscado agilizar el desarrollo de sus sistemas, mejorando los tiempos empleados, la seguridad surgiendo de esta manera los frameworks de desarrollo, que se definen como estructuras, que aceleran el proceso de programación, permiten reutilizar el código existente y promueven las buenas prácticas de desarrollo. (Gutierrez, 2017)

En aspectos más particulares tenemos los Frameworks de desarrollo de PHP que permiten mayor rendimiento en el desarrollo web, además brindan herramientas que permiten una estructura definida de esta manera dan lugar a crear aplicaciones con mayor facilidad y con una orientación a dar solución en menor tiempo posible a problemas. (Medhi Achour & Jakrub Vrana, 2016)

Hoy en día existe una gran diversidad de framework PHP, que al momento de elegir puede generar una confusión o una elección incorrecta, para esto existen criterios de selección como el tiempo de aprendizaje, la documentación de cada framework donde encontraremos sus rasgos más significativos para tener la claridad de elegir aquellos que de manera más optimas le permitan al desarrollador construir una mejor aplicación web. (F. Sierra, 2013)

En estudios previos de (Arcos Chalán & Chicaiza Inguillay, 2016) Mencionan que los desarrolladores de aplicaciones informáticas deben tener como prioridad dos aspectos, el primer aspecto a considerar es la construcción de aplicaciones en un tiempo determinado, el segundo aspecto es el desarrollo de aplicaciones que contengan estándares o patrones de diseño que permitan mayor



reutilización de código y fácil mantenimiento de los sistemas. Lo ideal es crear sistemas estandarizados, que aseguren el rendimiento sin importar futuras alteraciones, según como lo exponen.

En la presente investigación se ha realizado un análisis comparativo de dos framework PHP Yii y Laravel que han sido seleccionados según el criterio de popularidad de la investigación previa de (Prokofyeva & Boltunova, 2017), que además está reflejado en una tabla comparativa con los indicadores y su puntaje correspondiente. Además como regla de medición se considera al factor rendimiento del cual se han seleccionado previa investigación Tiempo de Respuesta, Cantidad de memoria RAM utilizada, Tiempo de Carga que serán probados mediante el desarrollo de un sistema realizado con cada framework, para luego obtener los resultados que permitirán determinar cuál es el mejor framework a utilizar.

Como resultado se ha obtenido que el framework Yii tiene un menor Tiempo de Respuesta con 14,05 segundos versus un 14,16 segundos del framework Laravel, en el indicador Uso de Memoria RAM el framework Yii hace un menor consumo con un resultado de 6,45 Megabytes(Mb) versus un 8,11 megabytes (Mb) del framework Laravel pero en el indicador Tiempo de Carga el framework Laravel obtiene un menor resultado de 1499094656,44242 microsegundos ( $\mu$ s), versus un 1499097950,92986 microsegundos ( $\mu$ s) del framework Yii.



## CAPITULO I: PROBLEMA DE INVESTIGACION

### 1.1 Situación Problemática

En la actualidad con el avance de las tecnologías de información, los desarrolladores de aplicaciones informáticas tienen como prioridad dos aspectos, el primer aspecto a considerar es el desarrollo de aplicaciones en un tiempo determinado, el segundo aspecto es el desarrollo de aplicaciones que contenga algún estándar o un patrón de diseño y mediante estos dos aspectos permitir una mayor reutilización de código y fácil mantenimiento de los sistemas. Los desarrolladores de software deben demostrar ser eficientes en el momento de plantear una solución a un problema, para esto se hace necesario buscar nuevas tecnologías que permitan simplificar la tarea de diseño y desarrollo de sistemas informáticos. Lo ideal principal es crear sistemas estandarizados, para que de esta manera aseguren el rendimiento y no tener problemas en futuras alteraciones, según como lo exponen (Arcos Chalán & Chicaiza Inguillay, 2016)

En la Investigación “Navigating the Framework Jungle for Teaching Web Application Development”. Se menciona que en muchos casos se desarrolla software simplemente utilizando habilidades para la programación, por la cual no es suficiente para realizar un buen desarrollo, para ello se propuso realizar la selección de frameworks de PHP, que contenga características más apropiadas para el desarrollo como la facilidad en el aprendizaje del framework, el uso en las organizaciones, metodología de cursos, soporte y capacitación requerida. (T. Chao, R. Parker, & Davey, 2013)

En la investigación” Análisis del Rendimiento de Frameworks PHP para Desarrollar Aplicaciones Web Óptimas”, el desarrollo de aplicaciones web se ha vuelto muy popular a nivel de que todas las empresas desean tener presencia en internet, esto origina nuevas herramientas de desarrollo como son los





framework, los cuales son muchos y con diversas características propias de cada uno, se mencionado que los framework son usados empíricamente sin considerar estudios previos referente a las características más sobresalientes de los principales marcos de desarrollo, además son elegidos a criterios propio no aseguran de ser los más utilizados en la actualidad. (Sánchez Osejo & Vera Cárdenas, 2011)

A partir de las investigaciones previas se determinó que el factor rendimiento es importante para medir el desarrollo web ,pero también se menciona que existen una gran de variedad de framework PHP ,lo cual necesario realizar un estudio comparativo que permita mostrar que framework son más populares y así seleccionarlos para compararlos usando el factor rendimiento como criterio fundamental que permitirá demostrar mediante un caso práctico que framework PHP es el mejor en términos de rendimiento, estos resultados será de utilidad a todas las personas interesados en el proceso de desarrollo web.

## **1.2 Formulación del problema**

¿Cuál es el mejor framework PHP en base al rendimiento mediante un análisis comparativo?

## **1.3 Delimitación de la Investigación**

La investigación se realiza en el Laboratorio de la escuela de Ingeniería de Sistemas de la Universidad Señor de Sipán.

Esta investigación se realiza en los meses comprendidos entre Abril y Julio del año 2017.

## **1.4 Justificación e Importancia**



El creciente avance de la tecnología web y la aparición de nuevas herramientas que agilizan el desarrollo web, hacen que utilicemos nuevas herramientas como son los framework PHP y debido a su gran diversidad se presenta la dificultad de conocer el más adecuado y que framework es eficaz en criterios de rendimiento.

Los framework de PHP son diversos, cada uno de ellos presentan características propias que al medirlas proporcionaremos resultados de gran aporte.

Una comparación de Framework PHP es importante antes de realizar la programación, esto ayudara a permitir una elección confiable y segura, de lo contrario podemos perder tiempo en el desarrollo como también tener problemas de rendimiento por no elegir el framework adecuado.

En esta investigación se ha realizado la comparación de framework, en donde se han elegido dos framework en base a investigaciones previas relacionado a frameworks PHP, además de considerar indicadores del factor rendimiento que luego se medirán en un sistema desarrollado con cada framework seleccionado.

Los resultados que se obtengan de la presente investigación serán de gran apoyo a programadores, docentes, estudiantes y personas que están inmersas en el desarrollo web en PHP Ya que podrán elegir un framework de PHP con indicadores de rendimiento permitirá un desarrollo más eficaz.

### **1.5 Limitación de la Investigación**

Una guía que permita una elección más práctica de framework.

La cantidad de framework que se puedan tomar para un caso práctico.



## 1.6 Objetivos

### 1.6.1 Objetivo general

Realizar un análisis comparativo para medir el rendimiento de frameworks PHP.

### 1.6.2 Objetivos específicos

- a) Analizar los framework PHP para su selección.
- b) Seleccionar los indicadores de Rendimiento para la comparación entre frameworks.
- c) Desarrollar un sistema con cada framework seleccionado.
- d) Evaluar los framework en base a indicadores de rendimiento.
- e) Identificar según cada indicador medido, que framework resulto mejor.



## CAPÍTULO II: MARCO TEÓRICO

### 2.1 Antecedentes de la Investigación

En la investigación “Estudio comparativo de productividad de frameworks PHP orientados a objetos para desarrollar el Sistema de Seguimiento de Incidentes de la Infraestructura de Red en la Escuela Superior Politécnica de Chimborazo”. Se plantea realizar una aplicación para que permita gestionar y dar seguimiento a las incidencias que se reportan diariamente en la Dirección de Tecnologías de la Información y Comunicaciones de la ESPOCH con el fin de reducir los tiempos de respuesta en la atención al cliente. Para realizar la comparación se utilizaron los frameworks CakePHP y Laravel, también se calcula el índice de productividad física con líneas ejecutables de medición de las líneas de código ejecutables (LDC) y el tiempo empleado (TE) para desarrollarlas (horas). Como resultados se obtiene que el código de las líneas por el tiempo empleado de CakePHP es de 92.903 LDC / TE y de Laravel con 39.102 LDC / TE. Finalmente el Sistema de la ESPOCH se realizó con el Framework ganador. (Samaniego Larrea, 2015) Esto indica que CakePHP permite el aumento de eficiencia en el desarrollo de aplicaciones web con un nivel más alto que Laravel.

Como aporte se puede decir que la comparación de framework es algo fundamental antes de realizar la programación, debido a que permite a que permitió comprobar características que sirvieron para el desarrollo a futuro.

En la investigación denominada “Análisis Comparativo de Frameworks para el Desarrollo de Aplicaciones Web en Java “. Menciona que al tener dificultad en la elección de un framework de calidad que posea las mejores características para el desarrollo de software, se planteó un caso Practico en un Sistema de Votación para evaluar cada framework (Spring, Struts, JSF, Angular JS), Luego de seleccionar los framework se utilizó el Modelo de Construcción de Calidad Individual y las características propuestas en la ISO 25010n y la técnica AHP



(Analytical Hierarchy Process). Como resultado se determinó que la ISO 25010n se adecua de forma correcta a la evaluación de frameworks.

Se obtuvo que el framework Spring es el que cumple de mejor manera las características como funcionalidad, fiabilidad, mantenibilidad, rendimiento, usabilidad, compatibilidad y portabilidad. (Acosta, 2015).

Como aporte a nuestra investigación se puede tomar en cuenta el indicador documentación según amazon, que permite la búsqueda de libros de framework.

Además en la Investigación “Estudio y análisis de los framework en PHP basados en el modelo vista controlador para el desarrollo de software orientado a la web”. Se analizó los rasgos más significativos de los distintos framework existentes en el mercado que permita al desarrollador tener claridad al momento de elegir el que mejor se adecue a su aplicación web, lo que se hizo fue una amplia Investigación de la información disponible de los diecisiete framework PHP en estudio y se realizó tablas comparativas en base a las características encontradas como autenticación, cache, base de datos entre otros.

Los framework en estudio demostraron que estos ahorran tiempo y tareas en el momento de un desarrollo web ya que permiten al usuario una mayor facilidad en el proceso de desarrollo web. (F. Sierra J. A., 2013)

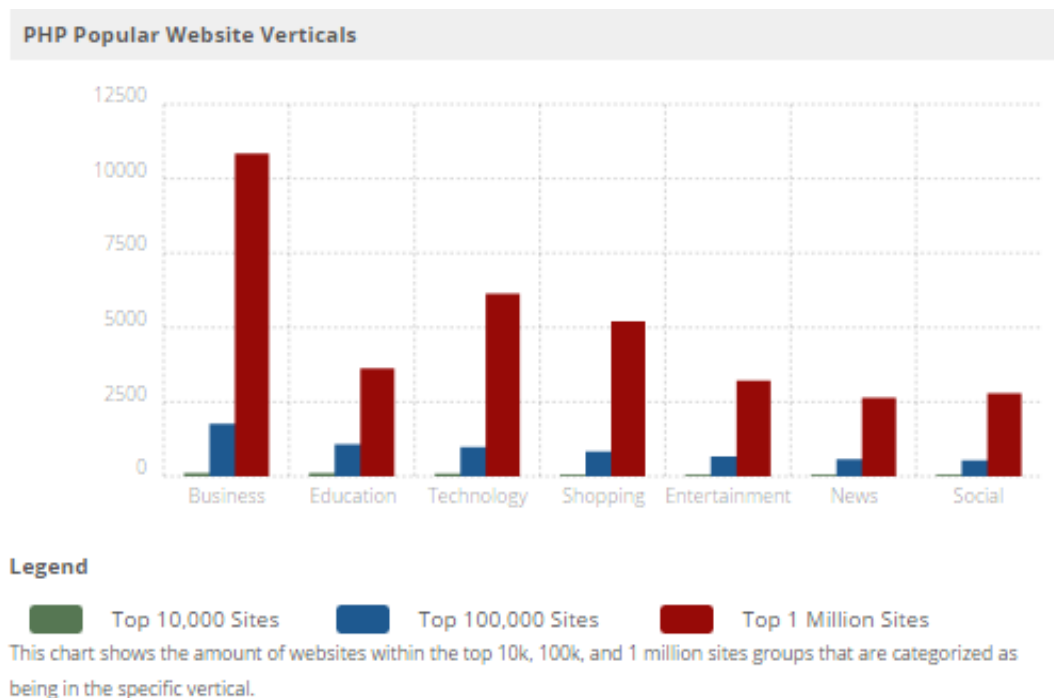
Se pudo concluir que cada framework tiene su propia forma de trabajo, sus características y lo que lo hace único como módulos plantillas y autenticaciones.

A pesar que en toda investigación se realiza una amplia información, se debe llegar a un caso Práctico, esto permitirá la mayor credibilidad de los estudios realizados y mayor confianza para elegir en este caso un framework de PHP.

Grafico. Este gráfico muestra la cantidad de sitios web que usan PHP en diversas áreas como Negocios, Educación, Tecnología, Redes Sociales, etc



Gráfico 1.PHP usado en los sitios Web



Fuentes: (Built With, 2017)

En la Investigación “Analyzing best practices on Web development frameworks: The lift approach”. Las compañías como Siemens e IBM y las redes sociales como Twitter y Foursquare han desarrollado y aplicado el uso del enfoque Escala y Ascensor, el cual ha generado grandes ventajas para el desarrollo web, pero al ser necesario un análisis comparativo que identifique las mejores prácticas y así permitir a los desarrolladores realizar más aplicaciones web basados en estos dos enfoques. Para esto se ha comparado ocho frameworks los cuales son JSF, Struts, CakePHP, Ruby on Rails Django y Catalyst, donde se ha realizado diferentes aplicaciones para identificar las mejores prácticas en desarrollo web, tomando en cuenta los siguientes indicadores Soporte Ajax, Cloud Computing, Mensajes de Error Personalizados, Extensibilidad y Depuración.

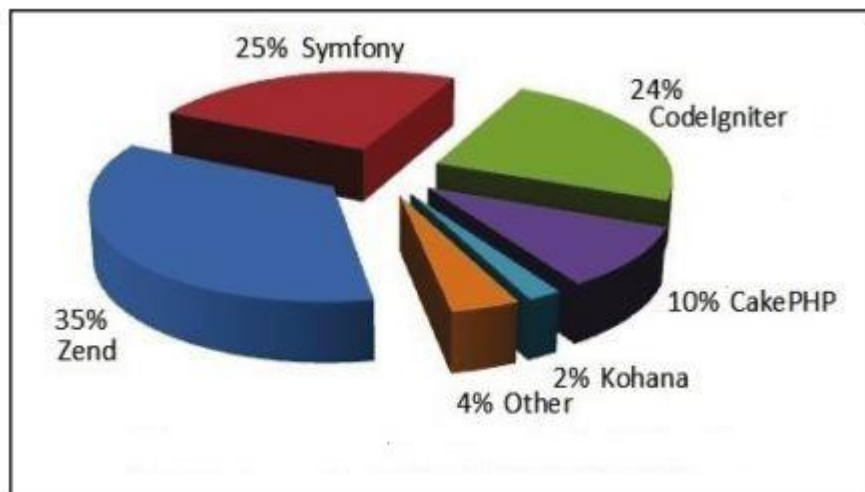
Como resultado se demostrado que utilizando el enfoque Escala y Ascensor hay un menor tiempo en el desarrollo web. (Salas Zárata, y otros, 2015)



Como aporte se la investigación previa se considera que al realizar un proyecto de desarrollo web permitirá una reducción de errores y de pérdida de tiempo.

Otro enfoque para determinar el marco más utilizado es encuestar a los desarrolladores. Se realizó una encuesta a los desarrolladores en línea en el portal Dzone, en la que se tenía más de mil encuestados. Y Zend Framework resultado ganador con un 35%.

Gráfico 2. Resultados de la encuesta de framework PHP



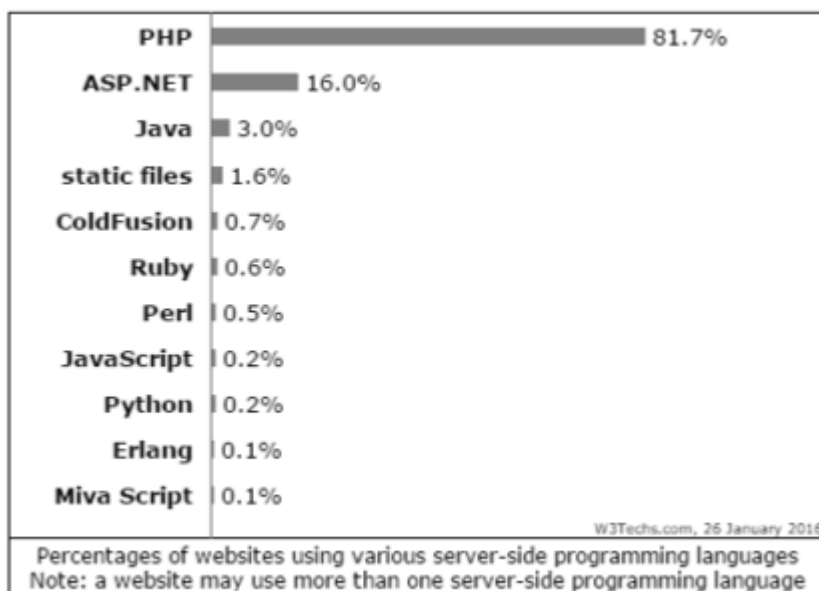
Fuente: (Salas Zárate, y otros, 2015)

## 2.2 Estado del arte

En la investigación "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems", En esta investigación se mostraba diversos esquemas de programación en PHP, de los cuales se eligen dos framework PhalconPHP y Symfony2 en base al criterio de popularidad y en base a la comparación de criterios como enrutamiento, motor de plantilla, en donde

también se muestra una descripción de la arquitectura y las características principales que poseen los framework seleccionados. Se eligió indicadores como Tiempo de Carga, Consumo de memoria y Tiempo de Respuesta, se realizaron las pruebas de rendimiento, usando la misma tarea para determinar la eficacia de los framework seleccionados, y para esto se seleccionó una sección de la lista de cajeros del " Sistema de Reserva de Entradas". Para realizar las pruebas se utilizó la herramienta Apache Benchmarking. (Prokofyeva & Boltunova, 2017).

Figura 1. Uso de lenguajes de programación



Fuente. (Prokofyeva & Boltunova, 2017)

Se demostró los resultados de cada indicador medido, el cual permite a los desarrolladores elegir un correcto framework PHP en base a indicadores como hacen recomendaciones que permiten.

Como Aporte a nuestra investigación es considerar el indicador rendimiento para permitir la comparación de diferentes framework PHP.





En la investigación "Análisis del Rendimiento de Frameworks PHP para Desarrollar Aplicaciones Web Óptimas", el desarrollo de aplicaciones web se ha vuelto muy popular a nivel de que todas las empresas desean tener presencia en internet, esto origina nuevas herramientas de desarrollo como son los framework, los cuales son muchos y con diversas características propias de cada uno, se menciona que los framework son usados empíricamente sin considerar estudios previos sobre las principales características más relevantes de los marcos de desarrollo, además son elegidos a criterios propios no aseguran de ser los más utilizados en la actualidad. Para tal investigación se han seleccionado tres framework de desarrollo web como son ZendFramework, CakePHP y CodeIgniter, para determinar cual tiene el mejor desempeño en el rendimiento y estabilidad. Determinando que el ámbito cuantitativo se ha obtenido mediante la comparación de Indicadores como: Ingeniería de carga, Línea Base, Cargas transaccionales de alta, Cargas transaccionales de baja e Integridad.

Y como resultado se ha obtenido que el mejor framework para el rendimiento en el desarrollo de aplicaciones web óptimas es CodeIgniter con un puntaje de 89.70%. (Sánchez Osejo & Vera Cárdenas, 2011)

También podemos mencionar al estudio "Análisis de los Frameworks Yii y Zend para mejorar la Usabilidad y la Compatibilidad del Sitio Web de la Cooperativa de Ahorros y Crédito Minga Ltda.". Se pretende mejorar la usabilidad y la compatibilidad en un sitio web para esto se realiza la comparación de los frameworks Yii y Zend, y se miden indicadores como tiempo de respuesta, eficiencia al realizar una tarea, facilidad de uso, satisfacción del usuario y compatibilidad del sitio web.

Utilizando como herramientas YSlow de Firebug, Pingdom, BrowserShots-screenshot.

Como resultado final se obtuvo el 91.25 % de usabilidad y compatibilidad con el framework Yii y el 80.75% con el framework ZEND. Se seleccionó el framework Yii ya que supera en un 10.50 %. (Miranda Chinlli & Rochina Rochina, 2015)



Estos resultados nos permite conocer las herramientas que determinan una medición o comparación entre framework, las cuales se pueden tomar para nuestra investigación.

En el siguiente estudio “Navigating the Framework Jungle for Teaching Web Application Development”. Se menciona que en muchos casos se desarrolla simplemente utilizando habilidades para la programación, por la cual no es suficiente para realizar un buen desarrollo web, en esta investigación se trata de resolver la incógnita de que si los cursos de desarrollo web deberían considerar el tema de usar los framework de PHP como forma de enseñanza en las aulas de pregrado para de esta manera aportar al desarrollo de software

A partir de la problemática planteada se había propuesto realizar la selección de un framework de php, que contenga características más apropiada para un plan de estudios, además de la revisión de información sobre del patrón de MVC (Model-View-Controller), en esta investigación se toman criterios como como la facilidad en el aprendizaje del framework, el uso en las organizaciones, metodología de cursos, soporte y capacitación requerida. Como resultado se menciona ganador al framework Yii, el cual cumple en mayor parte con las características preestablecidas por la investigación Como resultado se refiere a diversos factores pedagógicos y la forma de uso, un indicador importante es la forma de aprendizaje de los framework. (T. Chao, R. Parker, & Davey, 2013)

La forma de aprendizaje de los framework es un criterio importante para antes empezar a programar, ya que al tener un conocimiento previo la curva de aprendizaje será menor y esto se verá reflejado en el tiempo de desarrollo web.

## **2.3 Base Teórica Científica**

### **2.3.1 Framework web**

#### **a) Definición de Framework**



El termino framework se define como un conjunto de herramientas, librerías, convenciones y buenas prácticas que pretenden unir las tareas repetitivas en módulos fácilmente reutilizables. Además se menciona que un framework es un conjunto de componentes que aporta mucho a los lenguajes de programación con el desarrollo de aplicaciones o sistemas web.

Los objetivos principales que se obtiene al usar un framework son los siguientes: acelerar el desarrollo, permitir reutilizar y difundir las buenas prácticas de desarrollo. (Gutierrez, 2017)

### **b) Objetivos de los Frameworks Web.**

Entre los principales objetivos de los frameworks web están (Lopez Saavedra, 2009).

- b.1) Tener un rápido desarrollo.
- b.2) Permitir reutilizar código.
- b.3) Difundir las buenas prácticas de desarrollo gracias al uso de patrones de diseño.
- b.4) Disminuir el esfuerzo en el desarrollo.
- b.5) Tener como aliado a las metodologías de desarrollo Ágiles como XP, Scrum, AD y otros

### **c) Características de los Frameworks Web.**

Los frameworks presentan características que los identifica, entre las principales están: (Lopez Saavedra, 2009).

- c.1) Abstracción de urls y sesiones.
- c.2) Acceso a datos
- c.3) Controladores
- c.4) Autenticación y control de acceso
- c.5) Separación entre diseño y estilo.



#### **d) Ventajas de usar Frameworks Web.**

Al usar un framework, la velocidad de desarrollo aumenta, incrementando y permitiendo la productividad de desarrollo web, todo esto también va en relación al conocimiento de los aspectos básicos de utilización que presenta un framework web.

La reducción de costos es otra ventaja con respecto al trabajo que se debe realizar, ya que un framework permitirá al desarrollador dar prioridad a los aspectos esenciales que se debe realizar en un desarrollo web.

La utilización de estándares y convenciones de código de los frameworks, permite mejorarlas aplicaciones web al momento de realizar un trabajo en equipo, de esta manera permite una mejor comprensión del código y un trabajo más organizado. (José Samaniego, 2015).

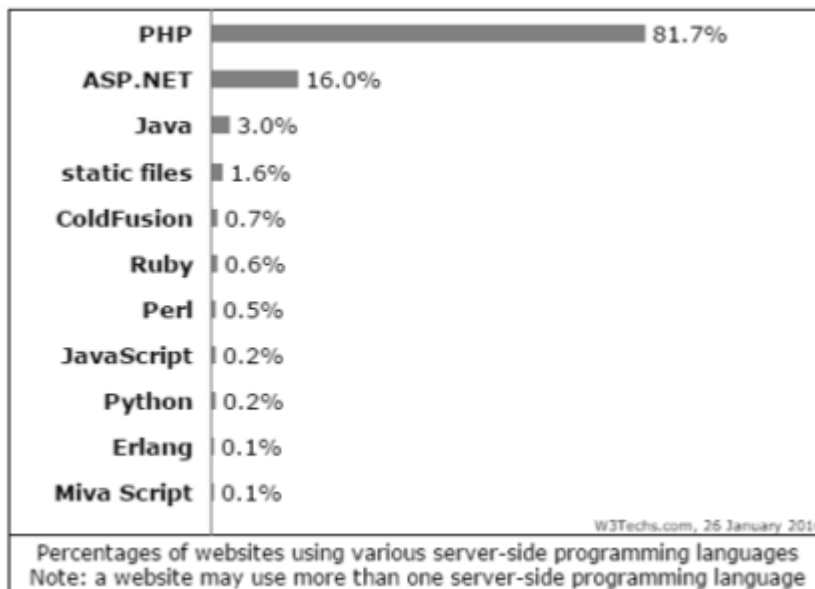
### **2.3.2 Lenguaje de Programación PHP**

El lenguaje PHP (Hypertext Pre-Processor) se define como un lenguaje de código abierto, el cual está destinado para el desarrollo web y que puede ser insertado en HTML. En este lenguaje el código es ejecutado en el servidor, de esta manera se genera sentencias HTML y envía hacia el cliente. (php.net, 2017)

Aunque este lenguaje se puede utilizar para cualquier tipo de programa, el punto fundamental es la generación dinámica, donde ha podido lograr su máxima popularidad. (Manuel & Ildefonso, 2011).



Figura 2. Uso de lenguajes de programación del lado del servidor para sitios web.



Fuente. (Prokofyevaa & Boltunovaa, 2017)

Figura 3. Lenguajes de Programación de Lado del Servidor

Ranking	Lenguaje	usage	Fecha
1.	PHP	82.6%	1 April 2017
2.	ASP.NET	15.1%	
3.	Java	2.6%	
4.	static files	1.5%	
5.	ColdFusion	0.6%	

© W3Techs.com

Fuente: (w3techs, 2017)

### 2.3.2.1 Características de PHP

A continuación muestran algunas de las características más importantes con respecto a este lenguaje. (José Samaniego, 2015).



- 1) Es un lenguaje libre, multiplataforma, además de ser seguro y confiable.
- 2) Está orientado al desarrollo de web, permitiendo el acceso a determinada información de una Base de Datos.
- 3) Tiene una mayor capacidad de conexión con distintos motores de base de datos, pero destaca su compatibilidad con los gestores de base de datos como MySQL y PostgreSQL.
- 4) Posee la capacidad de ampliar su potencial utilizando una amplia cantidad de extensiones.
- 5) Ofrece una Buena Documentación.
- 6) Se puede aplicar las técnicas de programación orientada a objetos (POO).

### **2.3.2.2 Componentes y Funcionamiento de PHP**

Los componentes del lenguaje PHP son los siguientes:

PHP es un lenguaje que funciona en servidores web, por lo que se necesita instalar un servidor web en la PC, como por ejemplo Apache. Un editor de texto para programar como Sublime Text.

Además de asegurarse de tener instalado y funcionando el intérprete de PHP versión 4 o 5, incluso la más actual, la versión 7.

Una opción recomendable es utilizar XAMPP, que es una herramienta que incluye Apache y PHP.



Figura 4. Funcionamiento del Lenguaje PHP



Fuente: (Grupo CODESI - Training & Certificación, 2016)

### 2.3.3 Frameworks de PHP

#### 2.3.3.1 FRAMEWORK YII

(Yii, 2017) Se define como un framework de código abierto, que es usado para el desarrollo de aplicaciones Web, permite el desarrollo rápido, este framework trabaja para simplificar el desarrollo de aplicaciones, permitiendo un desarrollo final eficiente y con la opción de ser extensible, además de permitir un fácil mantenimiento. Yii es un framework de PHP que permite mejorar el desarrollo Web.

#### Características

- a) Permite el desarrollo de un código limpio y reutilizable.
- b) Emplea el patrón MVC, lo que da lugar a una separación de la lógica y la presentación.
- c) Realiza la recolección de entrada de forma fácil y seguro.
- d) Posee la validación de entrada, el filtrado de salida, la inyección de SQL y también la prevención de Cross-site scripting.



- e) Permite el soporte de autenticación.
- f) Es compatible con la autorización a través del control de acceso basado en roles jerárquica (RBAC).
- g) En particular cada método o propiedad está debidamente documentada.
- h) Existen diversos libros, así como también tutoriales para que de esta manera el aprendizaje sea más sencillo.
- i) Yii proporciona una biblioteca de la extensión que consiste en los aportes que realizan cada usuario del framework los usuarios, esto va a permitir que la lista de características aumenten y se mantenga disponible siempre.
- j) El framework Yii ofrece la herramienta de migración de aplicaciones, en la que podemos migrar de otros framework PHP como Zend, CodeIgniter, CakePHP y Symphony.

### Estructura de una aplicación en Yii

A continuación se muestra la estructura de las diferentes carpetas que ofrece Yii Framework:

1. **Directorio config:** Contiene archivos de configuración
2. **Directorio assets:** Contiene archivos de recursos públicos.
3. **Directorio commands:** Este directorio posee las clases de comandos de consola
4. **Directorio controllers:** Contiene los archivos de clase de controladores.
5. **Directorio models:** Contiene archivos clase de modelos de las tablas.
6. **Directorio runtime:** Contiene archivos temporales generados.
7. **Directorio views:** Contiene archivos de vista de controladores y de diseño.
8. **Directorio web:** Aquí está la raíz de la aplicación web ,es decir contiene los archivos que permiten acceder vía Web





9. **Directorio vendor** Esta carpeta es administrada por Composer, aquí verás las dependencias y librerías del proyecto.
10. **index.php**: archivo de entrada de la aplicación web.

### 2.3.3.2 FRAMEWORK LARAVEL

(Laravel, 2017) Se define como un framework de aplicaciones web, la cual ayuda en el desarrollo de la mayoría de los proyectos web, además proporciona las herramientas necesarias para aplicaciones que son grandes y robustas. Así como también una excelente combinación de sencillez, elegancia e innovación las cuales son herramientas importantes que se necesita para construir cualquier aplicación.

#### Características

- a) Tiene una amplia y completa biblioteca de documentación.
- b) Presenta una capacidad de enrutamiento rápido.
- c) Muestra múltiples back-ends de sesión y caché de almacenamiento.
- d) Presenta una intuitiva ORM base de datos.
- e) Es un framework robusto en el procesamiento de trabajo en segundo plano.

#### Estructura de una aplicación en Laravel

1. **Directorio config** Aquí encontramos la configuración.
2. **Directorio database** Contiene la configuración de Base de Datos.
3. **Directorio public** Contiene los archivos estáticos de tu aplicación, en este directorio coloca los archivos html, javascript, css, vídeos e imágenes.
4. **Directorio resources** Aquí contiene los archivos de vistas, es decir, acá está el Frontend.



5. **Directorio storage** Este directorio contiene la carpeta temporal de Laravel, además es donde se autogeneran logs, cache de templates.
6. **Directorio tests** En este directorio es donde se guardan las clases que permiten realizar el unit testing a los diversos componentes de una aplicación web.
7. **Directorio vendor** Este directorio es administrada por Composer, aquí se encuentran las dependencias y librerías del proyecto.

### 2.3.4 Modelo Vista Controlador (MVC)

Se define como un patrón de arquitectura de software, en la cual se divide los datos de la aplicación, la interfaz de usuario y también la lógica de control, las cuales la separa en tres componentes distintos. (Zabala Hidalgo Ximena Liliana, 2008)

#### **Modelo**

Es la capa con la que el sistema funciona, es decir la lógica de datos garantiza la integridad de estos y donde permite derivar nuevos datos. El modelo es donde se permite el acceso a las bases de datos como SQLServer, MySQL y otras.

#### **Vista**

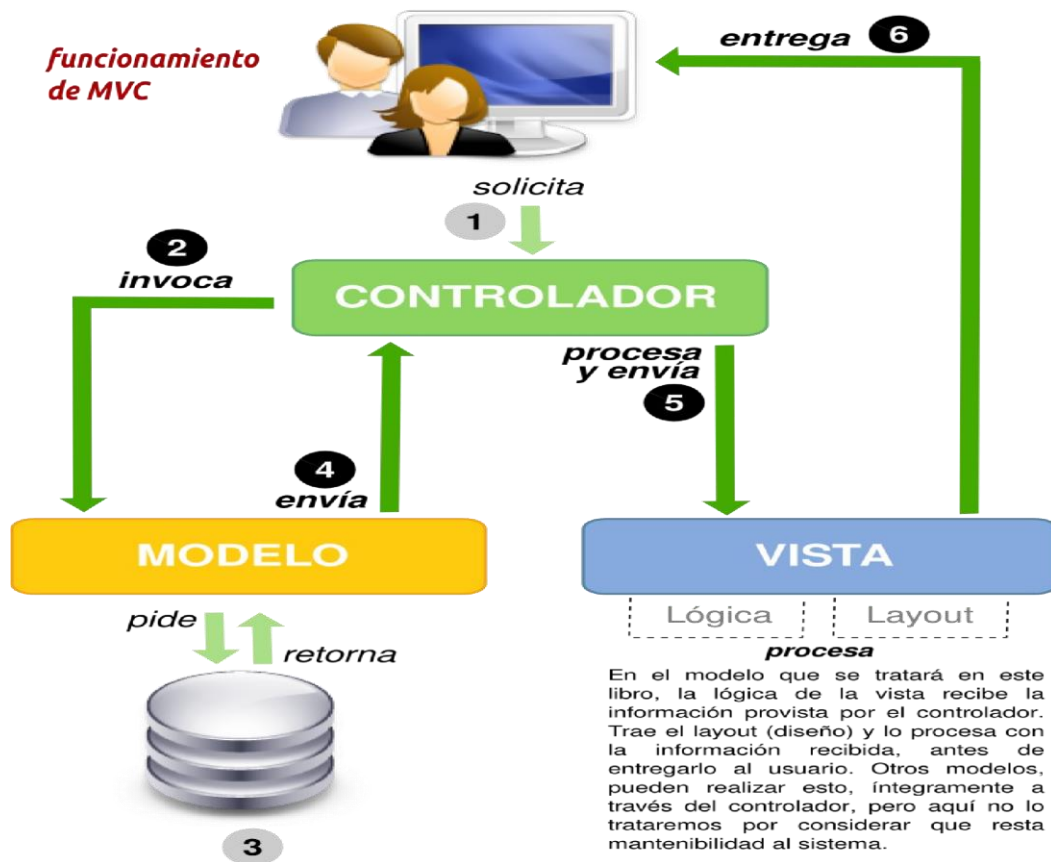
Es donde contiene el código para interactuar entre el usuario y la aplicación, el cual hace uso de HTML, PHTML, CSS.

#### **Controlador**

Esta capa responde a eventos, normalmente son acciones que el usuario invoca cambios en el modelo, es decir contiene el código necesario para responder a las acciones que se solicitan en la aplicación.



Figura 5. Funcionamiento del patrón Modelo -Vista-Controlador



Fuente: (Bahit, Eugenia, 2013)

### 2.3.5 MVC con PHP

Para realizar el uso de MVC en PHP es necesario de crear una estructura de fichero, a continuación se muestra en la siguiente figura.



Figura 6. Ficheros para implementar Modelo, Vista y Controlador en PHP



Fuente: (Victor Robles, 2013)

En este caso para el uso de los frameworks tendremos los directorios como:

**Config:** Aquí se crearán los ficheros de configuración de la base de datos y diferentes configuraciones a nivel global.

**Controller:** En la arquitectura MVC los controladores realizarán la función de recibir y filtrar datos que van a llegar desde las vistas, también la acción de llamar a los modelos y por último pasar los datos de estos hacia las vistas.

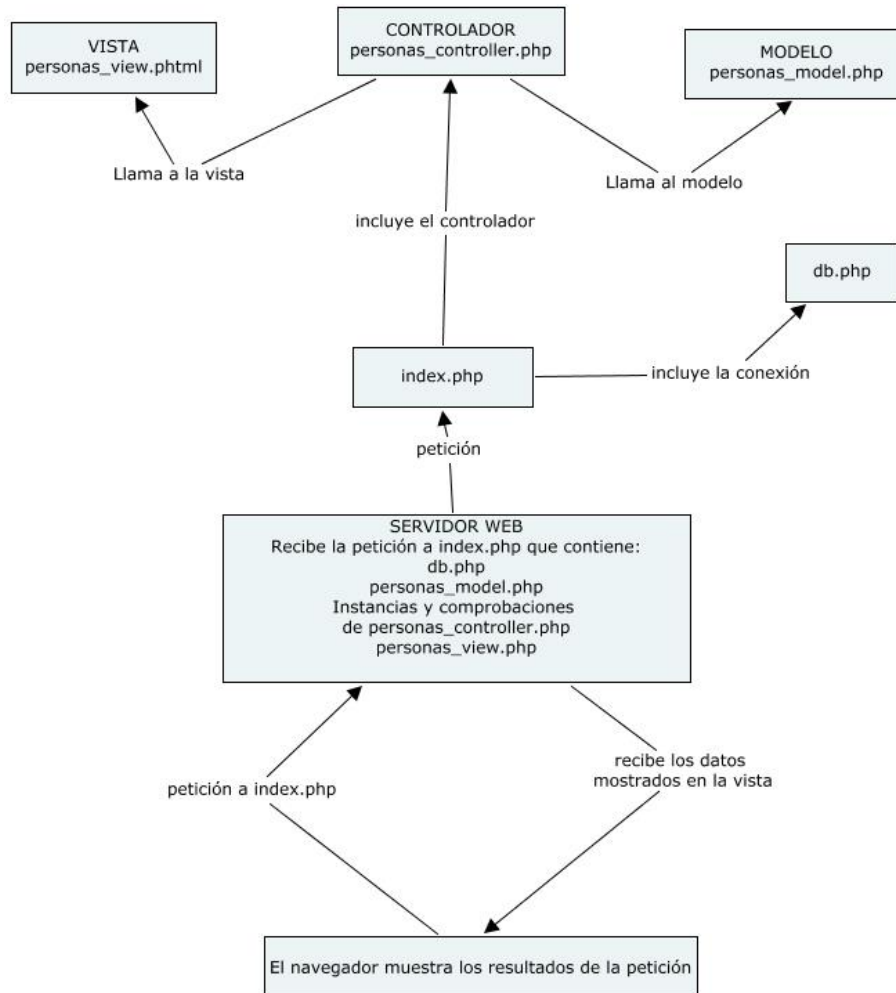
**Model:** Aquí se encontrarán los modelos, del paradigma orientado objetos tenemos que crear una clase por cada tabla de la base de datos, estas clases servirán para crear objetos de ese tipo de entidad. También habrá modelos de consulta hacia la Base de Datos, las cuales contendrán consultas más avanzadas que estarán relacionadas con varias entidades.

**View:** En esta carpeta estarán las vistas, es donde se mostrarán los datos, por lo tanto es lo que verá el usuario.

**Index** Este será el controlador frontal, es decir el lugar por el que pasará plenamente todo a la aplicación.



Figura 7. Esquema del uso de MVC en PHP



Fuente: (Victor Robles, 2013)

### 2.3.6 Base de Datos MySQL

(MySQL, 2017)MySQL se defino como uno de los sistemas de bases de datos más populares en el desarrollo de aplicaciones web, la cual se caracterizan por su alto rendimiento para trabajar con bases de datos.

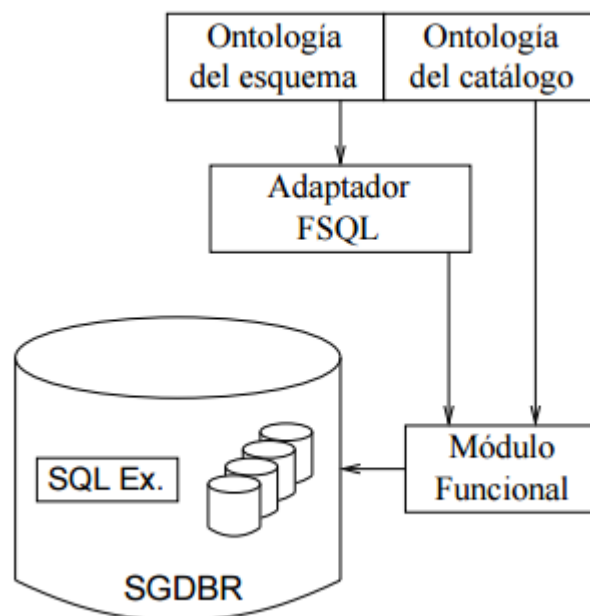


Además gran cantidad de aplicaciones web están desarrolladas con PHP y MySQL.

Es un sistema que representa a un Sistema Gestor de Base de Datos (SGBD), en donde se puede insertar las estructuras del catálogo, pero para la ejecución de sentencias de definición y manejo de datos difusos, se realizara en un módulo externo, la cual este será el que se ocupe de enviar las consultas a la base de datos en un lenguaje entendible, es decir en SQL para de esta manera obtener los resultados y dar formato a los datos se salida para que sean mostrados al usuario.

MySQL es la base de datos de código abierto más popular, ya que posee sus potentes funciones, además es fácil de configurar y usar. A continuación se muestra la Arquitectura de un SGBDR

Figura 8.Arquitectura de integración con un SGBDR



Fuente: (Carmen Martínez)



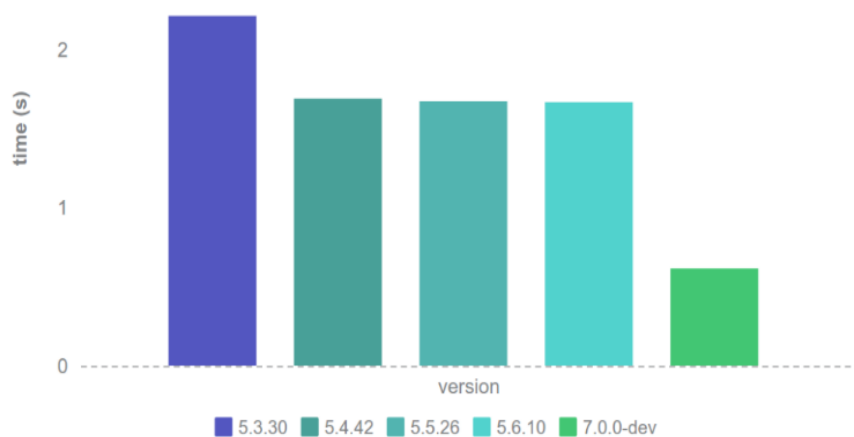
### 2.3.7 Rendimiento

El objetivo final de cualquier aplicación es una experiencia de usuario de alta calidad, las aplicaciones que se ejecutan lentamente y tiempos de respuesta más retrasadas van a permitir al abandono de los sitio web por parte de los usuarios.

Los clientes exigen respuesta rápidas casi en tiempo real de cualquier aplicaciones web, lo que un desarrollador debería considerar este factor como fundamental para tratar de no permitir largos retrasos.

Con el utilizar un framework de PHP se podrá eliminar los tiempos de respuesta lentos y reducir al mínimo la utilización de recursos, para de esta manera permitir el rápido acceso a las aplicaciones desarrolladas. (Zend, 2017)

Gráfico 3. Tiempo que tarda que tarda cada versión de PHP



Fuente: (Dzone, 2017)



## 2.4 Definición de Términos Básicos

- a) **PHP:** Se define como un lenguaje de programación de gran uso y de código del lado del servidor, el cual está diseñado para ser utilizado en el desarrollo web dinámico.
- b) **Pruebas de Rendimiento:** Son las diferentes pruebas que se hacen para poder determinar lo rápido que realiza una tarea un determinado sistema.
- c) **MVC:** Se define como un patrón de arquitectura de software, que divide los datos, la lógica de una aplicación y también el módulo que se encarga de trabajar los eventos y las comunicaciones.
- d) **GUI:** Graphical User Interface - Interfaz Gráfica de Usuario.
- e) **IDE :** Integrate Development Enviroment - Entorno de Desarrollo Integrado
- f) **COMPOSER:** Es un administrador de dependencias para los proyectos.





## CAPÍTULO III: MARCO METODOLÓGICO

### 3.1 Tipo y diseño de investigación

#### 3.1.1 Tipo de Investigación

La presente investigación pertenece a una investigación de tipo Tecnológica - Aplicada, esto requerido a que se utilizan los conocimientos en la práctica, para demostrar mejoras notables para una aplicación.

#### 3.1.2 Diseño de Investigación

El diseño para esta Investigación es Cuasi-Experimental, debido que no se utiliza ningún tipo de selección aleatoria, para seleccionar los grupos en que se prueba una variable.

### 3.2 Población y muestra

#### 3.2.1 Población

Se tomó los resultados de los top 10 framework PHP más usados en base al criterio de popularidad que se obtienen de los portales OpenWebinars, CodersEye, ArpaTech, DesignsParkle, Apptechmobile.

Los framework PHP seleccionado son Laravel, CodeIgniter, FuelPHP, CakePHP, Symfony, Zend, Phalcon, Yii, Slim y PHPixie.

#### 3.2.2 Muestra

Se realizó una tabla comparativa **Anexo 1**, en la que se detalla los criterios de selección.

Se eligió dos framework PHP: Yii y Laravel

### 3.3 Hipótesis

El mejor framework PHP en base a indicadores de rendimiento es Laravel.

### 3.10 Principios de rigor científico



- a) **Fiabilidad:** Se refiere al grado en que los instrumentos repiten las mismas medidas en circunstancias iguales.
- b) **Transferibilidad:** Es cuando se busca ofrecer el conocimiento sobre el contexto que va a permitir transferir las conclusiones a los mismos contextos
- c) **Neutralidad:** Refiere a la garantía de que los resultados obtenidos en una investigación no estarán desviados por motivaciones, intereses, y perspectivas propias del investigador.

### 3.4 Variables

#### 3.4.1 Independiente

Se tomó el Criterio de Popularidad en base en base a la investigación “Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems” (Prokofyeva & Boltunova, 2017)

#### 3.4.2 Dependiente

Medición de indicadores de Rendimiento en base a la investigación “Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems” (Prokofyeva & Boltunova, 2017)



### 3.5 Operacionalización

Tabla 1. Tabla de Operacionalización

VARIABLES	CONCEPTO	INDICADORES
<p><b>INDEPENDIENTE:</b> <b>Criterio de Popularidad</b></p>	<p>Se refiere al impacto que genera en la comunidad de desarrollo mediante las ventajas que esta muestra en relación a otros framework.</p> <p>Se proporciona características para el desarrollador y de gran importancia para una mejor selección.</p>	<ul style="list-style-type: none"> <li>➤ Compatibilidad de Base de Datos</li> <li>➤ Seguridad</li> <li>➤ Validación Integrada</li> <li>➤ Sistema de Plantillas Rápido y Flexible</li> <li>➤ Compatibilidad para PHP.</li> <li>➤ Documentación</li> <li>➤ Ruteo de URL Flexible</li> <li>➤ Scaffolding</li> <li>➤ Mapeo De Objetos Relacional - ORM</li> <li>➤ Fácil Lectura De Código</li> </ul>



VARIABLES	CONCEPTO	INDICADORES	FÓRMULAS
<p><b>DEPENDIENTE:</b></p> <p><b>Medición de Indicadores de Rendimiento</b></p>	<p>En informática, es la medida de aplicaciones desarrolladas. (Zend, 2017), también se le llama a la cuantificación de la velocidad con la que se hace un proceso y también incluye la cantidad de recursos que se usan</p>	<p>Tiempo de Respuesta (segundos=s)</p>	<p><math>TR=(TI+TF)/10</math></p>
	<p>En una PC, el rendimiento no solo depende del microprocesador, sino también de la suma de diversos componentes.</p>	<p>Uso de memoria RAM (megabytes=mb)</p>	<p>RAM = <math>(MI+MF)/(1024*1024)/2</math></p>
	<p>Con el utilizar un framework de PHP y un servidor adecuado se podrá eliminar los tiempos de respuesta lentos y reducir al mínimo la utilización de recursos permitiendo de esta manera el rápido acceso a las</p>	<p>Tiempo de Carga (microsegundos= <math>\mu</math>s)</p>	<p><math>TC=(TI+TF)/2</math></p>



	aplicaciones desarrolladas. (Zend, 2017)		
--	--	--	--

Fuente Propia.



### 3.6 Abordaje metodológico, técnicas e instrumentos de recolección de datos

#### 3.6.1 Abordaje Metodológico

**Cuasi-Experimental:** El diseño para esta Investigación será Cuasi-Experimental, debido a que no se utiliza ningún tipo de selección aleatoria, para seleccionar los grupos en que se prueba una variable.

#### 3.6.2 Método y técnica de recolección de datos

**Guía de observación:** Consiste en el registro seguro de comportamientos y las diversas situaciones observables a través de un conjunto de dimensiones e indicadores.

#### 3.6.3 Instrumentos de recolección de datos

**Investigación bibliográfica:** Son un conjunto de técnicas y estrategias que se emplean para acceder a aquellos documentos que contiene la información, estos medios son: libros, papers, publicaciones, tesis y también el uso de internet.

### 3.7 Procedimiento para la recolección de datos

El Proceso realizado ha sido los siguientes:

- a) **Se seleccionara los framework:** En este caso se obtendrá de investigaciones de confianza como artículos científicos, que permitan obtener los dos mejores framework PHP.
- b) **Utilizar criterios:** Se refiere a realizar una selección de indicadores que pertenezcan al factor rendimiento, para posteriormente ser usados.
- c) **Se desarrollara un sistema web,** Este sistema se ha desarrollado con los dos framework elegidos y se ha usado el lenguaje PHP, versión 5.6.30, además de la Base de Datos MySQL versión 5.2.44



**d) Se obtendrá resultados,** Se refiere a que cada indicador emitirá los resultados correspondientes, según los indicadores seleccionados

### 3.8 Análisis estadístico de datos

#### a) Tiempo de Respuesta

Los datos serán evaluados con unos indicadores relacionados al tiempo, donde se consideran dos valores para el resultado.

El tiempo total de la consulta resultara del tiempo total final menos el tiempo inicial, como se muestra en la siguiente formula:

$$TR=(TI+TF)/10$$

Donde:

TR = Tiempo de Respuesta

TI = Tiempo inicial de Respuesta

TF = Tiempo final de Respuesta

#### b) Uso de Memoria RAM

El uso de memoria RAM se midió mediante la función de php `memory_get_peak_usage()` para lo cual se realizó un cálculo entre el uso de memoria RAM final y el uso de memoria RAM inicial y el resultado obtenido en bytes se convirtió a megabytes.

$$RAM = (MI+MF)/1024*1024)/2$$

#### b) Tiempo de Carga



El Tiempo de Carga se midió mediante la función de php microtime () para lo cual se realizó un cálculo en microsegundo y aplicando la siguiente fórmula.

$$TC= (TI + TF)2$$

### 3.9 Principios éticos

- a) Veracidad:** Este principio se toma tal como lo expresa el Código Deontológico del Colegio de Ingenieros del Perú la cual se encuentra en el Capítulo III “Faltas Contra la Ética Profesional y Sanciones”, específicamente en el Sub Capítulo II “De la Relación con El Público” en su Artículo 105: Los ingenieros serán objetivos y veraces en sus informes, declaraciones o testimonios profesionales, expresarán opiniones en temas de ingeniería solamente cuando ellas se basen en un adecuado análisis y conocimiento de los hechos en competencia técnica suficiente y convicción sincera.
- b) Integridad:** Refiere a usar los diferentes conocimientos y habilidades adquiridas para poder mejorar el bienestar humano.
- c) Honestidad:** Corresponde a ser honesto e imparcial, y servir con plena fidelidad al público, a los empleados, y también a los clientes.
- d) Competitividad:** Es la manera de esforzarse para aumentar el nivel de competencia y obtener el prestigio de ingeniería como profesión.
- e) Apoyo:** Refiere al apoyar a la sociedad profesional y a las técnicas de sus correspondientes disciplinas.





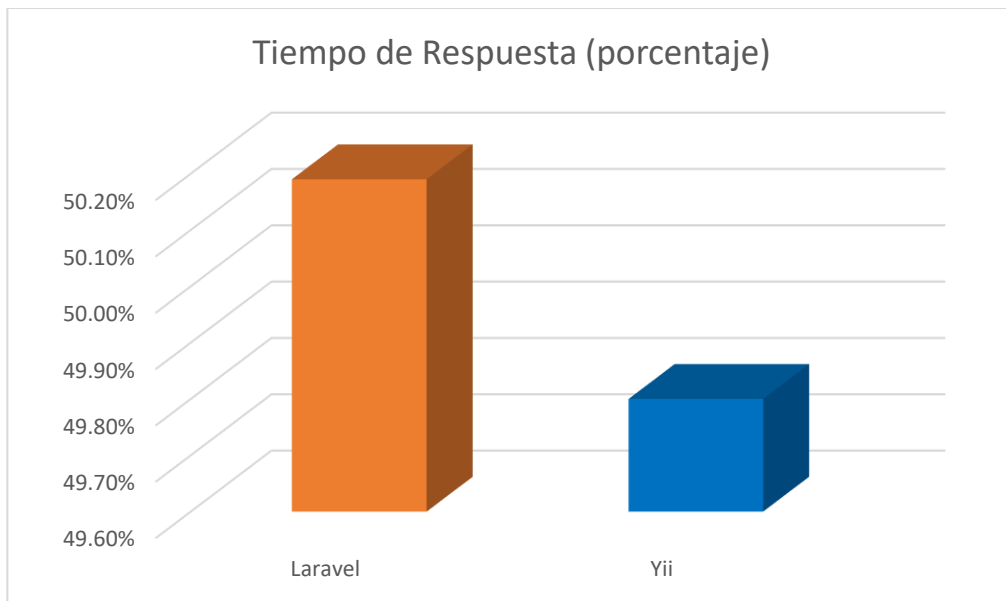
## CAPÍTULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

### RESULTADOS

#### 4.1. Resultados en tablas y gráficos

##### 4.1.1 Tiempo de Respuesta

Gráfico 4.Comparacion de Ambos Framework Laravel Vs Yii

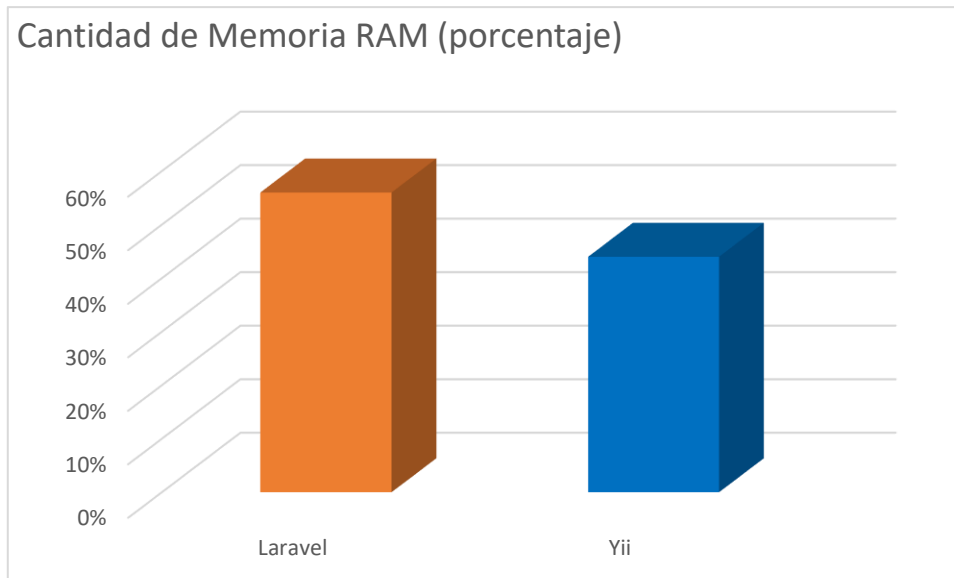


Luego de realizar las pruebas correspondientes explicadas a detalle en el Capítulo V.en la presente gráfica se pudo observar que el framework Yii tiene un menor Tiempo de Respuesta con 14,05 segundos versus un 14,16 segundos del framework Laravel, demostrando una diferencia 0,11 segundos.



#### 4.1.2 Consumo de Memoria RAM

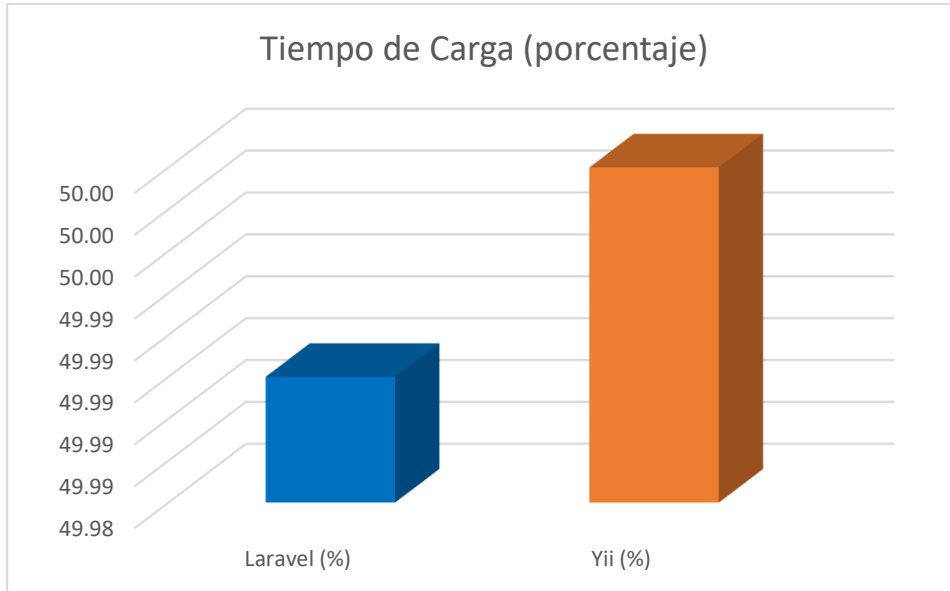
Gráfico 5. Comparación de ambos Framework Laravel Vs Yii



Para el indicador Consumo de Memoria RAM, se pudo observar que el framework Yii hace un menor consumo con un resultado de 6,45 megabytes (Mb) versus unos 8,11 megabytes (Mb) del framework Laravel, demostrando una diferencia 1,6 megabytes (Mb).



### 4.1.3 Tiempo de Carga

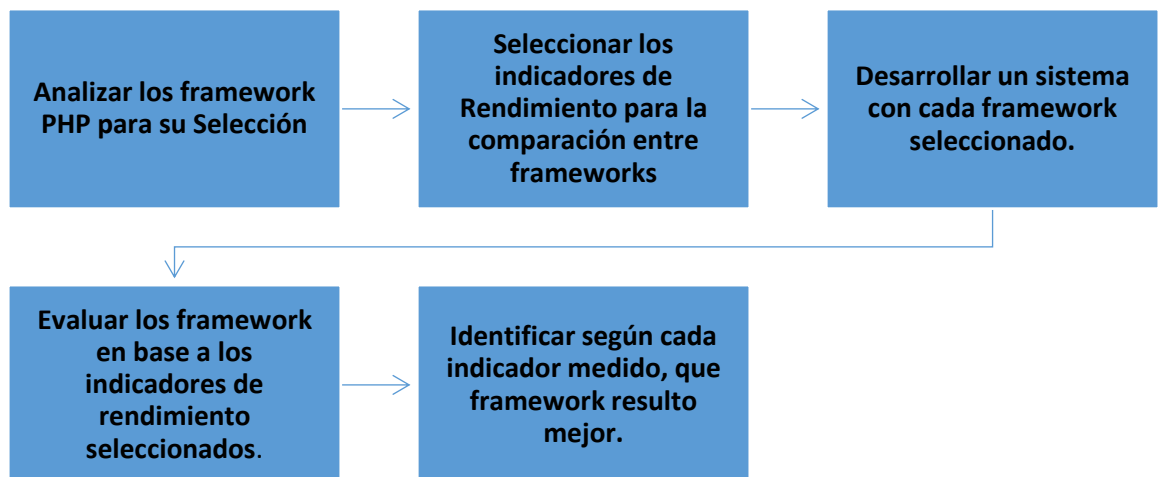


En el tercer indicador se pudo observar la presente gráfica en donde el framework Laravel realiza un menor Tiempo de Carga con un resultado de 149909,46564 microsegundos ( $\mu$ s), versus un 149909,79509 microsegundos ( $\mu$ s) del framework Yii demostrando una diferencia 0,32945 microsegundos ( $\mu$ s).



## CAPÍTULO V: PROPUESTA DE INVESTIGACIÓN

Gráfico 6. Diagrama de Flujo que muestra como es el Proceso de la Propuesta Planteado



Fuente: Elaboración Propia

### 5.1. Analizar los framework PHP para su Selección

Debido a la gran variedad de framework, la elección del framework PHP que mejor se adapte a las necesidad de un proyecto es una tarea complicada, pues todos ofrecen distintas características que mejoran el proceso de desarrollo, esto conlleva a que la elección de un framework PHP se realice de acuerdo a criterios definidos por los desarrolladores, la presente investigación se ha basado en el criterio de popularidad utilizada en la investigación de Boltunova y Prokofyeva, para la selección de los framework PHP se realizó una investigación previa de estudios publicados sobre los TOP 10 framework PHP, en portales web confiables como: OpenWebinars, CodersEye, ArpaTech, DesignsParkle, Apptechmobile.



**CUADRO COMPARATIVO DE VALOR BASADO EN CARACTERISTICAS DE LOS FRAMEWORK PHP**

FRAMEWORK	LARAVEL	CODEIGNITER	FUELPHP	CAKEPHP	SYMFONY	ZEND	PHALCON	YII	SLIM	PHPixie
<b>COMPATIBILIDAD DE BASE DE DATOS</b>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PostgreSQL</li> <li>✓ SQLite</li> <li>✓ SQL Server.</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PostgreSQL</li> <li>✓ SQLite</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PostgreSQL</li> <li>✓ SQLite</li> <li>✓ MongoDB</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ SQLite</li> <li>✓ PostgreSQL</li> <li>✓ SQL Server 2008</li> <li>✓ Oracle</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PostgreSQL</li> <li>✓ Microsoft SQL</li> <li>✓ MongoDB</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PostgreSQL</li> <li>✓ SQL Server</li> <li>✓ SQLite.</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PostgreSQL</li> <li>✓ SQLite</li> <li>✓ MongoDB.</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ MariaDB</li> <li>✓ SQLite</li> <li>✓ PostgreSQL</li> <li>✓ CUBRID</li> <li>✓ Oracle</li> <li>✓ MSSQL</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> </ul>	<ul style="list-style-type: none"> <li>✓ MySQL</li> <li>✓ PHPixie</li> <li>✓ SQLite</li> <li>✓ MongoDB.</li> </ul>
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/database">https://laravel.com/docs/5.4/database</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/database/configuration.html">https://codeigniter.com/user_guide/database/configuration.html</a> )	("FuelPHP", p. <a href="https://fuelphp.com/docs/#/api_docs">https://fuelphp.com/docs/#/api_docs</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3.0/en/orm/database-basics.html#database-configuration">https://book.cakephp.org/3.0/en/orm/database-basics.html#database-configuration</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/doctrine.html">https://symfony.com/doc/current/doctrine.html</a> )	("Zend Framework", n.d.-a, p. <a href="https://docs.zendframework.com/">https://docs.zendframework.com/</a> )	("PhalconPHP", p. <a href="https://phalconphp.com/es/">https://phalconphp.com/es/</a> )	("YiiFramework", p. <a href="http://www.yiiframework.com/doc-2.0/guide-db.html">http://www.yiiframework.com/doc-2.0/guide-db.html</a> )	("Slim", p. <a href="https://www.slimframework.com/docs/tutorial/first.html">https://www.slimframework.com/docs/tutorial/first.html</a> )	("PHPixie", p. <a href="https://phpixie.com/components/database.html">https://phpixie.com/components/database.html</a> )



<b>SEGURIDAD</b>	<ul style="list-style-type: none"> <li>✓ Autenticación</li> <li>✓ API Autenticación</li> <li>✓ Autorización</li> <li>✓ Encriptación</li> <li>✓ Hashing</li> <li>✓ Password Reset</li> </ul>	<ul style="list-style-type: none"> <li>✓ Seguridad URI.</li> <li>✓ Filtrado XSS</li> <li>✓ Protección CSRF</li> <li>✓ Validación dato de entrada</li> </ul>	<ul style="list-style-type: none"> <li>✓ Filtrado de entrada</li> <li>✓ Filtrado URI</li> <li>✓ Filtrado XSS</li> <li>✓ Codificación de Salida</li> <li>✓ Protección de testigos CSRF</li> <li>✓ Prevención de inyección SQL</li> <li>✓ Un marco Auth Seguro</li> </ul>	<ul style="list-style-type: none"> <li>✓ Encriptación.</li> <li>✓ Hashing de datos.</li> <li>✓ Datos Aleatorios Seguros.</li> <li>✓ CSRF</li> <li>✓ Manejo de Devoluciones de llamada Blackhole</li> <li>✓ Restricciones acciones SSL</li> <li>✓ Restricciones comunicación entre controladores</li> <li>✓ Prevención de manipulación de formularios</li> </ul>	<ul style="list-style-type: none"> <li>✓ Autenticación</li> <li>✓ Autorización</li> <li>✓ Encriptación de contraseñas</li> <li>✓ Roles</li> </ul>	<ul style="list-style-type: none"> <li>✓ Autenticación</li> <li>✓ Criptografía</li> <li>✓ Control de acceso</li> <li>✓ Serialización</li> <li>✓ Contenedor de inyección SQL</li> <li>✓ Validación.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Hashing</li> <li>✓ CSRF</li> <li>✓ Generación Aleatoria de datos</li> </ul>	<ul style="list-style-type: none"> <li>✓ Autenticación</li> <li>✓ Autorización</li> <li>✓ Prevención de Cross-site Scripting</li> <li>✓ Prevención de ataque de Cookies</li> <li>✓ Prevención de Falsificación de Solitudes entre Sitios</li> <li>✓ Inicio de sesión basado en Cookies</li> <li>✓ Filtro de control de acceso</li> <li>✓ Control de acceso de funciones</li> </ul>	<ul style="list-style-type: none"> <li>✓ Control de Inyección de dependencia</li> </ul>	<ul style="list-style-type: none"> <li>✓ Criptografía</li> <li>✓ Tokens Aleatorio</li> <li>✓ Entrega segura de cookies</li> </ul>
------------------	---	---	---	---	---	---	--	--	---	---



								✓ Jerarquía de autorización		
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/authentication">https://laravel.com/docs/5.4/authentication</a> )	("CodeIgniter," n.d., p. <a href="https://www.codeigniter.com/user_guide/general/security.html">https://www.codeigniter.com/user_guide/general/security.html</a> )	("FuelPHP," n.d., p. <a href="https://fuelphp.com/features">https://fuelphp.com/features</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3.0/en/security.html">https://book.cakephp.org/3.0/en/security.html</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/security.html">https://symfony.com/doc/current/security.html</a> )	("Zend Framework", p. <a href="https://framework.zend.com/learn">https://framework.zend.com/learn</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2/security">https://docs.phalconphp.com/en/3.2/security</a> )	("Yii", p. <a href="http://www.yiiframework.com/doc/guide/1.1/en/topics/auth">http://www.yiiframework.com/doc/guide/1.1/en/topics/auth</a> )	("Slim", p. <a href="https://www.slimframework.com/">https://www.slimframework.com/</a> )	("PHPixie", p. <a href="https://phpixie.com/">https://phpixie.com/</a> )
<b>VALIDACION INTEGRADA</b>	<ul style="list-style-type: none"> <li>✓ RegisterController</li> <li>✓ Auth</li> <li>✓ Validación de Rutas</li> </ul>	<ul style="list-style-type: none"> <li>✓ Validación datos de entrada</li> </ul>	<ul style="list-style-type: none"> <li>✓ Validación de formularios</li> </ul>	<ul style="list-style-type: none"> <li>✓ Validación de entrada</li> </ul>	<ul style="list-style-type: none"> <li>✓ Uso de Clase Validator</li> </ul>	<ul style="list-style-type: none"> <li>✓ Uso de Clase Validator</li> </ul>	<ul style="list-style-type: none"> <li>✓ Uso de clase Validation</li> </ul>	<ul style="list-style-type: none"> <li>✓ Validación de datos de entrada</li> <li>✓ Permite la creación de clases modelo para validación</li> </ul>	<ul style="list-style-type: none"> <li>✓ Validación propia a cada desarrollador</li> </ul>	<ul style="list-style-type: none"> <li>✓ Validación simple de formularios</li> </ul>



								✓ Validación de controlador		
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/authentication#included-authentication">https://laravel.com/docs/5.4/authentication#included-authentication</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/general/security.html">https://codeigniter.com/user_guide/general/security.html</a> )	("FuelPHP", p. <a href="https://fuelphp.com/features">https://fuelphp.com/features</a> )	("CakePHP", p. <a href="https://cakephp.org/#">https://cakephp.org/#</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/validation.html">https://symfony.com/doc/current/validation.html</a> )	("Zend Framework", p. <a href="https://framework.zend.com/learn">https://framework.zend.com/learn</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2/db-models-validation">https://docs.phalconphp.com/en/3.2/db-models-validation</a> )	("Yii", p. <a href="http://www.yiiframework.com/doc/guide/1.1/en/form-overview">http://www.yiiframework.com/doc/guide/1.1/en/form-overview</a> )	("Slim", p. <a href="https://www.slimframework.com/docs/">https://www.slimframework.com/docs/</a> )	("PHPixie", p. <a href="https://phpixie.com/components.html">https://phpixie.com/components.html</a> )
<b>SISTEMA DE PLANTILLAS</b>	<ul style="list-style-type: none"> <li>✓ Plantillas Blade</li> <li>✓ Herencia de Plantillas</li> <li>✓ Componentes y Ranuras</li> <li>✓ Visualización de datos a través de las plantillas</li> </ul>	✓ Uso de Clase Parser para motor de plantillas.	✓ Uso de Paquete Parser	✓ Uso de plantillas XML	✓ Sistema de Plantillas Twig.	✓ Uso de plantillas XML	✓ Uso de plantillas XML	<ul style="list-style-type: none"> <li>✓ CWebService</li> <li>✓ CWebServiceAction</li> <li>✓ WSDL</li> </ul>	✓ Uso de plantilla Twig-View	✓ Definidas por el mismo desarrollador
<b>RAPIDO Y FLEXIBLES</b>	<ul style="list-style-type: none"> <li>✓ Estructuras de control</li> </ul>									





	<ul style="list-style-type: none"> <li>✓ Incluye Sub – Vistas</li> <li>✓ Láminas de Extensión</li> </ul>									
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/blade">https://laravel.com/docs/5.4/blade</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/libraries/parser.html">https://codeigniter.com/user_guide/libraries/parser.html</a> )	("FuelPHP", p. <a href="https://fuelphp.com/features">https://fuelphp.com/features</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3.0/en/index.html">https://book.cakephp.org/3.0/en/index.html</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/templating.html">https://symfony.com/doc/current/templating.html</a> )	("Zend Framework", p. <a href="https://framework.zend.com/learn">https://framework.zend.com/learn</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2/db-layer">https://docs.phalconphp.com/en/3.2/db-layer</a> )	("YiiFramework", p. <a href="http://www.yiiframework.com/doc/guide/1.1/en/topics/webservice">http://www.yiiframework.com/doc/guide/1.1/en/topics/webservice</a> )	("Slim", p. <a href="https://www.slimframework.com/docs/features/templates.html">https://www.slimframework.com/docs/features/templates.html</a> )	("PHPixie", p. <a href="https://phpixie.com/">https://phpixie.com/</a> )
<b>COMPATIBILIDAD PARA PHP</b>	PHP >= 5.6.4	PHP >= 5.6	PHP = 5.3	PHP >= 5.6	PHP =5.4	PHP >= 5.6	PHP >= 5.5	PHP >= 5.4	PHP 5.5	PHP = 5.3
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4">https://laravel.com/docs/5.4</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/general/requirements.html">https://codeigniter.com/user_guide/general/requirements.html</a> )	("FuelPHP", n.d., p. <a href="https://fuelphp.com/features">https://fuelphp.com/features</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3">https://book.cakephp.org/3</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/">https://symfony.com/doc/current/</a> )	("Zend Framework", p. <a href="https://framework.zend.com/doc">https://framework.zend.com/doc</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2/installation">https://docs.phalconphp.com/en/3.2/installation</a> )	("YiiFramework", p. <a href="http://www.yiiframework.com">http://www.yiiframework.com</a> )	("Slim", p. <a href="https://www.slimframework.com/docs/start/installation">https://www.slimframework.com/docs/start/installation</a> )	("PHPixie", p. <a href="https://phpixie.com/docs/php-environment">https://phpixie.com/docs/php-environment</a> )



				.0/en/index.html)	setup.html)	s.zendframework.com/tutorials/getting-started/overview/)		om/doc-2.0/guide-start-installation.html)	allation.html)	nt-setup.html)
<b>DOCUMENTACION</b>	<ul style="list-style-type: none"> <li>✓ Detallada</li> <li>✓ Entendible</li> <li>✓ Forums</li> <li>✓ Twiter</li> <li>✓ Congresos Online</li> </ul>	<ul style="list-style-type: none"> <li>✓ Detallada</li> <li>✓ Entendible</li> </ul>	<ul style="list-style-type: none"> <li>✓ Muy General</li> <li>✓ Poco Detallada</li> <li>✓ Foro.</li> </ul>	<ul style="list-style-type: none"> <li>✓ Detallada</li> <li>✓ Entendible</li> <li>✓ Comunidad</li> </ul>	<ul style="list-style-type: none"> <li>✓ Muy General</li> <li>✓ Poco detallada</li> <li>✓ Comunidad</li> </ul>	<ul style="list-style-type: none"> <li>✓ Muy General</li> <li>✓ Blog</li> </ul>	<ul style="list-style-type: none"> <li>✓ Muy General</li> <li>✓ Blog</li> </ul>	<ul style="list-style-type: none"> <li>✓ Detallada</li> <li>✓ Libros</li> <li>✓ Tutoriales</li> <li>✓ Forums</li> <li>✓ Chat</li> <li>✓ Wiki</li> <li>✓ Contribución de otros desarrolladores para la mejora del framework</li> </ul>	<ul style="list-style-type: none"> <li>✓ Muy General</li> <li>✓ Blog</li> </ul>	<ul style="list-style-type: none"> <li>✓ Muy General</li> <li>✓ Poco Detallada</li> <li>✓ Blog</li> </ul>
Fuente:	("Laravel," n.d., p. <a href="https://lara">https://lara</a>	("CodeIgniter", p. <a href="https://codeignit">https://codeignit</a>	("FuelPHP", p. <a href="https://fue">https://fue</a>	("Cake PHP", p. <a href="https://s">https://s</a>	("Symfony", p. <a href="https://s">https://s</a>	("Zend Framework", p. <a href="https://docs">https://docs</a>	("PhalconPHP", p. <a href="https://docs">https://docs</a>	("YiiFramework", p. <a href="http://ww">http://ww</a>	("Slim", p. <a href="https://www.slimfr">https://www.slimfr</a>	("PHPi xie", p. <a href="https://">https://</a>



	vel.com/docs/5.4)	er.com/user_guide/)	lphp.com/docs/)	https://book.cakephp.org/3.0/en/index.html)	ymfony.com/doc/current/index.html)	ework," n.d.-a, p. https://framework.zend.com/learn)	phalconphp.com/en/3.2)	w.yiiframework.com/doc/)	amework.com/docs/)	phpixie.com/docs.html)
<b>RUTEO DE URL FLEXIBLE</b>	<ul style="list-style-type: none"> <li>✓ Generación de URLs a rutas asignadas</li> <li>✓ Grupos de rutas</li> <li>✓ Enrutamiento de subdominio</li> <li>✓ Enlaces del modelo de ruta</li> <li>✓ Acceso a la ruta actual</li> </ul>	<ul style="list-style-type: none"> <li>✓ Reglas de enrutamiento estático</li> <li>✓ Rutas Reservadas</li> </ul>	<ul style="list-style-type: none"> <li>✓ Hace uso de la clase Route</li> </ul>	<ul style="list-style-type: none"> <li>✓ Hace uso de la clase Routing</li> </ul>	<ul style="list-style-type: none"> <li>✓ Hace uso de la clase Routing</li> </ul>	<ul style="list-style-type: none"> <li>✓ Hace uso de la clase Routing</li> </ul>	<ul style="list-style-type: none"> <li>✓ Hace uso de la clase Routing</li> </ul>	<ul style="list-style-type: none"> <li>✓ Creación de URLs dinámica</li> <li>✓ Permite definir reglas para URL</li> <li>✓ Permite Parametrizar Rutas</li> <li>✓ Permite agregar sufijos a URL</li> <li>✓ Permite asignar Reglas dinámicamente</li> <li>✓ Normalización de URL</li> </ul>	<ul style="list-style-type: none"> <li>✓ Enrutamiento HTTP</li> </ul>	<ul style="list-style-type: none"> <li>✓ Enrutamiento HTTP</li> </ul>



Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/routing">https://laravel.com/docs/5.4/routing</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/general/routing.html">https://codeigniter.com/user_guide/general/routing.html</a> )	("FuelPHP", p. <a href="https://fuelphp.com/features">https://fuelphp.com/features</a> )	("CakePHP", p. <a href="https://api.cakephp.org/3.4/namespaces/Cake.Routing.html">https://api.cakephp.org/3.4/namespaces/Cake.Routing.html</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/routing.html">https://symfony.com/doc/current/routing.html</a> )	("Zend Framework", p. <a href="https://symfony.com/doc/current/routing.html">https://symfony.com/doc/current/routing.html</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2/routing">https://docs.phalconphp.com/en/3.2/routing</a> )	("YiiFramework", p. <a href="http://www.yiiframework.com/doc-2.0/guide-runtime-routing.html">http://www.yiiframework.com/doc-2.0/guide-runtime-routing.html</a> )	("Slim", p. <a href="https://www.slimframework.com/">https://www.slimframework.com/</a> )	("PHPixie", n.d.-b, p. <a href="https://phpixie.com/components/htpt.html">https://phpixie.com/components/htpt.html</a> )
<b>SCAFFOLDING</b>	<ul style="list-style-type: none"> <li>✓ Scaffolding JavaScript</li> <li>✓ Scaffolding CSS</li> </ul>	✓ No implementa	✓ Scaffolding Básico	✓ No implementa	✓ No implementa	✓ No implementa	✓ No implementa	<ul style="list-style-type: none"> <li>✓ Generación de Class Active Record</li> <li>✓ Generación de CRUD</li> <li>✓ Generador de Modulo</li> <li>✓ Generador de Controller</li> <li>✓ Generador de</li> </ul>	✓ No implementa	✓ No implementa



								Extensi ón		
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/frontend">https://laravel.com/docs/5.4/frontend</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/">https://codeigniter.com/user_guide/</a> )	("FuelPHP," n.d., p. <a href="https://fuelphp.com/features">https://fuelphp.com/features</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3.0/en/index.html">https://book.cakephp.org/3.0/en/index.html</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/index.html">https://symfony.com/doc/current/index.html</a> )	("Zend Framework", p. <a href="https://framework.zend.com/learn">https://framework.zend.com/learn</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2">https://docs.phalconphp.com/en/3.2</a> )	("YiiFramework", p. <a href="http://www.yiiframework.com/doc-2.0/guide-start.html">http://www.yiiframework.com/doc-2.0/guide-start.html</a> )	("Slim", p. <a href="https://www.slimframework.com/docs/">https://www.slimframework.com/docs/</a> )	("PHPixie", p. <a href="https://phpixie.com/docs.html">https://phpixie.com/docs.html</a> )
<b>MAPEO DE OBJETOS RELACIONAL - ORM</b>	✓ Si implementa (Eloquent, Relationships, Collections, Mutators, Serialization)	✓ No implementa	✓ No implementa	✓ Si implementa	✓ Si implementa	✓ No implementa	✓ Si implementa	✓ Si implementa (Active Record)	✓ No implementa	✓ No implementa
Fuente:	("Laravel", p. <a href="https://laravel.com/docs/5.4/eloquent">https://laravel.com/docs/5.4/eloquent</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/">https://codeigniter.com/user_guide/</a> )	("FuelPHP", p. <a href="https://fuelphp.com/docs/">https://fuelphp.com/docs/</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3.0/en/orm.html">https://book.cakephp.org/3.0/en/orm.html</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/doctrine.html">https://symfony.com/doc/current/doctrine.html</a> )	("Zend Framework", p. <a href="https://framework.zend.com/manual/2.3/en/user-guide/database-and-">https://framework.zend.com/manual/2.3/en/user-guide/database-and-</a>	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2">https://docs.phalconphp.com/en/3.2</a> )	("YiiFramework", p. <a href="http://www.yiiframework.com/doc-2.0/guide-index.html">http://www.yiiframework.com/doc-2.0/guide-index.html</a> )	("Slim", p. <a href="https://www.slimframework.com/">https://www.slimframework.com/</a> )	("PHPixie", p. <a href="https://phpixie.com/docs.html">https://phpixie.com/docs.html</a> )



						models.html)				
<b>FACIL LECTURA DE CODIGO</b>	✓ Entendible	✓ Entendible	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Difícil de Comprender	✓ Entendible	✓ Entendible	✓ Difícil de Comprender	✓ Difícil de Comprender
Fuente:	("Laravel," n.d., p. <a href="https://laravel.com/docs/5.4/views#creating-views">https://laravel.com/docs/5.4/views#creating-views</a> )	("CodeIgniter", p. <a href="https://codeigniter.com/user_guide/general/views.html">https://codeigniter.com/user_guide/general/views.html</a> )	("FuelPHP", p. <a href="https://fuelphp.com/docs/general/packages.html">https://fuelphp.com/docs/general/packages.html</a> )	("CakePHP", p. <a href="https://book.cakephp.org/3.0/en/and-examples/bookmarks/intro.html">https://book.cakephp.org/3.0/en/and-examples/bookmarks/intro.html</a> )	("Symfony", p. <a href="https://symfony.com/doc/current/templating.html">https://symfony.com/doc/current/templating.html</a> )	("Zend Framework", p. <a href="https://framework.zend.com/learn">https://framework.zend.com/learn</a> )	("PhalconPHP", p. <a href="https://docs.phalconphp.com/en/3.2/views">https://docs.phalconphp.com/en/3.2/views</a> )	("YiiFramework", n.d., p. <a href="http://www.yiiframework.com/doc-2.0/guide-structure-views.html#layouts">http://www.yiiframework.com/doc-2.0/guide-structure-views.html#layouts</a> )	("Slim," n.d.-b, p. <a href="https://www.slimframework.com/docs/">https://www.slimframework.com/docs/</a> )	("PHPixie", p. <a href="https://phpixie.com/components/template.html">https://phpixie.com/components/template.html</a> )

Tabla 2. Tabla de Niveles con Puntaje

Nivel	Puntaje
Excelente	3
Regular	2
Pésimo	1



Tabla 3. Puntaje a cada Framework según los criterios establecidos

FRAMEWORK	LARAVEL	CODEIGNITER	FUELPHP	CAKEPHP	SYMFONY	ZEND	PHALCON	YII	SLIM	PHPixie
COMPATIBILIDAD DE BASE DE DATOS	3	2	3	3	3	3	3	3	1	3
SEGURIDAD	3	2	3	3	2	3	2	3	1	2
VALIDACION INTEGRADA	3	1	1	1	1	1	1	3	1	1
SISTEMA DE PLANTILLAS RAPIDO Y FLEXIBLES	3	1	1	1	1	1	1	3	1	1
COMPATIBILIDAD PHP	1	2	3	2	2	2	2	2	2	3
DOCUMENTACION	3	1	2	2	2	1	1	3	1	2
RUTEO DE URL FLEXIBLE	3	2	1	1	1	1	1	3	1	1



<b>SCAFFOLDING</b>	3	1	2	1	1	1	1	3	1	1
<b>MAPEO DE OBJETOS RELACIONAL - ORM</b>	3	1	1	2	2	1	2	3	1	1
<b>FACIL LECTURA DE CODIGO</b>	3	3	1	1	1	1	3	3	1	1
<b>TOTAL</b>	<b>28</b>	16	18	17	16	15	17	<b>29</b>	11	16



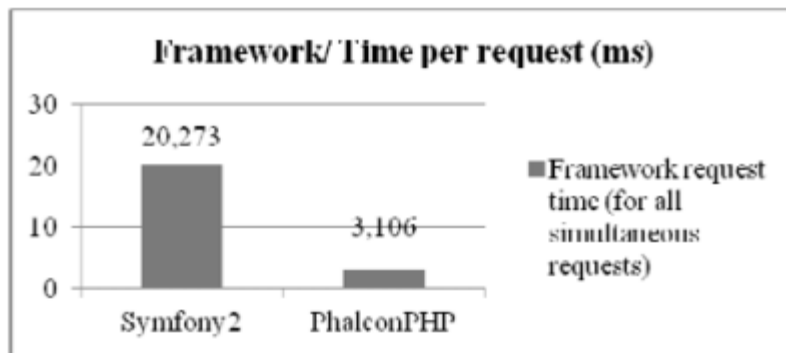


De los criterios evaluados con los puntajes correspondientes y usando el instrumento de validación Juicio de Expertos **Anexo 1** ,se obtiene como resultado la elección de los dos framework PHP con más alto puntaje los cuales son Yii y Laravel.

### 5.2 Seleccionar los indicadores de Rendimiento para la comparación entre frameworks

Para la selección de los indicadores de rendimiento se tomó como base la investigación de "Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems" (Prokofyeva & Boltunova, 2017) en la cual se realizó una comparación de dos framework Symfony2 y PhalconPHP para medir el rendimiento y eficiencia, de ahí se pudo extraer los indicadores Cantidad de Memoria RAM, Tiempo de Consulta a la Base de Datos y la Cantidad de Archivos Cargados por Solicitud que serán de utilidad en la presente investigación.

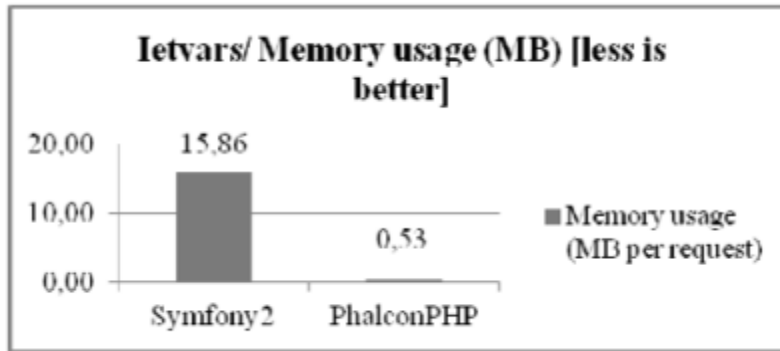
Figura 9.Comparación de Tiempo de Consulta



Fuente. (Prokofyeva & Boltunova, 2017)

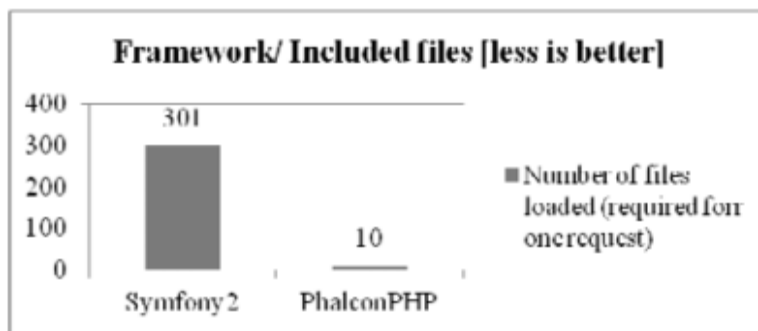


Figura 10. Consumo de memoria del marco



Fuente. (Prokofyeva & Boltunova, 2017)

Figura 11. Número de archivos cargados



Fuente. (Prokofyeva & Boltunova, 2017)

### 5.3. Desarrollar un sistema con cada framework seleccionado

Para poder comprobar y medir los indicadores del rendimiento, en los framework seleccionados es necesario realizar un caso práctico, para lo cual se ha desarrollado un sistema web de ventas SisVentas, desarrollado con el framework Yii y Laravel, para el desarrollo se ha usado la metodología SCRUM.



### 5.3.1 Metodología SCRUM

#### 5.3.1.1 Descripción del Sistema Propuesto

Se ha realizado el Sistema de Ventas - SISVENTAS que permite llevar el control de productos, realiza el registros de ventas, permite registrar las compras también permite gestionar el almacén y realiza el mantenimiento correspondiente.

El sistema está integrado por los siguientes módulos: Almacén, Compras, Ventas y Mantenimiento.

#### 5.3.1.2 Personas y Roles del Proyecto

En el desarrollo del proyecto los integrantes del equipo de trabajo estuvo conformado por 2 personas con los siguientes datos y roles detallados en la siguiente Tabla

Tabla 4. Personas y Roles Scrum

Persona	Contacto	Rol
Gino Paul Gonzales Custodio	gcustodio@crece.uss.edu.pe	Product Owner
Pedro Zuloeta Quispe	pezuqui@outlook.com	Scrum Master
Luis Humberto Llatas Yovera	lyoveraluis@crece.uss.edu.pe	Scrum Team
Alex Jhampier Rojas Herrera	rherreraalex@crece.uss.edu.pe	Scrum Team



### 5.3.1.3 Historias de Usuario

Las historias de usuario (HU) se desarrollan en la ejecución de cada sprint, de acuerdo a la planificación. Las historias de usuario son aquellas que dan a conocer la funcionalidad que va a tener el sistema para su posterior implementación.

Para establecer la prioridad de los requerimientos se estableció un rango entre 1 y 10, considerando a 1 como menor prioridad y 10 como mayor prioridad; esto considerado según la importancia de su desarrollo.

Prioridad	Rango
Alta	8 - 10
Medio	5 - 7
Bajo	3 - 4
Muy Bajo	1 - 2

Para los puntos estimados se ha empleado la técnica de Planning Poker que permite realizar una estimación inicial del proyecto rápida y fiable, cuyo objetivo es obtener una medida de tamaño relativa de todas las historias respecto a sí mismas considerando la serie de posibles valores [1, 2, 3, 5, 8, 13, 21, 34, 55, 89] y así como para los puntos estimados [1 equivale a 1 hora de trabajo].

En la tapa de planificación del sistema es donde se realiza la organización, estimación y se da la prioridad a las historias de usuario, estos procesos son conocidos como Product Backlog en la metodología SCRUM, así como se describe en la siguiente Tabla.



### 5.3.1.4 Product Backlog

<b>Id</b>	<b>Descripción del Requerimiento</b>	<b>Prioridad</b>	<b>Estimación</b>
HT-01	Instalación de herramientas de desarrollo	<b>10</b>	<b>8</b>
HT-02	Diseño de la Base de Datos	<b>10</b>	<b>19</b>
HT-03	Diseño de Interfaces	<b>10</b>	<b>16</b>
HU-04	Determinar Permisos y Roles	<b>10</b>	<b>18</b>
HU-05	Debe permitir el mantenimiento de Usuarios	<b>10</b>	<b>21</b>
HU-06	Debe permitir el mantenimiento de Ventas	<b>10</b>	<b>25</b>
HU-07	Debe permitir el mantenimiento Producto	<b>9</b>	<b>21</b>
HU-08	Debe permitir el mantenimiento de Cliente	<b>9</b>	<b>21</b>
HU-09	Debe permitir el mantenimiento de Compra	<b>9</b>	<b>24</b>
HU-10	Debe permitir el mantenimiento de Proveedor	<b>9</b>	<b>21</b>
HU-11	Debe permitir el mantenimiento de Almacén	<b>9</b>	<b>21</b>
HU-12	Debe permitir el mantenimiento de Trabajador	<b>9</b>	<b>21</b>



### 5.3.1.5 Planificación SCRUM

La planificación de un proyecto de software es importante ya que es la base para el desarrollo, además se debe estimar recursos técnicos, financieros, así como tiempos y recurso humano todo esto es necesario para el desarrollo y funcionamiento de aplicaciones de software.

### 5.3.1.6 Reuniones SCRUM

Utilizando la metodología SCRUM se logró establecer los tiempos de desarrollo, así como la planificación de los entregables que se deben mostrar al finalizar cada Sprint.

Además se realizaron diversas reuniones entre el dueño del producto y el equipo de desarrollo con la finalidad de establecer ciertas tareas o historias de usuario para cada sprint.

Al finalizar cada sprint se revisa el entregable funcional entre los miembros del desarrollo y terceros, así como también el análisis de los procesos aplicados en base a la metodología, estableciendo si son adecuados o si necesitan ser mejorados.

Finalmente se estableció que el proyecto se desarrollará en un rango de 5 a 10 horas diarias, teniendo como fecha de inicio 02/05/2017 y fecha de finalización 30/06/2017.

En todas las reuniones realizadas se contó con la presencia del dueño del proyecto y los dos desarrolladores.

### 5.3.1.7 Desarrollo de Sprint del Proyecto

En el desarrollo se consideró 13 sprint, de los cuales tenemos 10 entregables de software funcional y 3 soluciones técnicas para su implantación y funcionamiento en plena producción; cada sprint tiene un esfuerzo total en puntos de historias,

los cuales están basados en la suma de estos puntos por cada HU o HT desarrollada en el sprint. Se ha establecido una duración de 61 horas en el primer Sprint y 209 horas en el segundo sprint que corresponde a dos meses de trabajo.

### 5.3.1.8 Sprints Backlog

ID	Descripción	Fecha de Inicio	Fecha Final
SP1	Análisis y Diseño	02/05/2017	29/05/2017
SP2	Gestión de Mantenimiento de Módulos para el Sistema de Ventas	30/05/2017	30/06/2017

#### Sprints 1 Análisis y Diseño

Se obtuvo la información de las reuniones realizadas con el cliente y el equipo de trabajo, de la cual se tuvo como resultado los diferentes requisitos y prioridades que posee el sistema necesario para la realización de la aplicación web de ventas, como son las herramientas de desarrollo, el diseño de base de datos y también el diseño de la interfaz de usuario.

Sprint 1				
<b>Inicio:</b> 02/05/2017	<b>Fin:</b> 29/05/2017	<b>Esfuerzo Estimado:</b> 61 h	<b>Esfuerzo Real:</b> 65 h	
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo Estimado (horas)	Tipo	Responsable



HT-01	Para desarrollar necesito la instalación de herramientas de desarrollo	8	Análisis	Alex/Luis
HT-02	Diseño de la Base de Datos	19	Diseño	Alex
HT-03	Diseño de Interfaz de Usuario	16	Diseño	Luis

### Herramientas de Software para el Desarrollo Web

Tabla 5.Requerimientos del Software utilizado para la Programación

Requerimientos de Software	
Nombre	Versión
Editor de texto SublimeText 3	3126
Apache	2.4.25
Composer	1.4.2
MySQL	5.2.44
PHP	5.6.30
Framework Yii 2	2.0.12
Framework Laravel	5.4.23





**Hardware de Desarrollo**

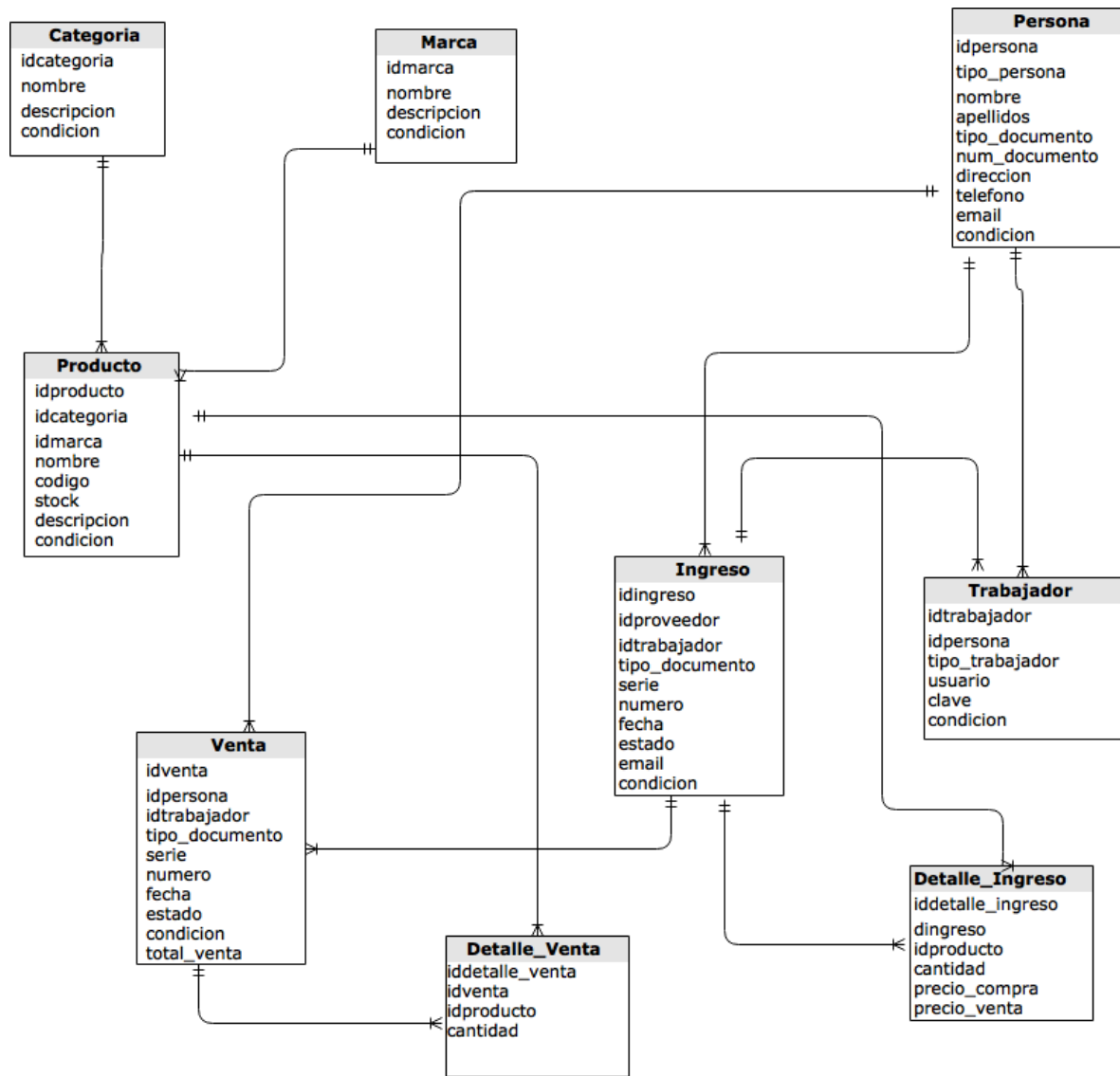
Tabla 6. Hardware utilizado para la Programación

<b>Hardware Utilizado</b>			
<b>Modelo</b>	<b>Descripción</b>	<b>Sistema Operativo</b>	<b>Memoria RAM</b>
<b>LENOVO 20354</b>	Intel(R) Core(TM)i7-4510u CPU @ 2.00GHz (4CPUs)	Windows 8.1	16 RAM
<b>Toshiba Satellite L745</b>	Intel(R) Core(TM)i5-4230M CPU @ 2.40 GHz (4 CPUs), ~2.4GHz	Windows 7 Ultimate 64 bits	4GB



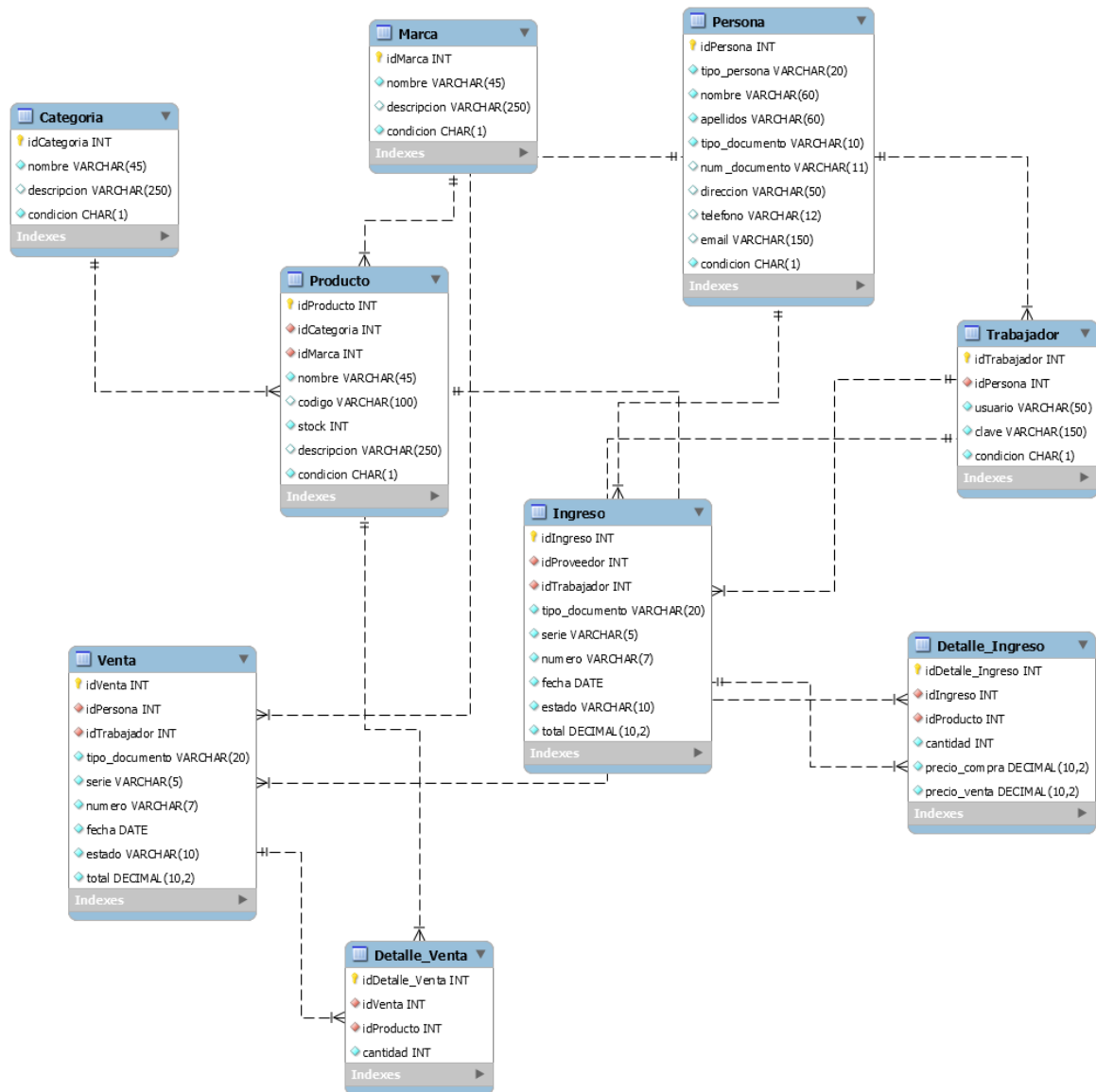
## Diseño de la Base de Datos Modelo Conceptual

Figura 12. Modelo Conceptual de la Base de Datos



### Modelo Físico

Figura 13. Modelo Físico en MySQL



## Diseño de la Interfaz gráfica del usuario

### Interfaz de para el Framework Laravel

#### Login



The image shows a login form interface within a window. The window has a title bar with a close button (X) in the top right corner. The form itself is titled "Login" in blue text. It contains two input fields: "Usuario" with the value "lilian@gmail.com" and "Clave" with masked characters ".....". Below the password field is a checkbox labeled "Recordar". At the bottom of the form, there is a blue button labeled "Iniciar Sesión" and a link that says "¿Olvidaste tu contraseña?".

Interfaz Principal de Ventas con el framework Laravel

**SISTEMA DE VENTAS**

VENTAS

rperezana@gmail.com

Mantenimiento

- Categoría
- Marca
- Producto
- Cilientes y Proveedores
- Trabajadores

Ingreso

- Compras
- Productos

Ventas

- Ventas

Cerrar Sesión

LISTADO DE VENTAS

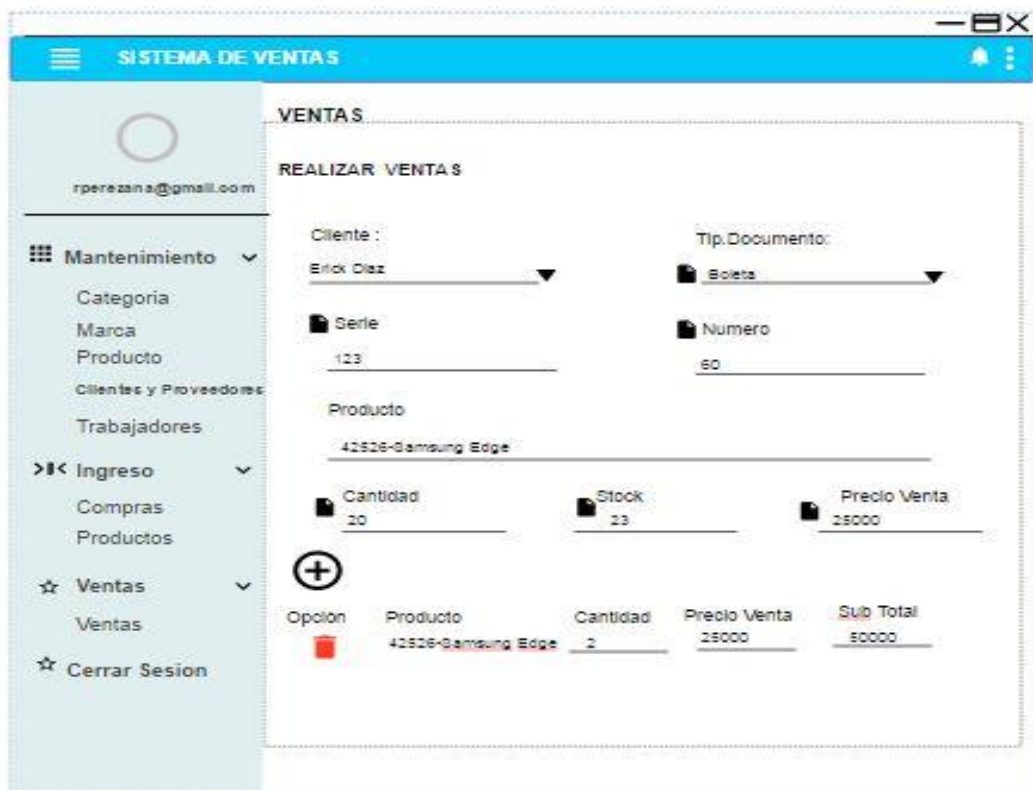
Buscar : \_\_\_\_\_

#	Ciliente	T.Documento	Señe	Número	Fecha	Total	Estado	Opciones
1	Aly Rojas	Boleta	147	30	2017-06-07 10:13:30	160.00	Anulado	
2	Lorena Diaz	Factura	148	31	2017-06-08 12:43:36	120.00	Aceptado	
3	Eleana Perez	Boleta	149	32	2017-06-09 08:13:20	100.00	Aceptado	
4	Karina Wong	Factura	150	33	2017-06-10 02:04:10	80.00	Aceptado	

ANTERIOR 1 SIGUIENTE



### Interfaz de Realizar una Venta



## Interfaz de para el Framework Yii

### ➤ Login

### ➤ Interfaz Principal de Ventas con el framework Yii

Cliente	T.Documento	Serie	Número	Fecha	Total	Estado	Opciones
Alex	Boleta	156	9939	2017-05-05 20:00:00	158.00	Aceptado	Ver
Karen	Factura	100	9939	2017-06-17 23:50:12	149.00	Aceptado	Ver



➤ Interfaz de Realizar una Venta

Yii 2.0    Categorías    Marcas    Clientes y/o Proveedores    Crear Usuarios    Productos    Ingresos    Ventas    Login

[Ir a Lista de Ventas](#)

### Ventas

Proveedor

Tipo de Documentado     Serie     Número

Producto     Cantidad     Stock

Precio de Venta     [Agregar](#)

Opción	Producto	Cantidad	P.Venta	Sub Total
<input type="checkbox"/>	TV Samsung	2	200.00	400.00
<input type="checkbox"/>	Reproductor mp3	6	300.00	1800.00
Total				18400.00

[Guardar](#)





**Sprint 2 Gestión de Mantenimiento de Módulos para el Sistema de Ventas**

Sprint 2				
<b>Inicio:</b> 02/05/2017	<b>Fin:</b> 29/05/2017	<b>Esfuerzo Estimado:</b> 209 h	<b>Esfuerzo Real:</b> 215 h	
Pila del Sprint				
Backlog ID	Descripción	Esfuerzo Estimado (horas)	Tipo	Responsable
HU-04	Determinar Permisos y Roles	18	Programación	Alex/Luis
HU-05	Debe permitir el mantenimiento de Usuarios	21	Programación	Alex/Luis
HU-06	Debe permitir el mantenimiento de Ventas	25	Programación	Alex/Luis
HU-07	Debe permitir el mantenimiento Producto	21	Programación	Alex/Luis
HU-08	Debe permitir el mantenimiento de Cliente	21	Programación	Alex/Luis



HU-09	Debe permitir el mantenimiento de Compra	24	Programación	Alex/Luis
HU-10	Debe permitir el mantenimiento de Proveedor	21	Programación	Alex/Luis
HU-11	Debe permitir el mantenimiento de Almacén	21	Programación	Alex/Luis
HU-12	Debe permitir el mantenimiento de Trabajador	21	Programación	Alex/Luis

En este sprint se desarrolló la codificación de los siguientes módulos: Almacén, Compras, Ventas y Mantenimiento, los cuales son detalladas a continuación en la Tabla.



## Implementación con Framework Laravel

Laravel, propone una forma de desarrollar aplicaciones web de un modo mucho más ágil, con el framework Laravel podemos usar el patrón de diseño MVC (Modelo-Vista-Controlador), donde al igual que otros frameworks PHP, el controlador es programado como una clase.

Por lo tanto, un Controlador es una clase PHP que dispone de métodos públicos que son el punto de entrada final de una petición HTTP (Request PHP) a nuestra aplicación.

## Resultados del Sistema de Ventas -SISVENT Realizado con el Framework Laravel

Figura 14. Pantalla de Logueo

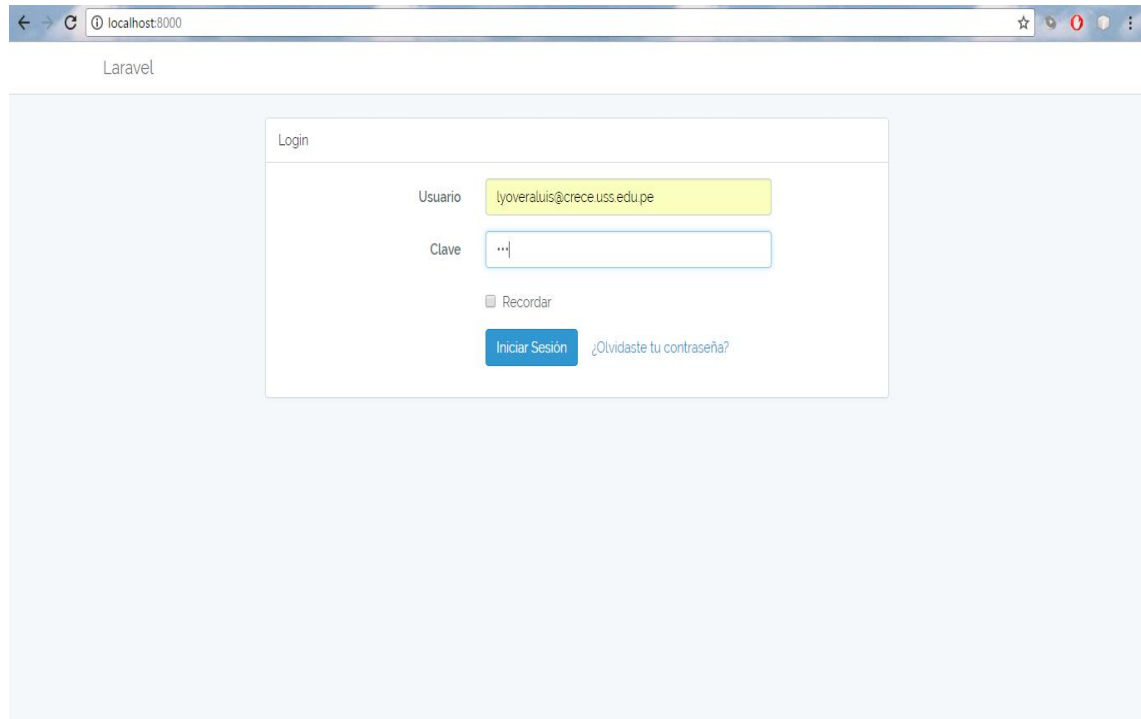


Figura 15. Pantalla Categorías

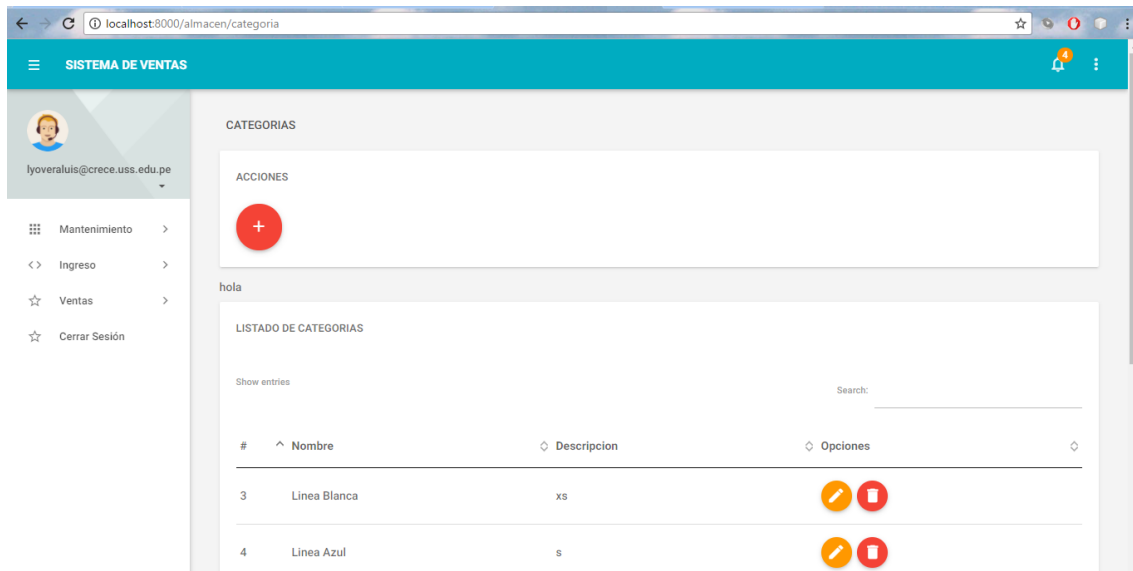


Figura 16. Pantalla Agregar una Nueva Categoría

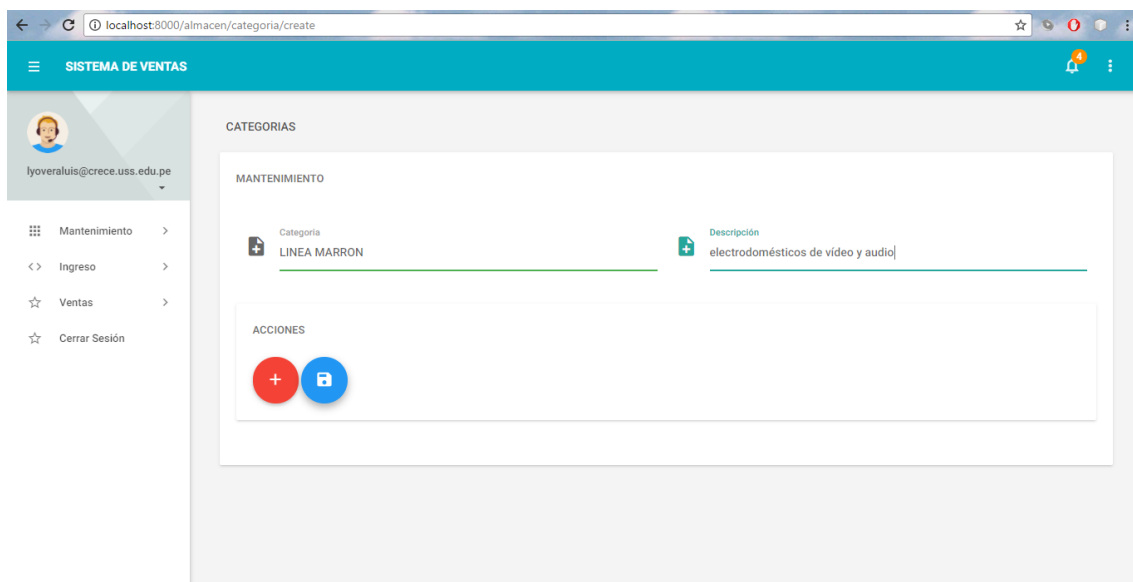


Figura 17. Pantalla Editar Categorías



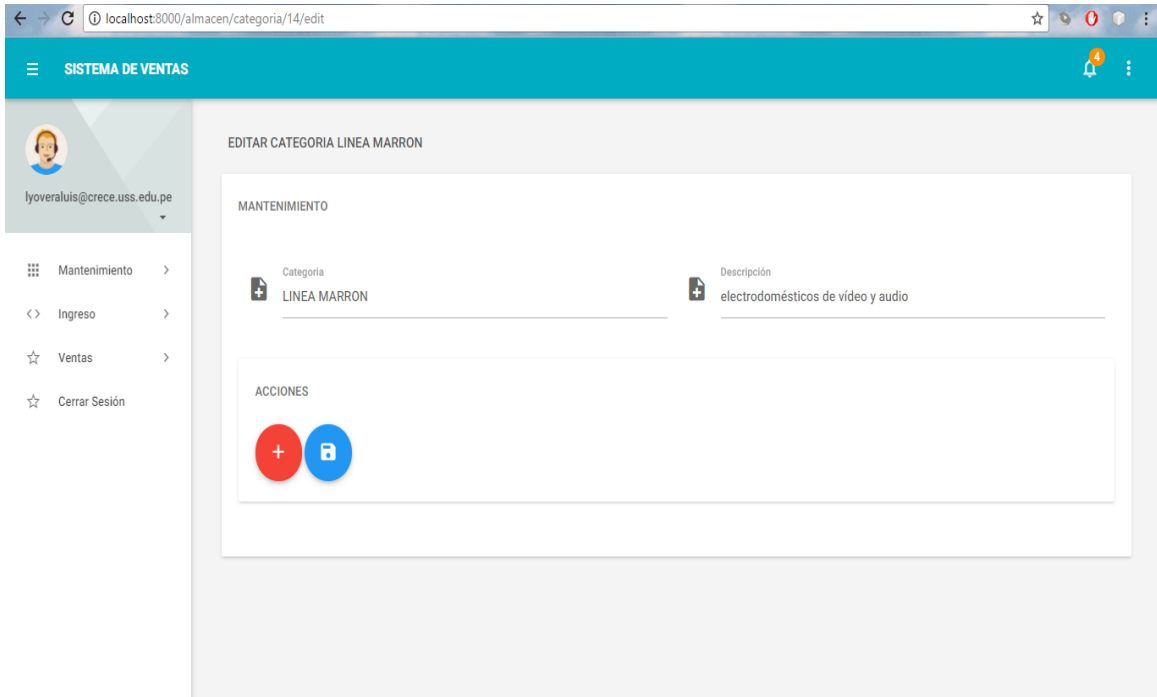


Figura 18. Pantalla Eliminar Categorías

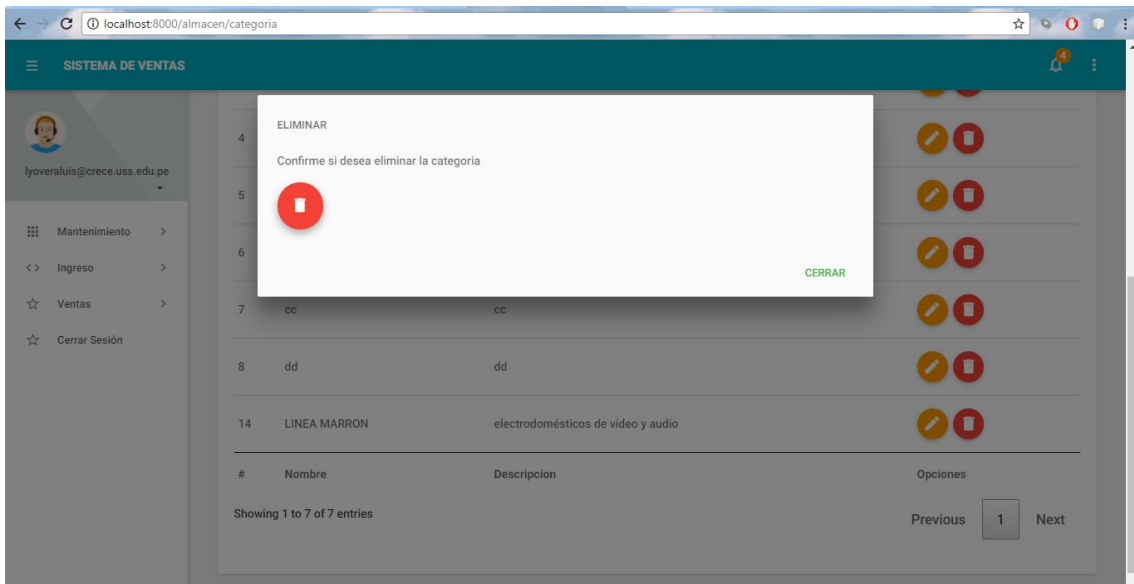


Figura 19. Pantalla Marcas

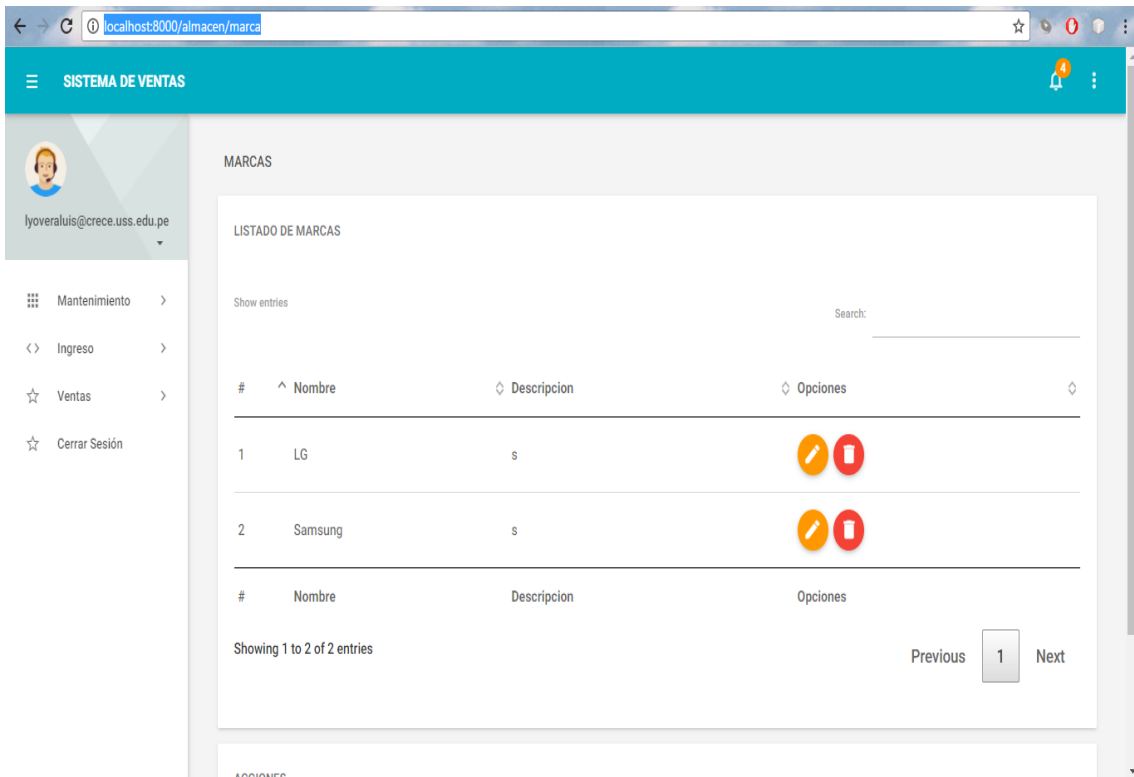


Figura 20. Pantalla Listar Marca

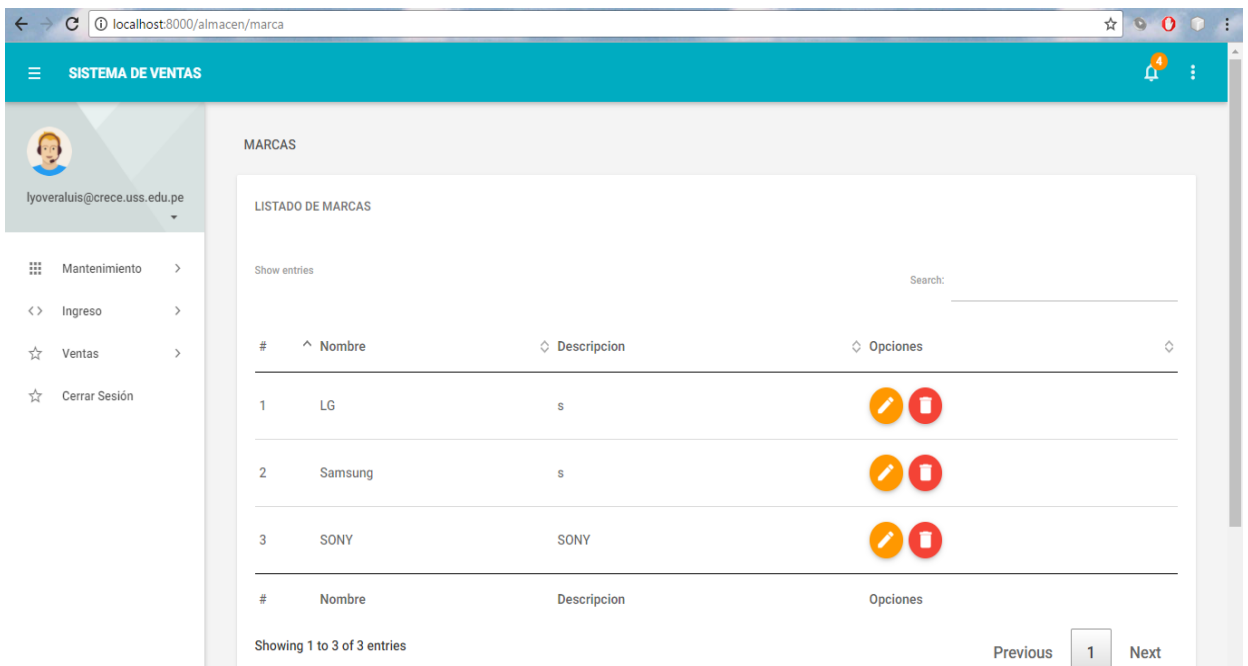


Figura 21. Pantalla Productos

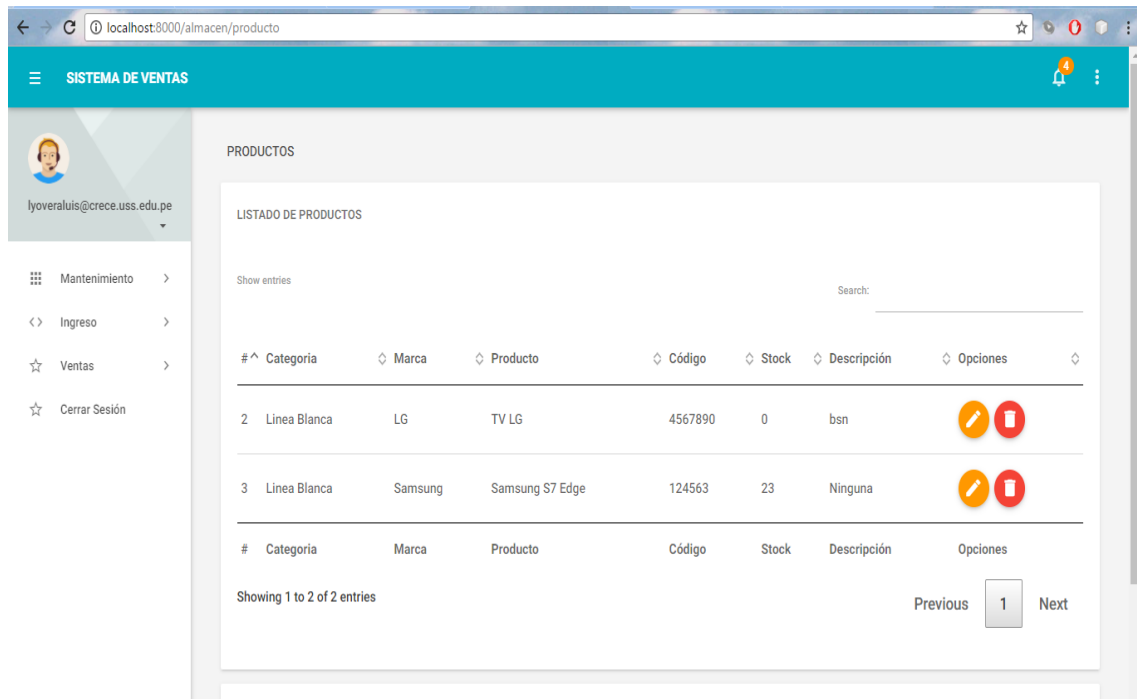


Figura 22. Pantalla Agregar Producto

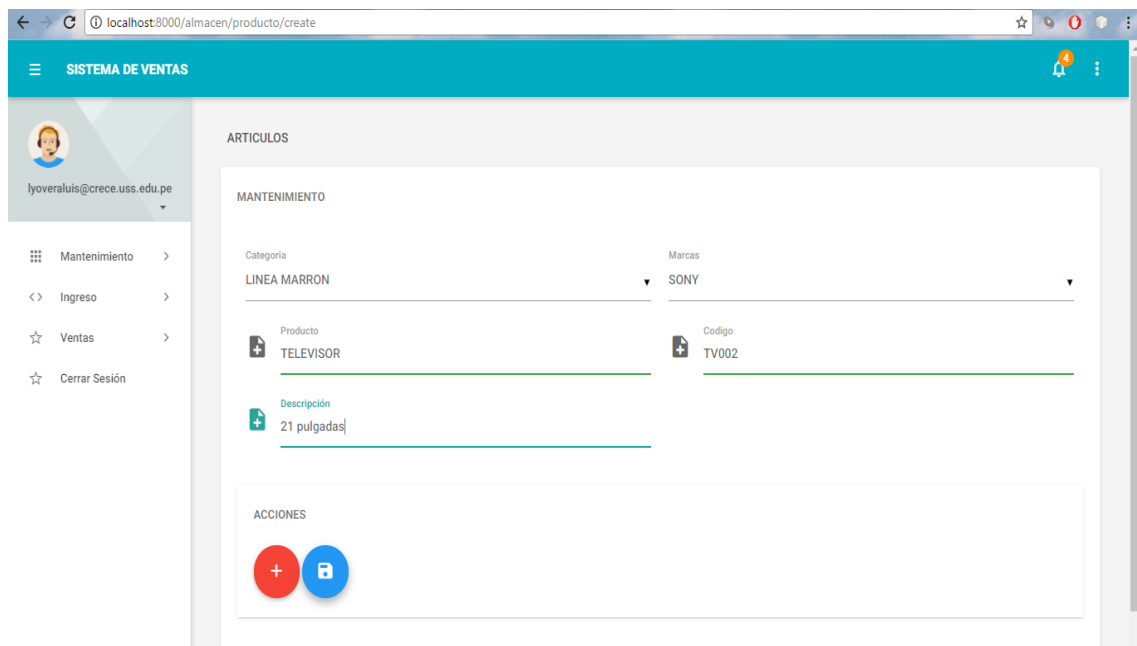




Figura 23. Pantalla Clientes y Proveedores

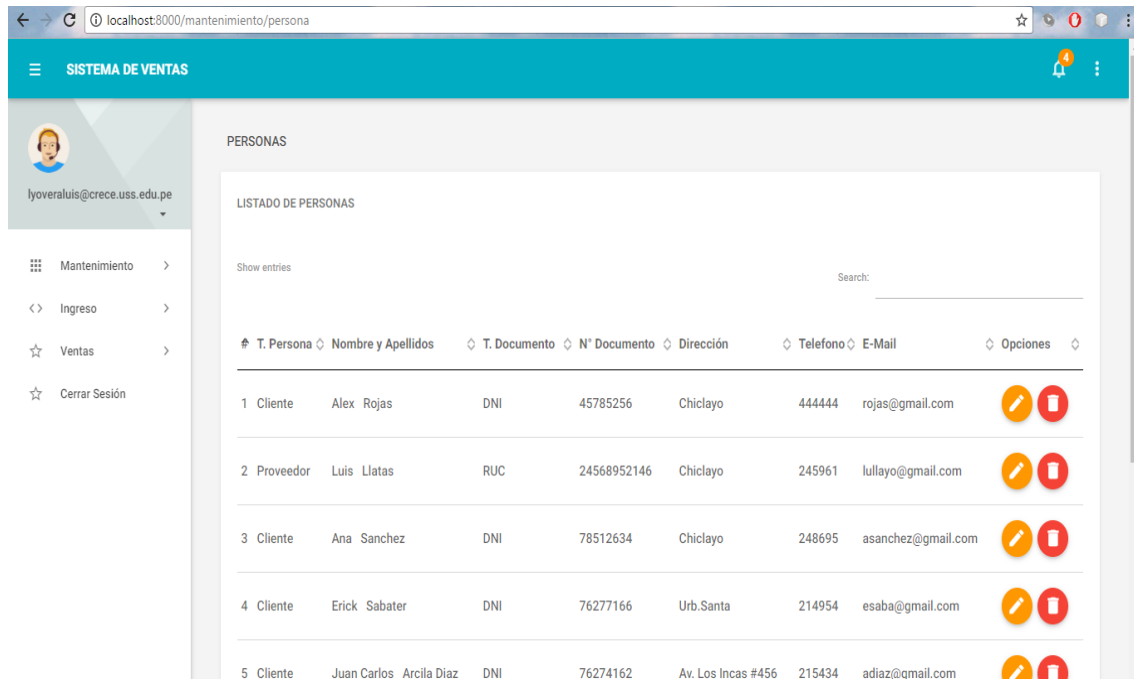


Figura 24. Pantalla Listar Clientes

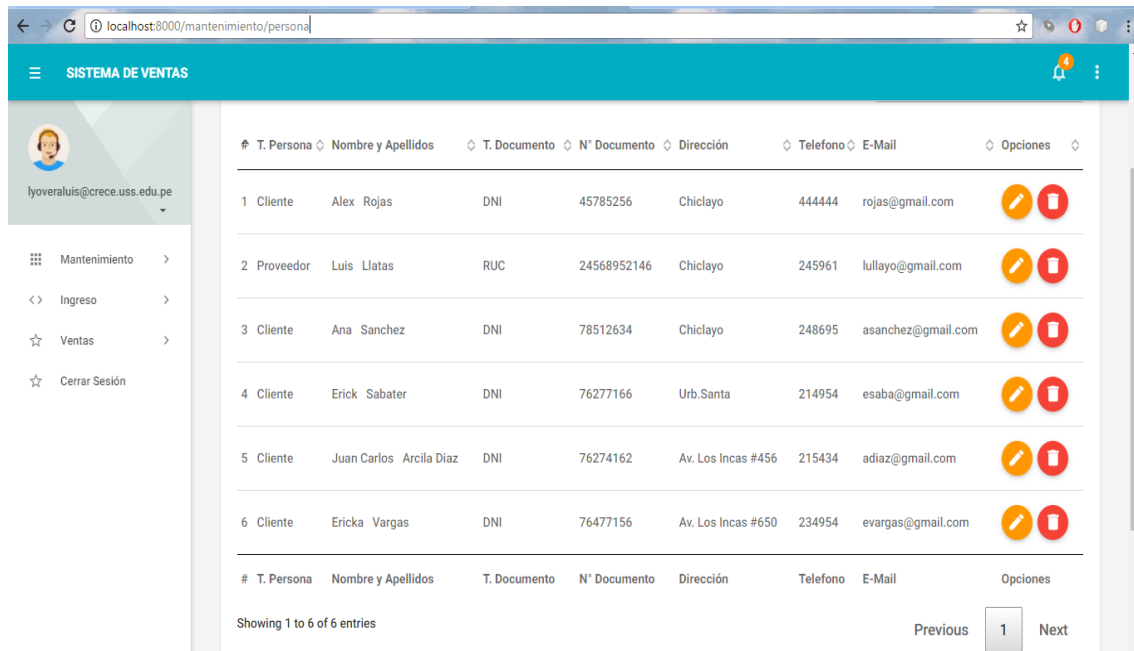


Figura 25. Pantalla Agregar Proveedor

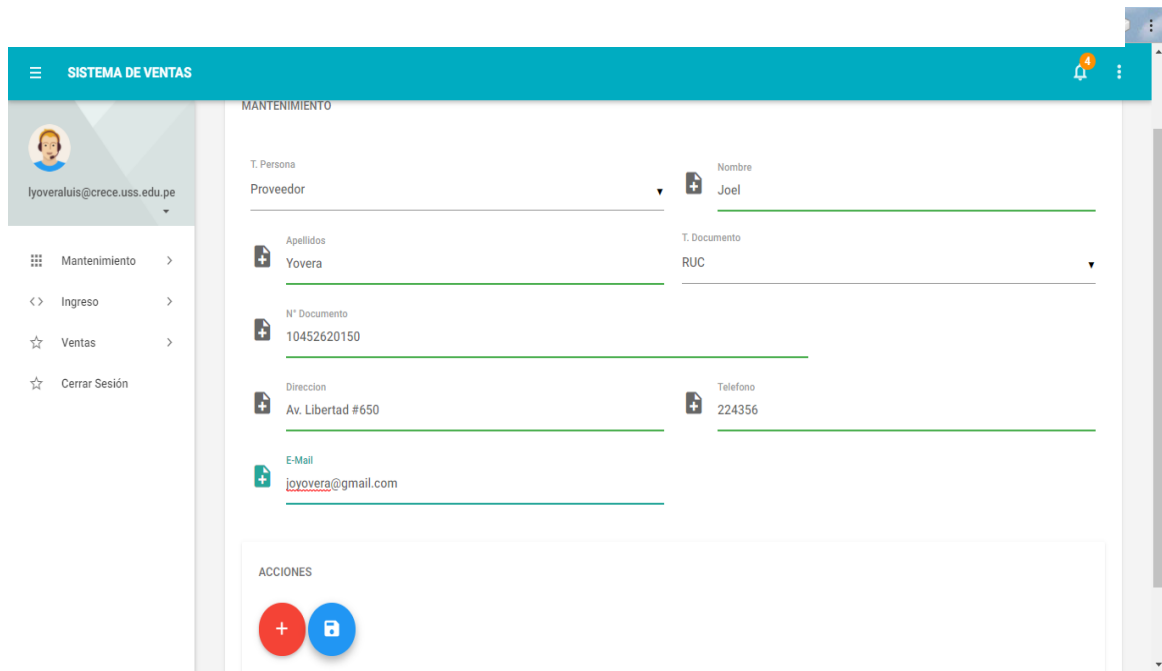


Figura 26. Pantalla Trabajadores

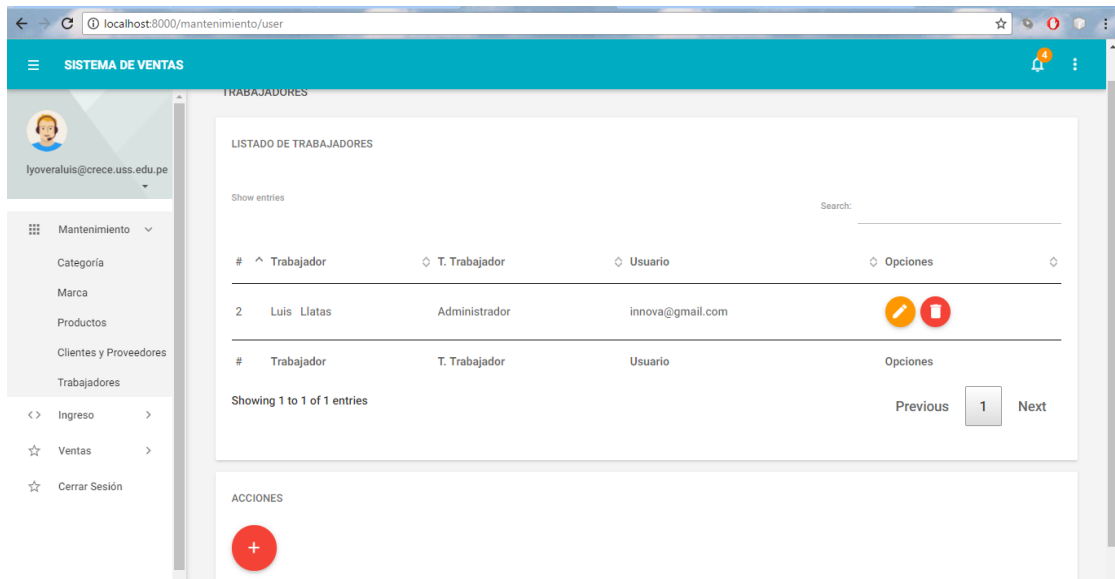


Figura 27. Pantalla Agregar Nuevo Trabajador

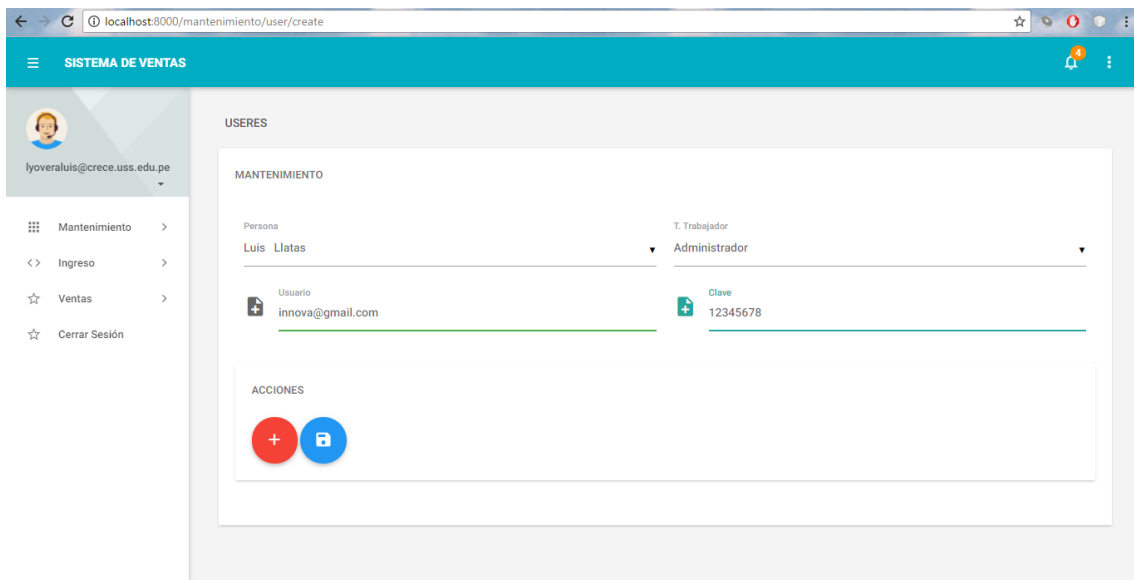


Figura 28. Pantalla Editar Trabajador

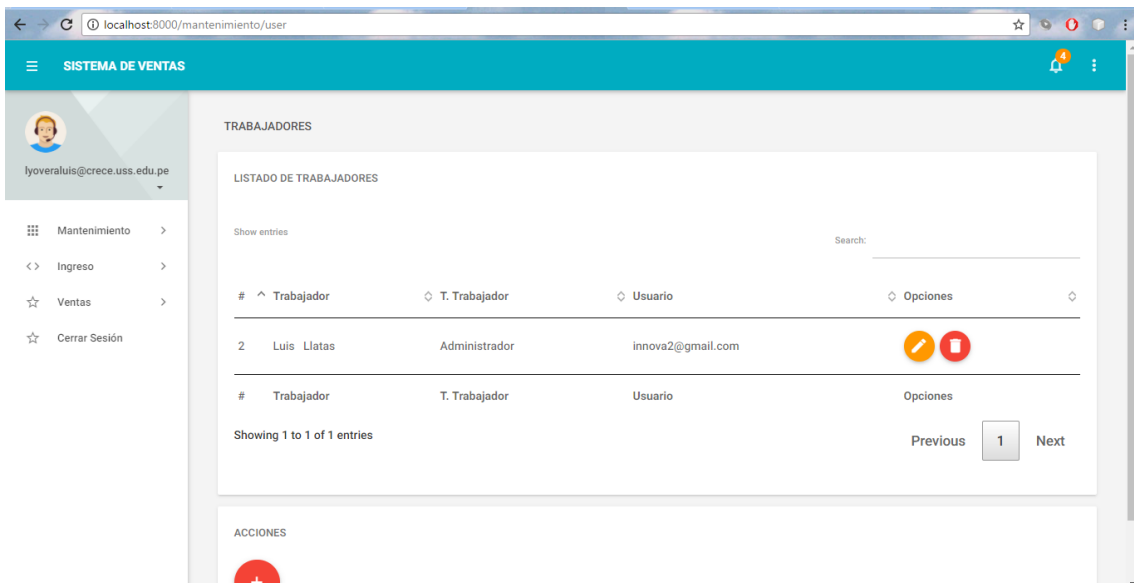


Figura 29. Pantalla Ingreso-Compras

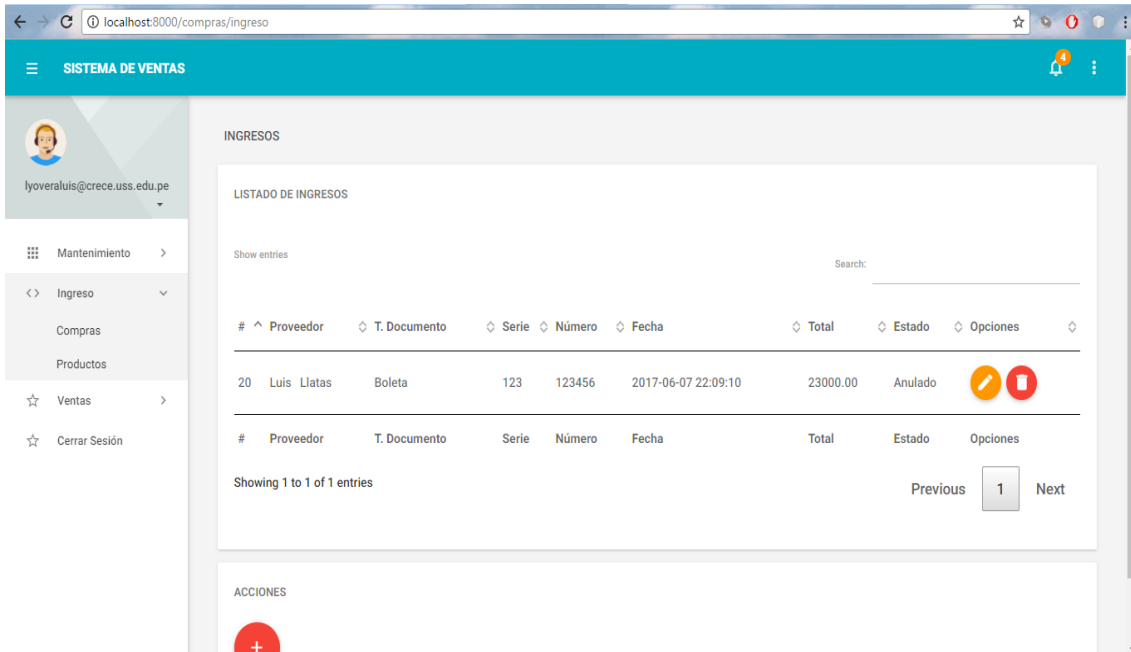


Figura 30. Pantalla Agregar Ingreso-Compra

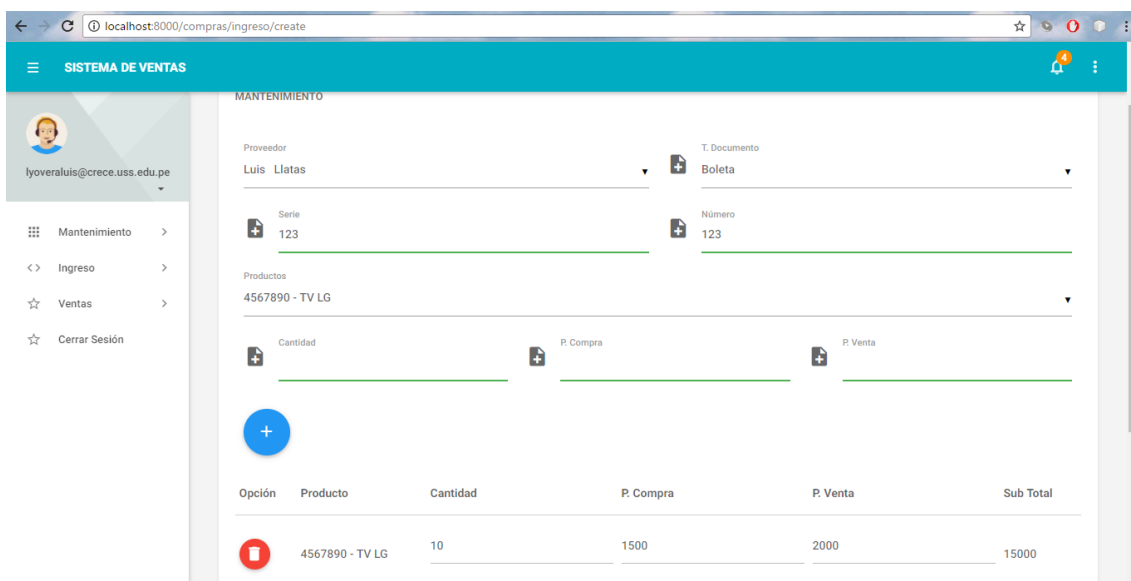


Figura 31. Pantalla Listado de Ingresos

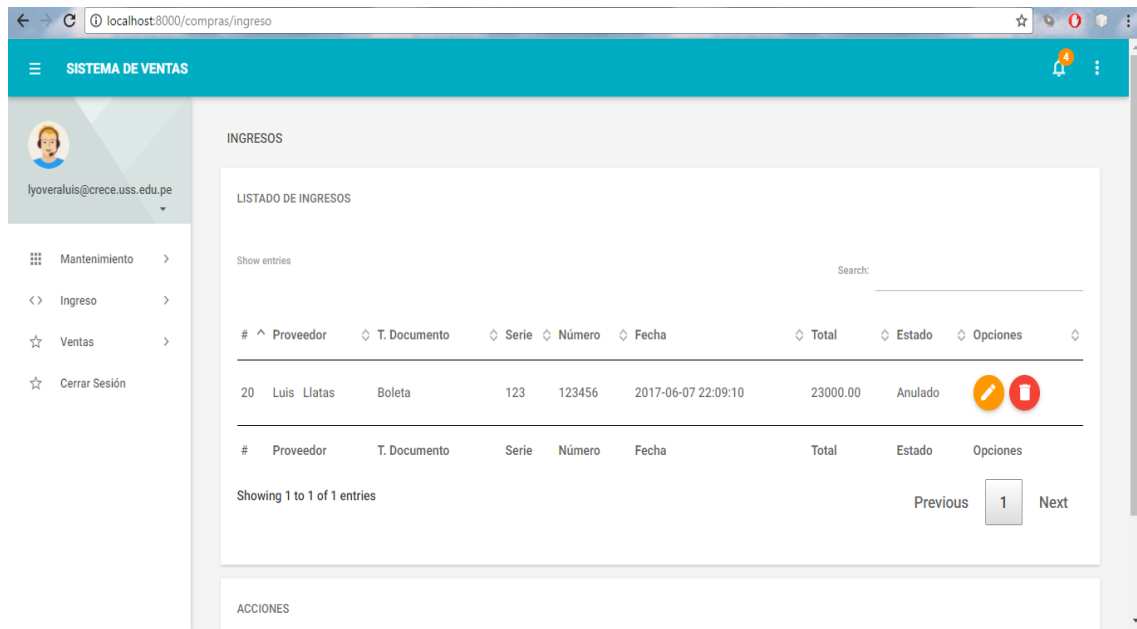


Figura 32. Pantalla Ingreso-Producto

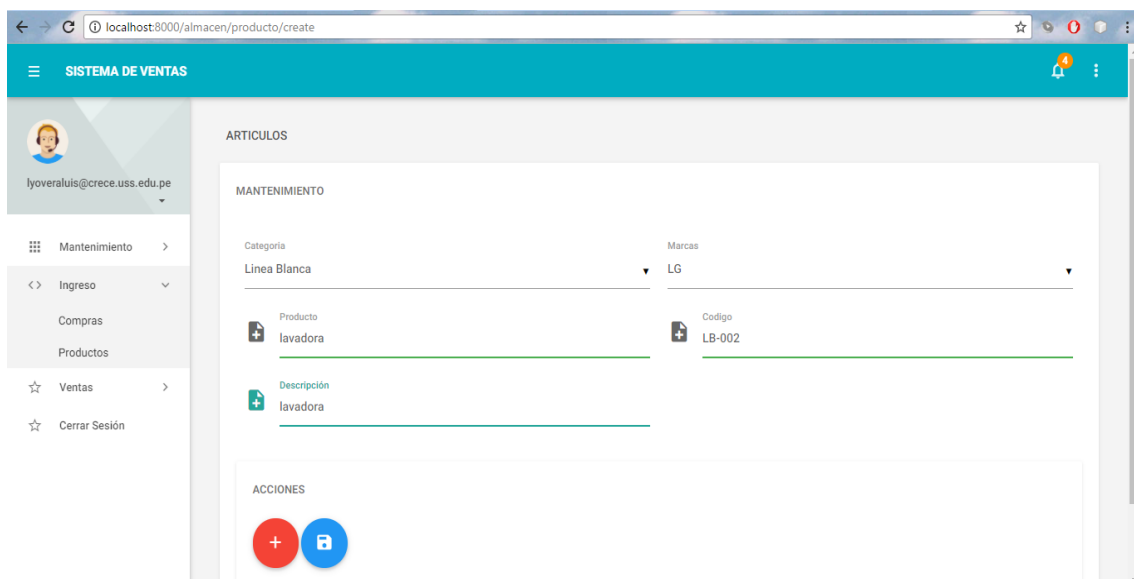


Figura 33. Pantalla de Productos

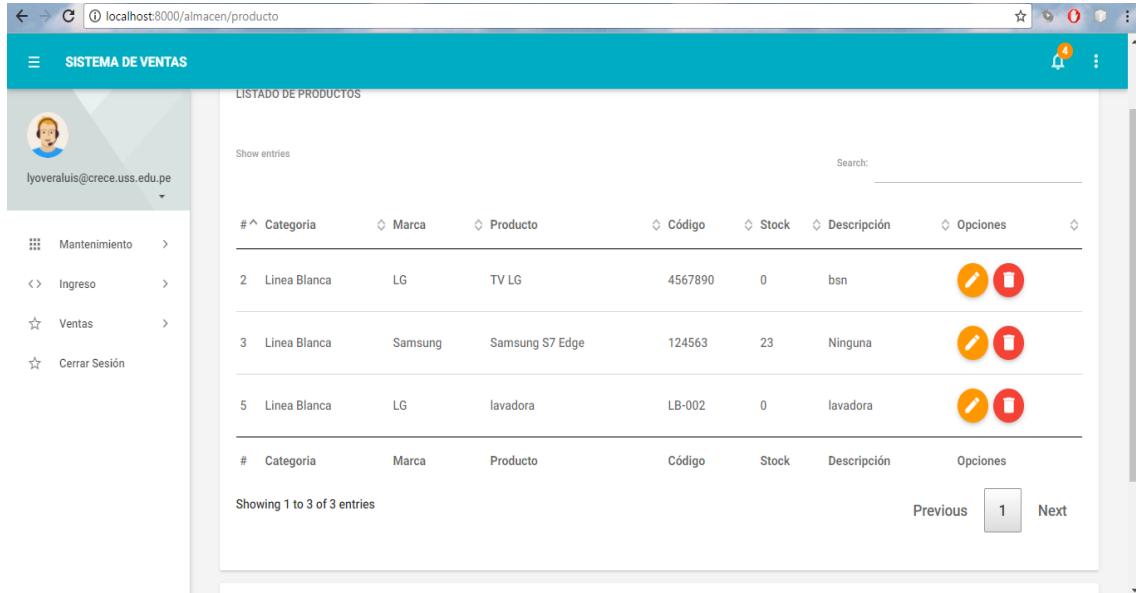


Figura 34. Pantalla Ventas

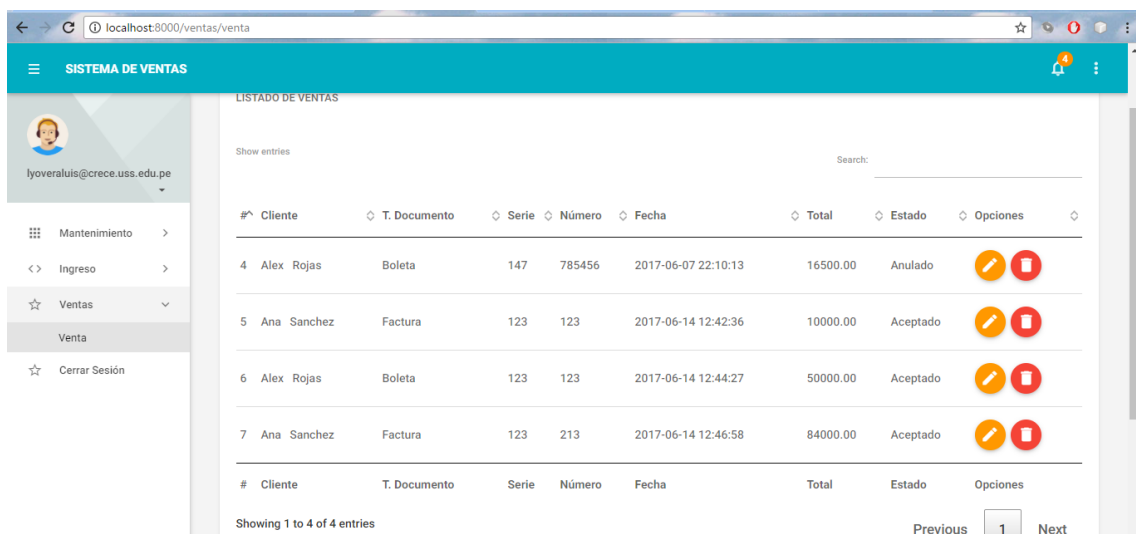


Figura 35. Pantalla Agregar una Venta

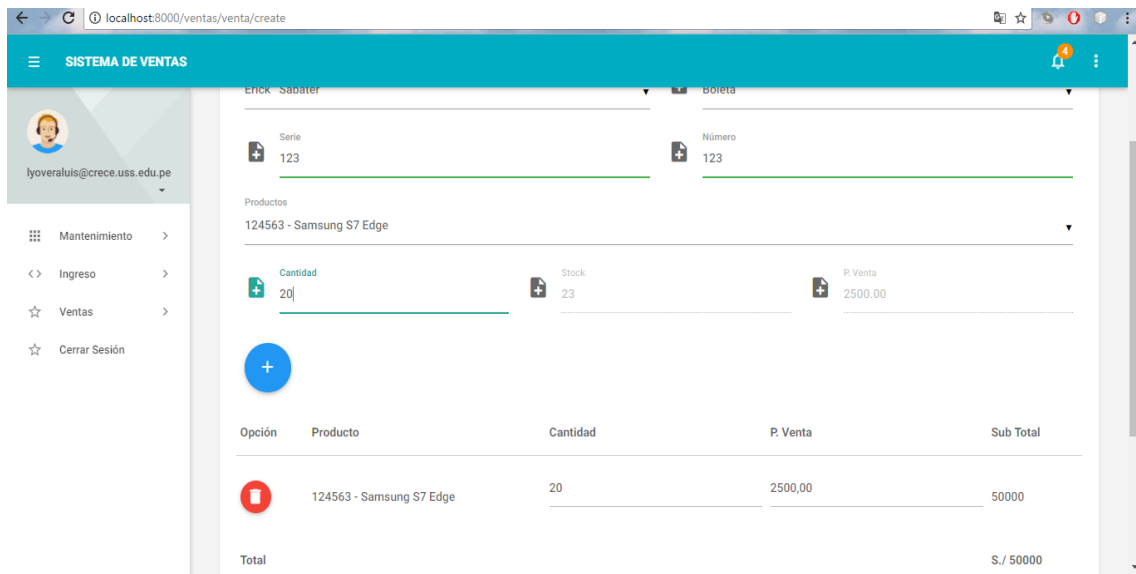


Figura 36. Pantalla Confirmar Venta

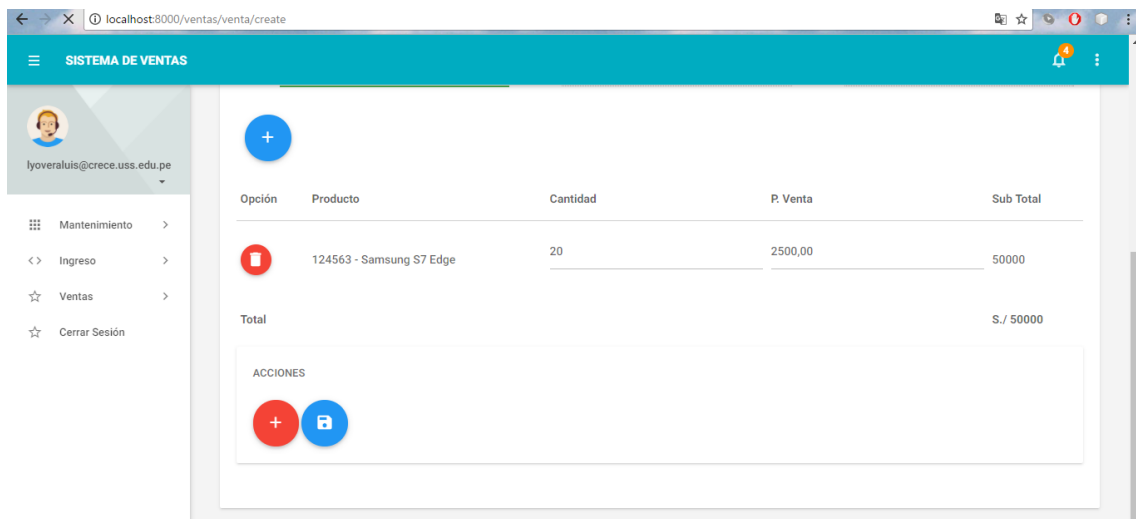












Figura 37. Pantalla Listado de Ventas

#	Cliente	T. Documento	Serie	Número	Fecha	Total	Estado	Opciones
4	Alex Rojas	Boleta	147	785456	2017-06-07 22:10:13	16500.00	Anulado	 
5	Ana Sanchez	Factura	123	123	2017-06-14 12:42:36	10000.00	Aceptado	 
6	Alex Rojas	Boleta	123	123	2017-06-14 12:44:27	50000.00	Aceptado	 
7	Ana Sanchez	Factura	123	213	2017-06-14 12:46:58	84000.00	Aceptado	 
8	Erick Sabater	Boleta	123	123	2017-06-21 11:45:27	50000.00	Aceptado	 

Showing 1 to 5 of 5 entries

Previous 1 Next





### Implementación con Framework Yii

PHP: Yii 2.0 requiere PHP 5.4.0 o superior, y es que hace uso de las últimas características de PHP como los namespaces (forma de encapsular objetos), traits (sirve para reutilizar series de métodos en diferentes clases sin necesidad de extender o implementar).

### Resultados del Sistema de Ventas -SISVENTAS Realizado con el Framework Yii

Figura 38.Pantalla de Logueo

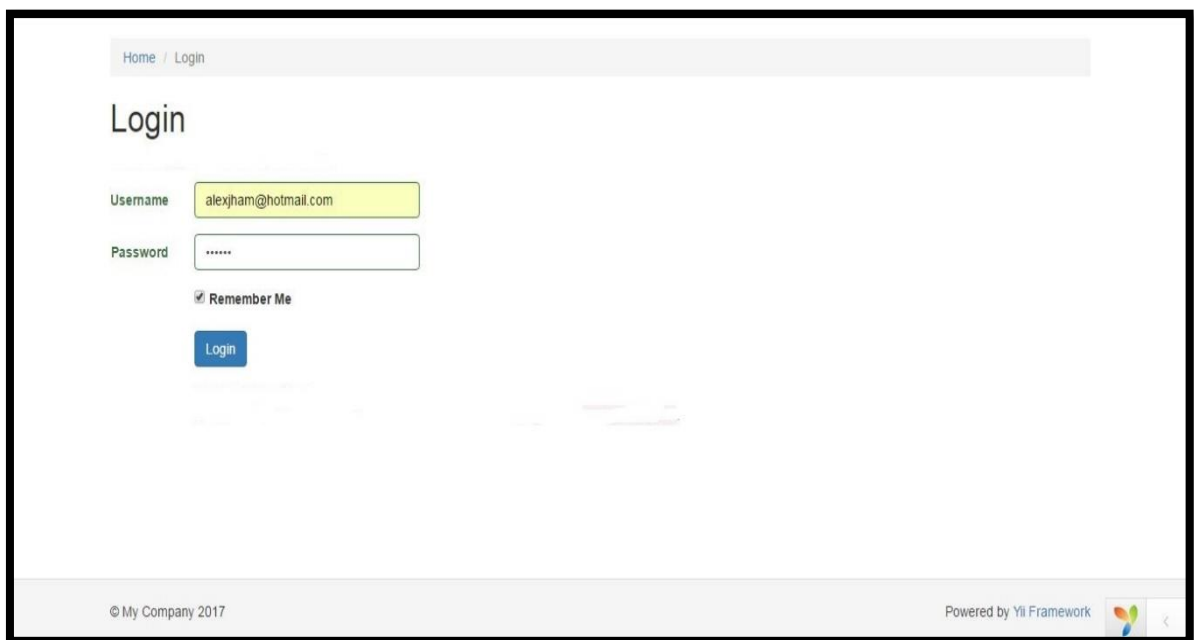


Figura 39. Pantalla Categorías

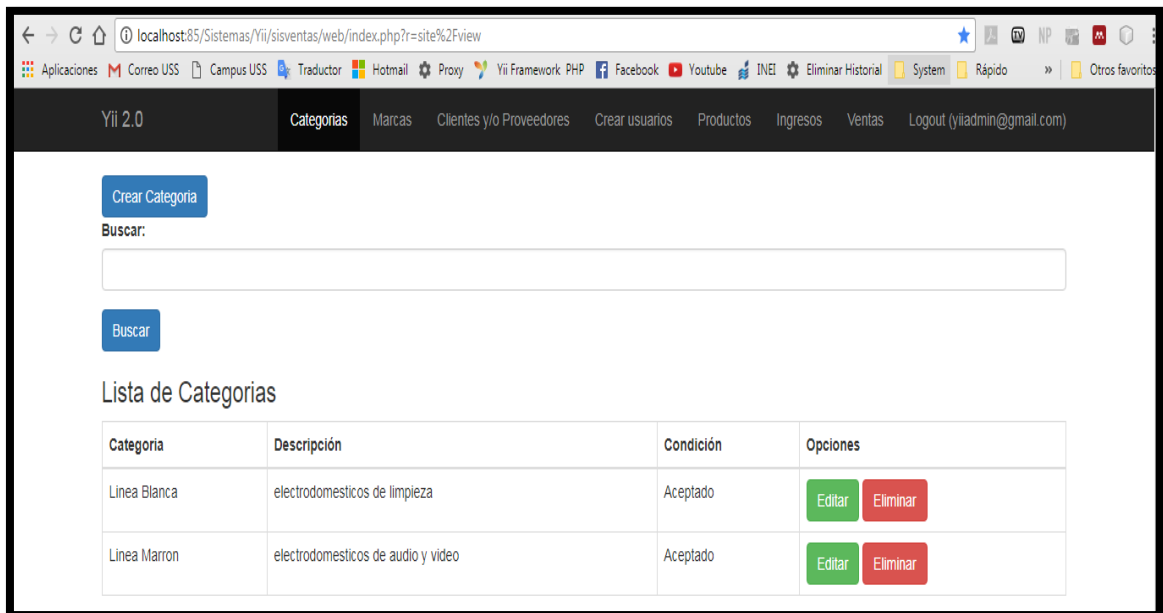


Figura 40. Pantalla Agregar Nueva Categoría

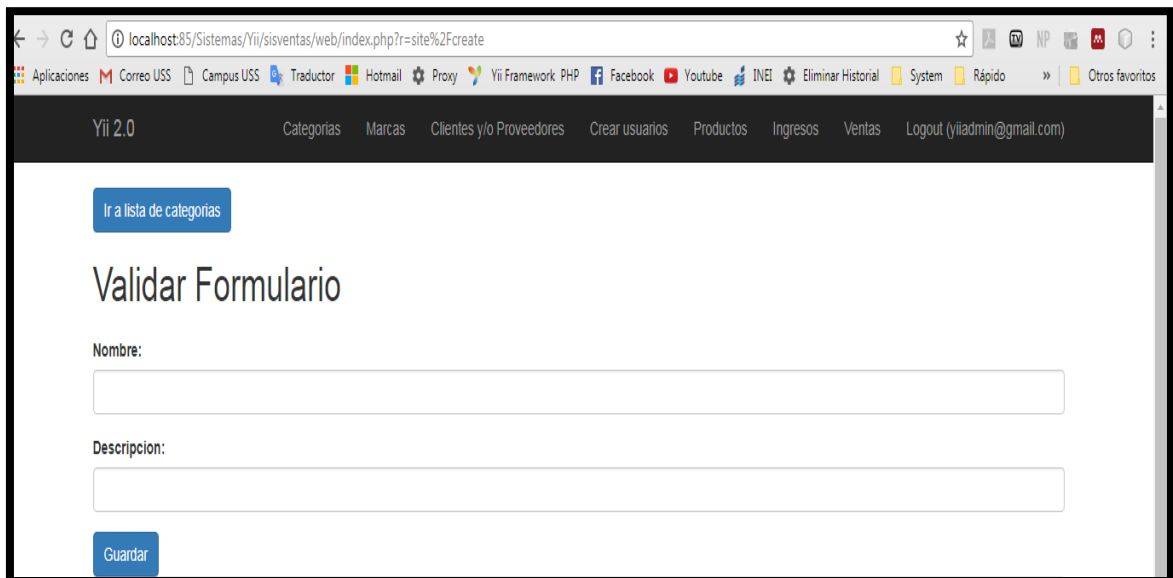


Figura 41. Pantalla Editar Categorías

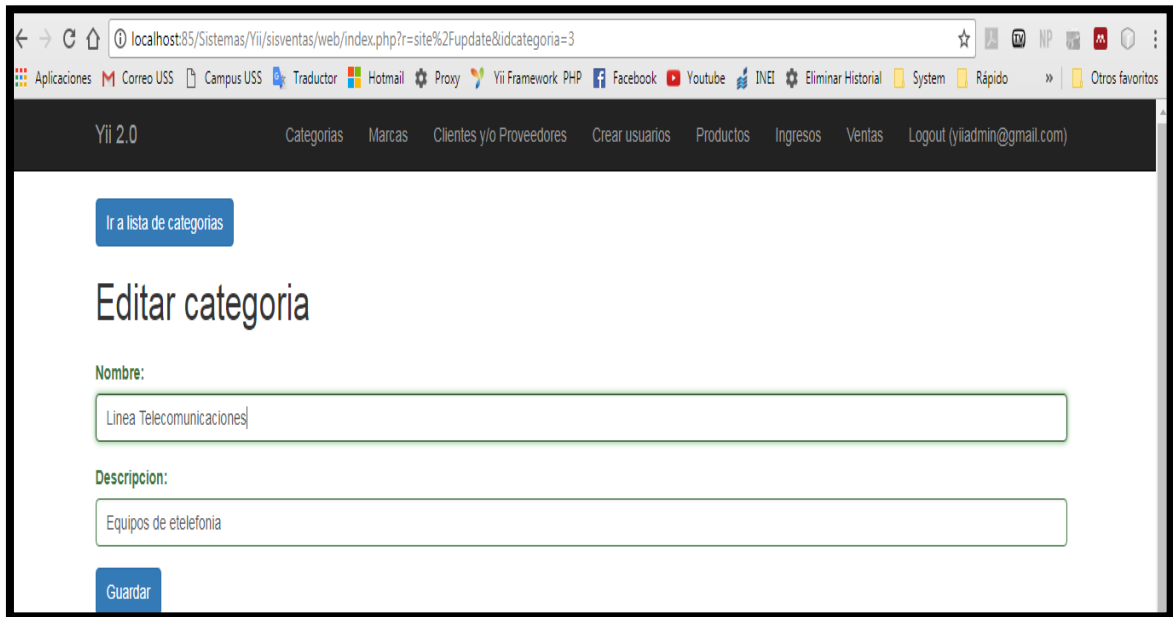


Figura 42. Pantalla de Eliminar Categoría

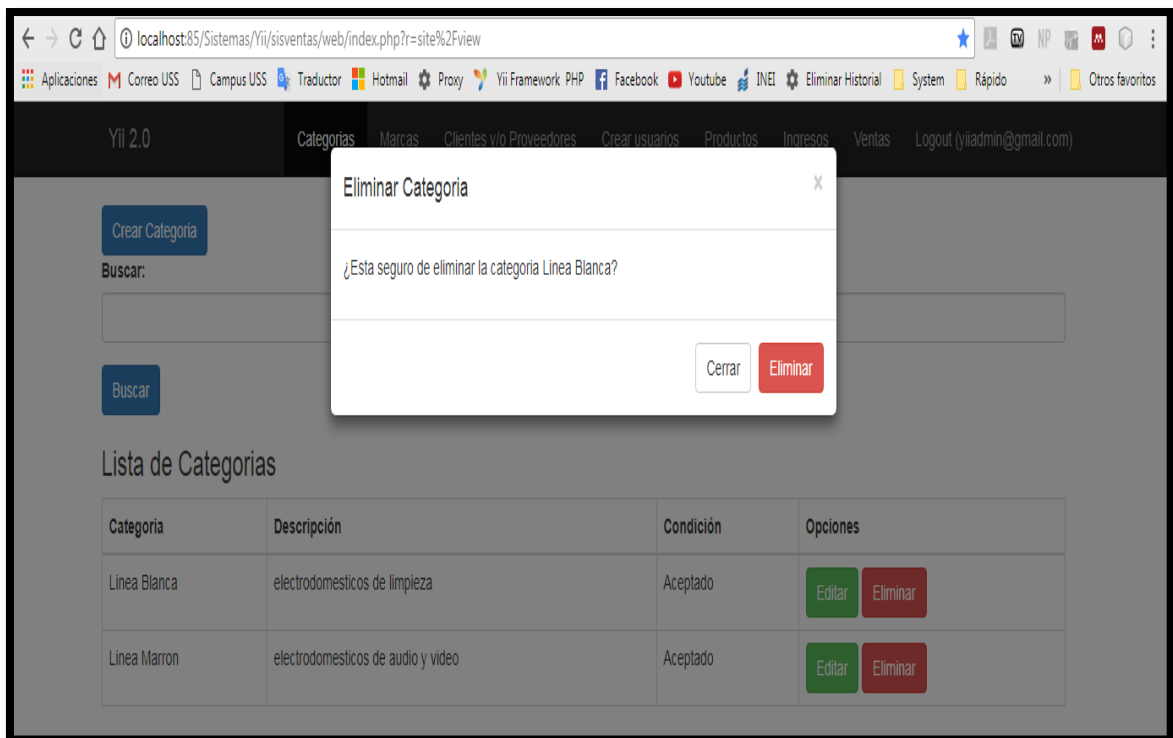


Figura 43. Pantalla Marcas

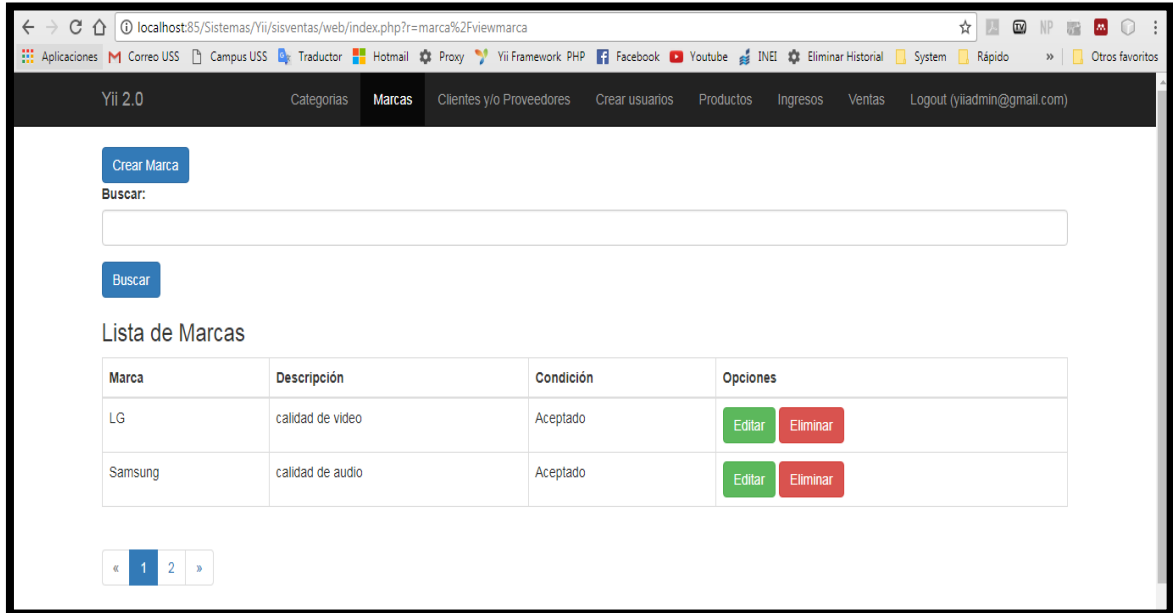


Figura 44. Pantalla Listar Marcas

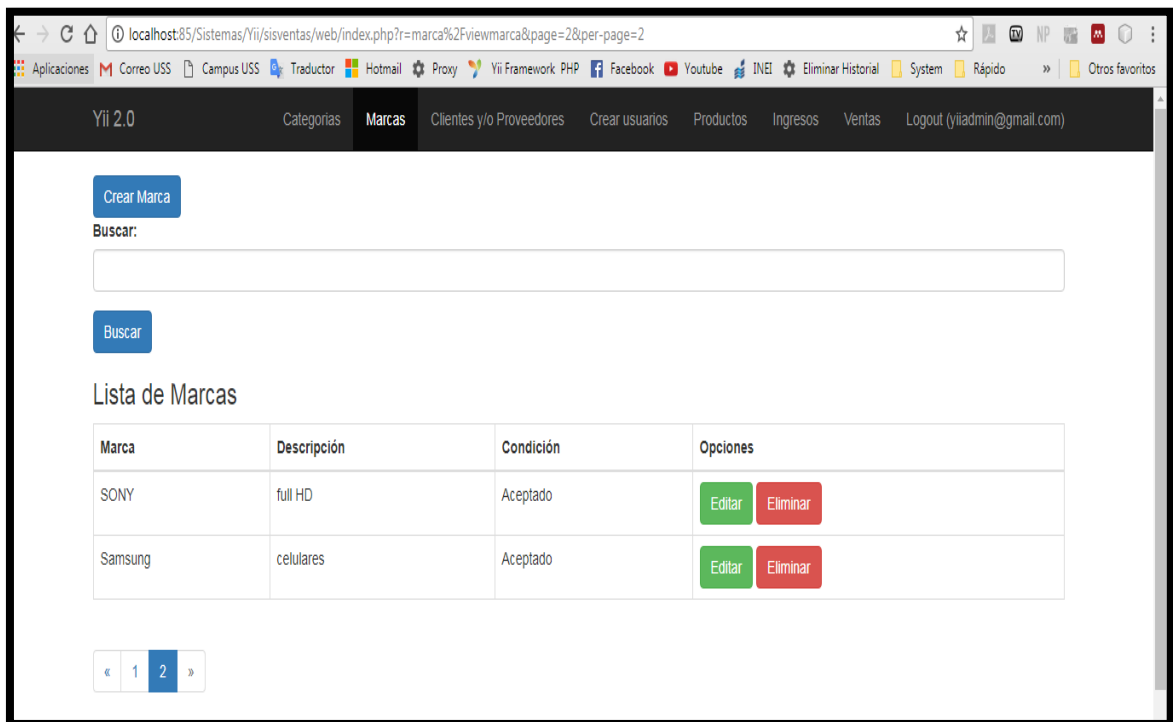


Figura 45. Pantalla Productos

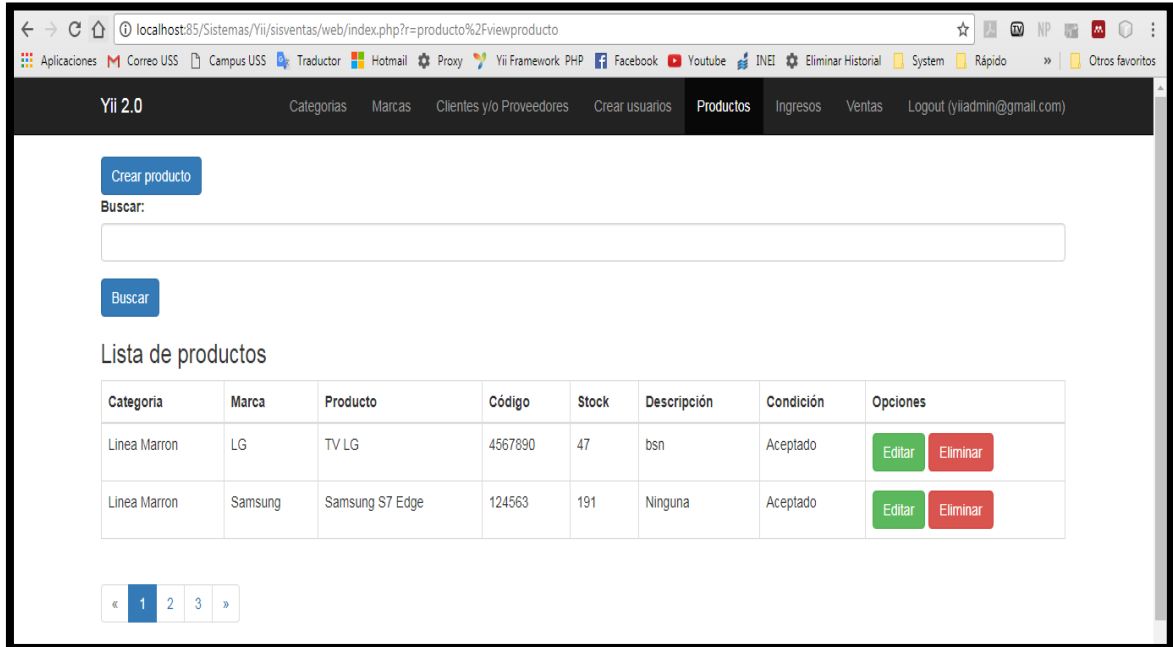


Figura 46. Pantalla Agregar Nuevo Producto

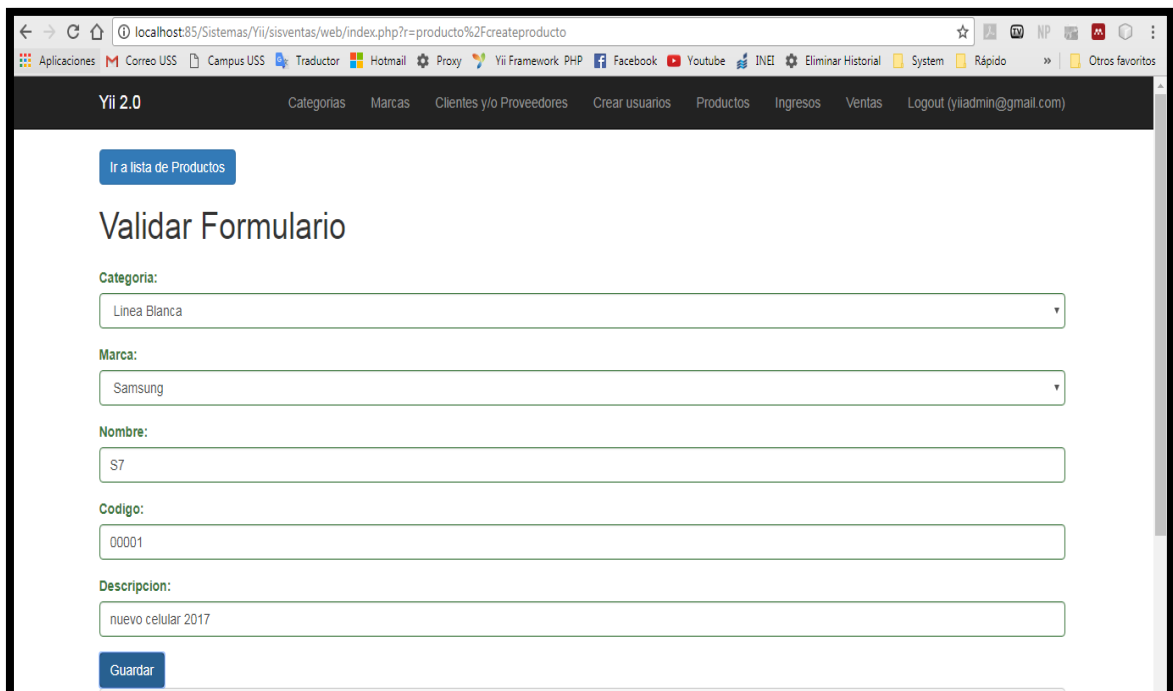


Figura 47. Pantalla Persona

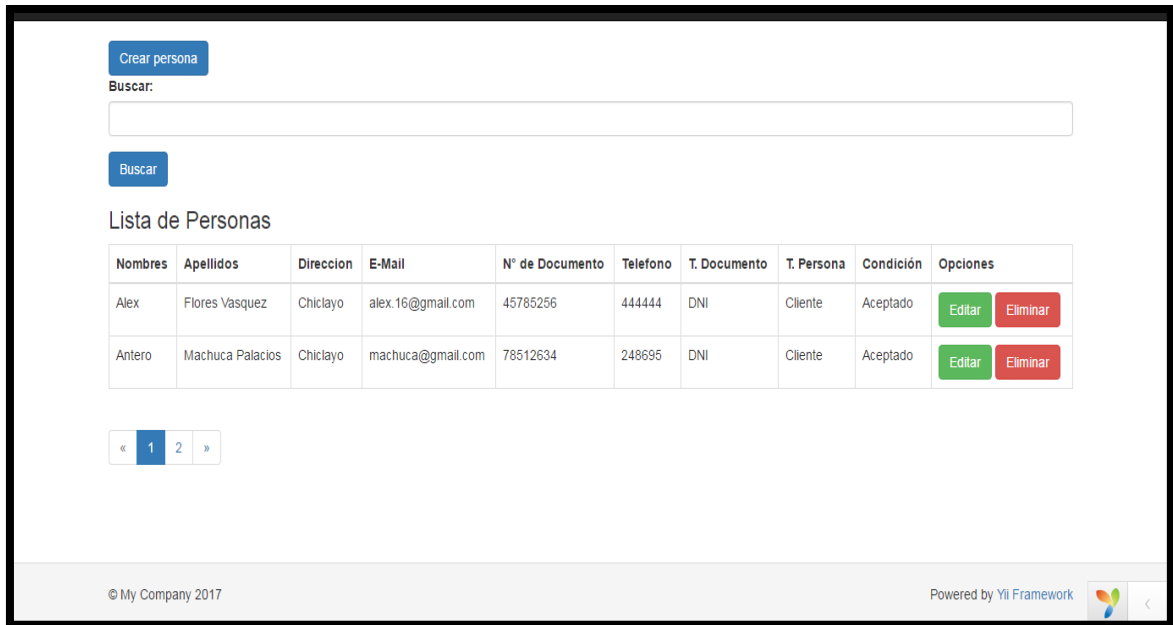


Figura 48. Pantallazo Proveedor

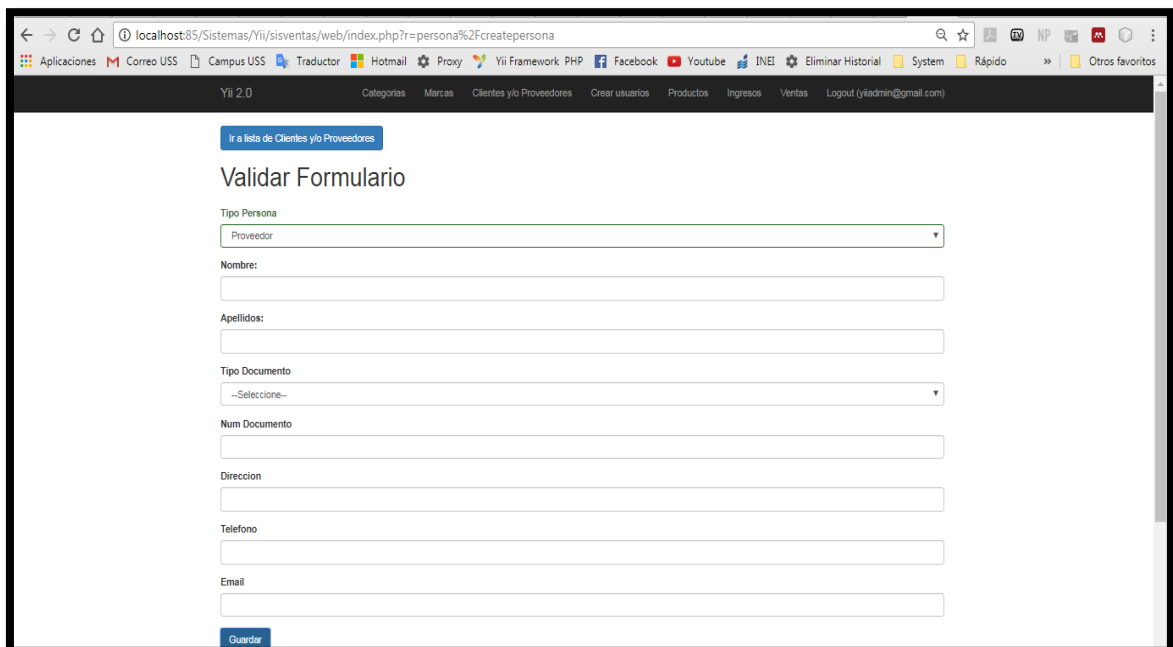


Figura 49. Pantallazo Editar Trabajador

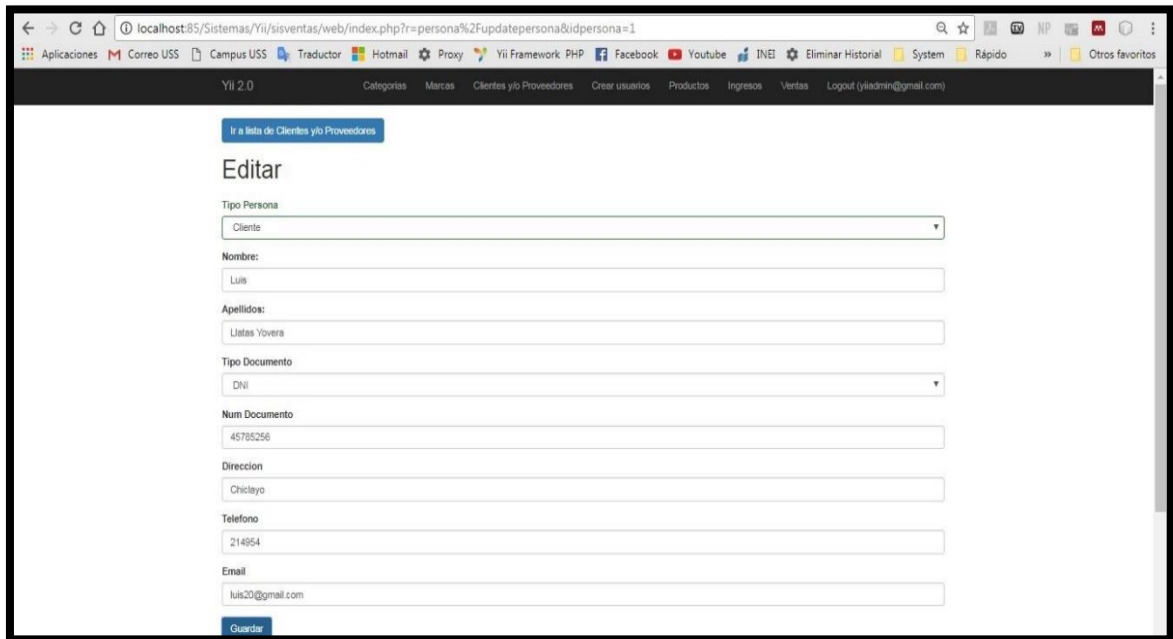


Figura 50. Pantalla Lista de Ingresos

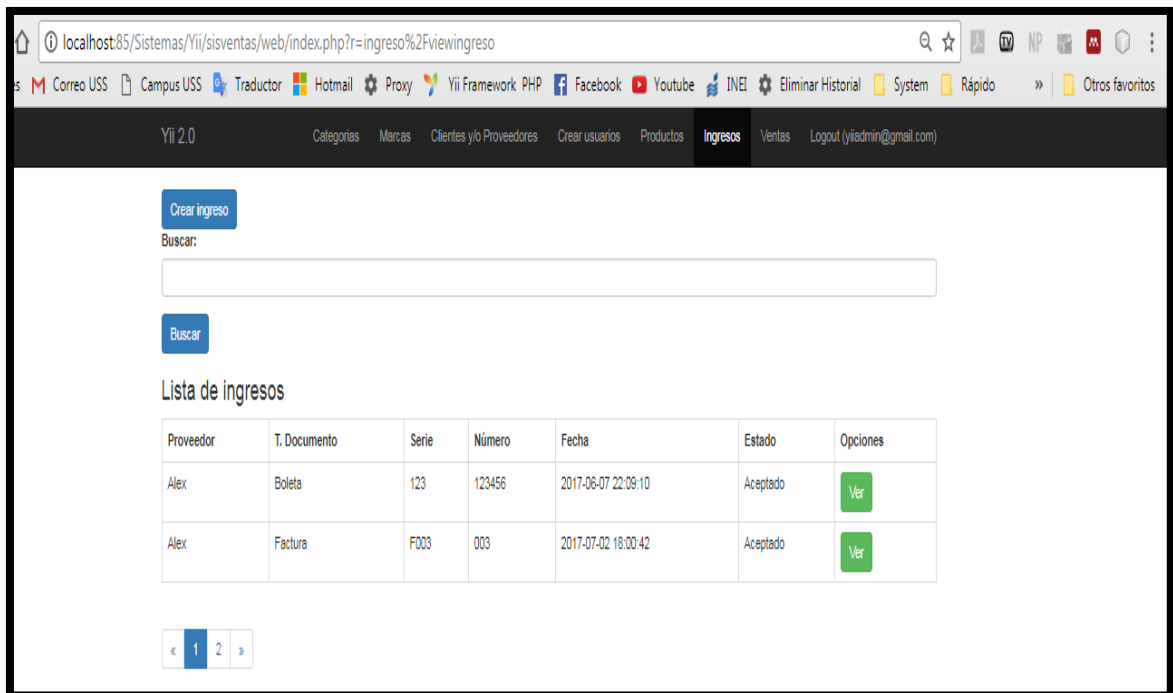


Figura 51. Pantalla Agregar Nuevo Ingreso

Ir a lista de Ingresos

### Ingresos

Proveedor:

Tipo Documento:  Serie:  Número:

Producto:  Cantidad:  P. Compra:

P. Venta:

Opción	Producto	Cantidad	P. Compra	P. Venta	Sub Total
Total					

Figura 52. Pantalla Productos

Buscar:

### Lista de productos

Categoría	Marca	Producto	Código	Stock	Descripción	Condición	Opciones
Linea Marron	LG	TV/LG	4567890	47	bsn	Aceptado	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>
Linea Marron	Samsung	Samsung S7 Edge	124563	191	Ninguna	Aceptado	<input type="button" value="Editar"/> <input type="button" value="Eliminar"/>





Figura 53. Pantalla Ventas

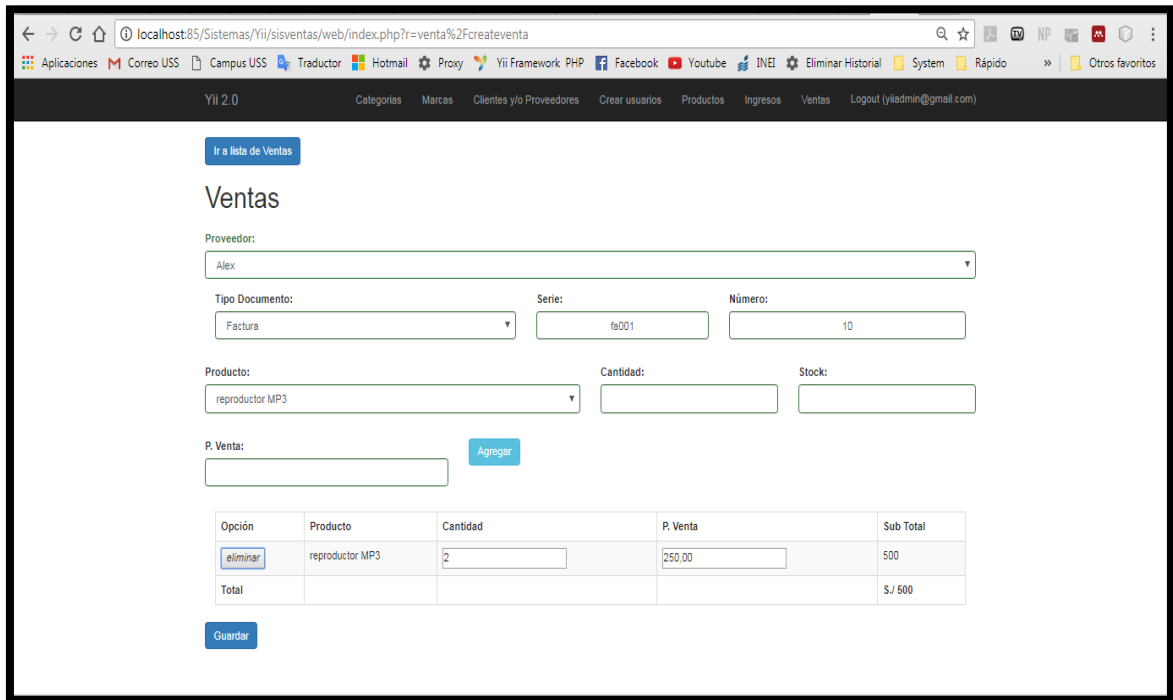


Figura 54. Pantalla Listar Venta

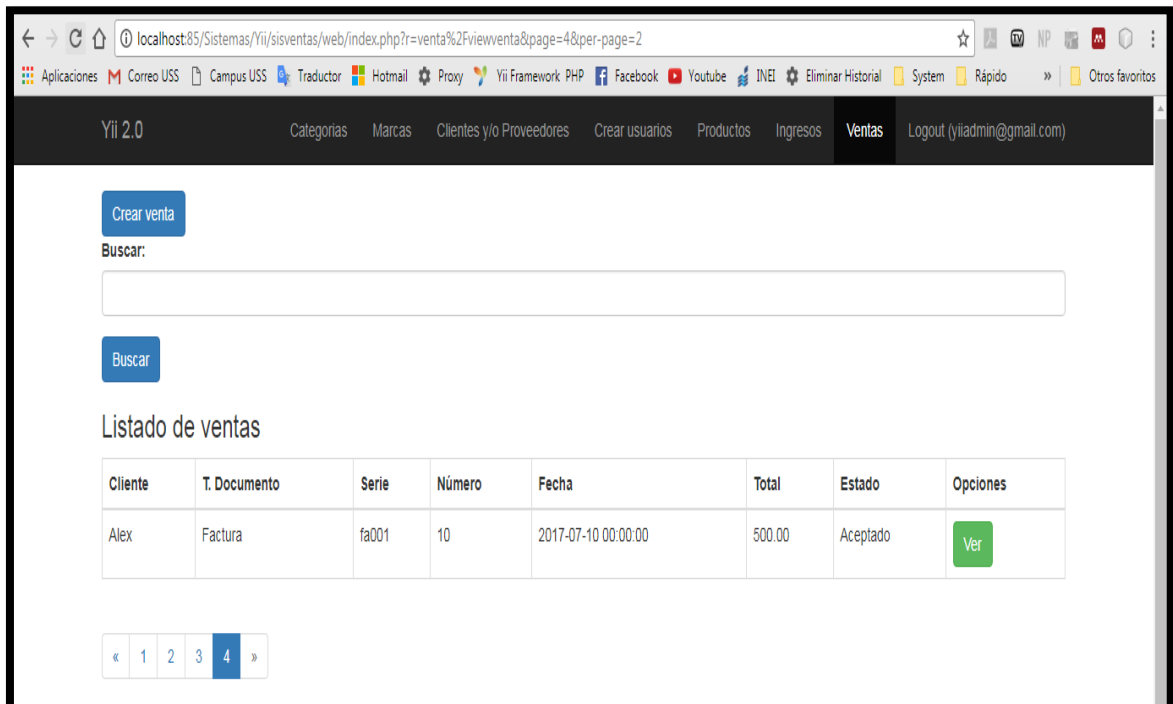


Figura 55. Pantalla Agregar Nuevo Trabajador

Yii 2.0    Categorías    Marcas    Clientes y/o Proveedores    **Crear usuarios**    Productos    Ingresos    Ventas    Logout (yiiadmin@gmail.com)

## Usuarios y/o Trabajadores

Idpersona

Email

Tipo Trabajador

Password

Password Repeat



## Justificación MVC en la presente Investigación

Se utilizó el patrón de diseño de software para programación MVC, ya que los framework PHP Yii y Laravel nos proporciona esta característica, que ha sido consultada en sus web oficiales.

Se presenta un resumen de MVC en cada framework según las fuentes en consulta.

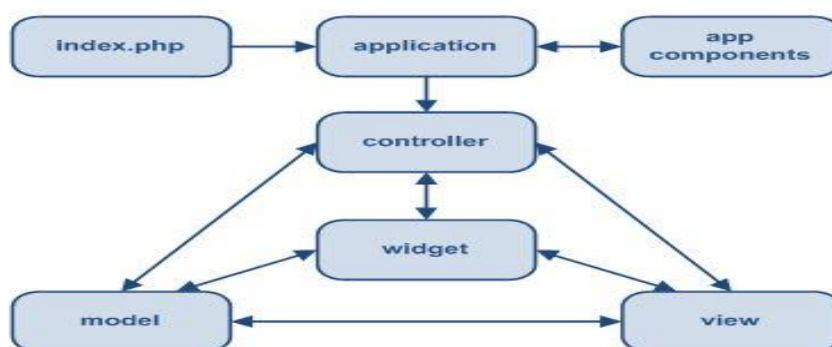
### Yii

Yii implementa el patrón de diseño modelo-vista-controlador (MVC), el cual es ampliamente aceptado en la programación web. MVC tiene como objetivo separar la lógica de las consideraciones de la interfaz de usuario, de tal manera que los desarrolladores puedan cambiar cada parte sin tener que afectar a la otra.

En MVC, el modelo representa la información (los datos); la vista contiene elementos de la interfaz de usuario, como texto, entradas de formulario; y el controlador maneja la comunicación entre el modelo y la vista.

Además cabe recalcar que cuando se implementa MVC, Yii también introduce un controlador frontal, llamado Application, el cual encapsula el contexto de ejecución para poder realizar el procesamiento de una solicitud. La aplicación recoge la información sobre una solicitud del usuario y luego la envía a un controlador apropiado para su posterior tratamiento.

Estructura estática de la aplicación Yii



**Fuente:** <http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>

## Laravel

### Modelo

Laravel incluye un sistema de mapeo de datos relacional que tiene por nombre Eloquent ORM que facilita la creación de modelos que además dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la creación de modelos.

La forma de crear Modelos en Laravel es como el siguiente ejemplo

```
1 use Illuminate\Database\Eloquent\Model;
2
3 class Libro extends Model {
4
5     // nombre de la tabla con la información de los libros
6     protected $table = 'tb_libros';
7 }
```

Incluso el código del modelo puede ser más simple aún, en el caso que el nombre de la tabla coincida con el nombre de la clase.

### Vista

Laravel posee de paquete un sistema de procesamiento de plantillas llamado Blade, lo cual favorece un código mucho más limpio en las Vistas, que además trae un sistema de Caché que lo hace más rápido. El sistema Blade de Laravel, permite una sintaxis mucho más reducida en su escritura.

Por ejemplo, presentamos dos líneas de código PHP:



<?= \$mi\_nombre;?>

En Blade se escribiría de la siguiente forma:

```
{{ $mi_nombre }}
```

### El Controlador

Los controladores son los que tienen la lógica de la aplicación y es donde se permite organizar el código en clases sin tener que escribirlo todo en las rutas. Todos los controladores deben extenderse de la clase BaseController.

Por ejemplo de un controlador en Laravel sería así:

```
class UserController extends BaseController {
    public function mostrarPerfil($id)
    {
        $user = User::find ($id);
        return View::make('user.profile', array ('user' => $user));
    }
}
```

Estos podrían ser llamados en las rutas de diversas formas, pero la más común usando rutas es la siguiente:

```
Route::get('user/{id}', 'UserController@mostrarPerfil');
```

**Fuente:** <https://laracasts.com/series/laravel5fundamentals/episode9>

**5.4 Evaluar los framework en base a los indicadores de rendimiento seleccionados.**

**5.4.1 PLAN DE PRUEBAS PARA MEDIR EL RENDIMIENTO**

Se tiene como finalidad dar a conocer el orden de los pasos realizados para la ejecución de pruebas de rendimiento ,realizada sobre el software SISVENT (Sistema de Ventas) además de realizar la determinada descripción de herramientas, funciones usadas.

**5.4.1.1 Hoja resumen de Proyecto y de modificaciones**

Proyecto		Tipo de Proyecto	
SISVENT- Sistema de Ventas		Proyecto Web Sistema de Ventas	
Documentos Evaluación relacionados			
Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems.doc			
Equipo de Proyecto			
Jefe de Equipo	Gino Paul Gonzales Custodio	Programadores	Luis Humberto Llatas Yovera / Alex Jhampier Rojas Herrera

Versión	Fecha	Cambios Respecto de la versión anterior	Preparado Por	Aprobado Por
1.0	24/07/2017	Versión Inicial	Luis Yovera / Alex Rojas	Gino Paul Gonzales Custodio



### 5.4.1.2 Objetivo del Plan de Pruebas

El presente documento, tiene por finalidad entregar las pautas y definir la estrategia que se seguirá para llevar a cabo la evaluación de los frameworks PHP en base al indicador de rendimiento (tiempo de respuesta, consumo de memoria RAM, tiempo de carga).

El objetivo general del plan es establecer el procedimiento y condiciones para la aplicación del plan de pruebas de tal manera que se defina formalmente lo que se va comparar y como se ejecutarán dichas pruebas.

#### 5.4.1.2.1 Documentos Relacionados

Nombre	Descripción
Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems	Investigación Base para la realización del plan de pruebas.

### 5.4.1.3 Alcance de Las Pruebas

A través de los siguientes cuadros se describen los requerimientos de las pruebas de rendimiento del sistema SISVENT.

#### 5.4.1.3.1 Cuadro Resumen de las Pruebas

<p><b>Módulos del Sistema SisVent a ser probados:</b></p>	<p>Módulos:</p> <ul style="list-style-type: none"> <li>- Categoría</li> <li>- Marca</li> <li>- Producto</li> <li>- Persona</li> <li>- Trabajador</li> </ul>
---	---



<p><b>Objetivos de las Pruebas</b></p>	<p>En estos módulos se realizarán las pruebas para:</p> <p>Comparar los framework PHP: Yii y Laravel a través del indicador de rendimiento, en condiciones iguales a cada uno y de esta manera obtener resultados confiables.</p> <ul style="list-style-type: none"> <li>• Comprobar en el indicador de Tiempo de Respuesta, que framework hace un menor uso y esto mediante el uso indicado de peticiones a prueba.</li> <li>• Permitir definir qué framework hace menor consumo de Memoria RAM, mediante la cantidad de recursos usados por los Framework's</li> <li>• Mencionar mediante la comparación del indicador del Tiempo de Carga la cantidad de milisegundos que usa cada framework.</li> </ul>
<p><b>Detalle del orden de ejecución de los módulos</b></p>	<p>Los módulos se deben ejecutar en forma independiente para luego obtener un promedio por cada prueba, el orden de prueba a cada módulo será el siguiente:</p> <ul style="list-style-type: none"> <li>- Categoría</li> <li>- Marca</li> <li>- Producto</li> <li>- Persona</li> <li>- Trabajador</li> </ul>
<p><b>Responsabilidad de la Prueba</b></p>	<p>Las pruebas son responsabilidad del equipo de investigación encargado del plan de pruebas:</p> <ul style="list-style-type: none"> <li>- Llatas Yovera Luis Humberto</li> <li>- Rojas Herrera Alex Jhampier</li> </ul>





5.4.1.4 Tipo de Requerimiento a medir

5.4.1.4.1 Requerimientos No Funcionales para el Sistema de Ventas

(Sisvent)

Nombre	Tipo	Nivel Criticidad (Bajo, Medio, Alto)
INDICADOR RENDIMIENTO	No Funcional	Alto

5.4.1.5 Criterios de Aprobación / Rechazo

5.4.1.5.1 Rango de Aceptación en Tiempo de Respuesta

Fuente: BULTONOVA Y PROKOFYEVA (2017)

NIVEL DE ACEPTACIÓN	RANGO 1:Minimo Valor	RANGO 2:Maximo Valor
Aceptable	$\geq 14\ 0000$ milisegundos (14 segundos)	$\leq 15$ milisegundos (15 segundos)
Deficiente	$< 14\ 0000$ milisegundos (14 segundos)	$> 15$ milisegundos (15 segundos)

5.4.1.5.2 Rango de aceptación en Consumo de Memoria RAM

Fuente: BULTONOVA Y PROKOFYEVA (2017)

NIVEL DE ACEPTACIÓN	RANGO 1:Minimo Valor	RANGO 2:Maximo Valor
Aceptable	$\geq 6291456$ bytes ( 6Mb )	$\leq 9437184$ bytes ( 9Mb )
Deficiente	$< 6291456$ bytes ( 6Mb )	$> 9437184$ bytes ( 9Mb )



### 5.4.1.5.3 Rango de aceptación en Tiempo de Carga

Fuente: BULTONOVA Y PROKOFYEVA (2017)

NIVEL DE ACEPTACIÓN	RANGO 1:Minimo Valor	RANGO 2:Maximo Valor
<b>Aceptable</b>	$\geq 1400.0946564$ microsegundos (0.0014 segundos)	$\geq 1500.0946564$ microsegundos (0.0015 segundos)
<b>Deficiente</b>	$< 140009.46564$ microsegundos (0.14 segundos)	$> 150009.46564$ microsegundos (0.15 segundos)

### 5.4.1.6 Casos de Pruebas

Con la finalidad de que los datos obtenidos durante el desarrollo de las pruebas sean de carácter objetivo y científico se utilizó como base la investigación de Prokofyeva, N., & Boltunova, V. (2017). Para establecer valores aceptables de evaluación de indicadores de rendimiento.

#### 5.4.1.6.1 Bases De Datos, Servidor Y Hardware Para El Plan De Pruebas

Base de Datos : SISVENT- dbsisventas (MySQL)

Servidor Web : SISVENT- Xampp v3.2.2



### Hardware a utilizar

Hardware Utilizado			
Modelo	Descripción	Sistema Operativo	Memoria RAM
<b>LENOVO 20354</b>	Intel(R) Core(TM)i7-4510u CPU @ 2.00GHz (4CPUs)	Windows 8.1	16 RAM

Hardware Utilizado			
Modelo	Descripción	Sistema Operativo	Memoria RAM
<b>Toshiba Satellite L745</b>	Intel(R) Core(TM)i5-4230M CPU @ 2.40 GHz (4 CPUs), ~2.4GHz	Windows 7 Ultimate 64 bits	4GB

#### 5.4.1.6.2 Software Y Funciones A Usar

Indicador	Herramienta / Función
Tiempo de Respuesta	<b>Apache 2.4.25</b>
Consumo de Memoria RAM	<b>(memory_get_usage )</b>
Tiempo de Carga	<b>(microtime )</b>



### 5.4.1.6.3 Escenario De Las Pruebas

**Para las pruebas No Funcionales se debe comprobar que:**

Que el rendimiento de los Framework sea el óptimo a través de los indicadores (Tiempo de Respuesta, Consumo de Memoria RAM y Tiempo de Carga) a través de la funcionalidad permita los siguientes:

Nombre	Descripción
<b>SISVENT</b>	Se deberá aprobar la comparación con un 100% de las pruebas de rendimiento ejecutadas pero con un 90% de aceptación. Esto quiere decir el 90% de las pruebas deben ser exitosas. El restante 10% pueden existir errores medios o bajos, pero no graves.

### 5.4.1.7 Casos De Pruebas

#### ❖ Tiempo de Respuesta

- Se indica que para realizar el plan de Pruebas se iniciara con una base de 100 peticiones al servidor, hasta las 1000 peticiones.
- Recalcamos que la cantidad estándar tomada fue de 1000 peticiones, las cuales están descritas en la Investigación de Prokofyevaa & Boltunovaa, 2017.
- La explicación a de las herramientas y funciones utilizadas se explica al finalizar el plan de pruebas propuesto.

#### Primera Prueba en laptop Toshiba

- Tiempo de Respuesta en Laravel

Cantidad de Peticiones al Servidor Web	Tiempo de respuesta en milisegundos (ms)	Valor de Prueba
100	3000	



200	5300	
300	10000	
400	13000	
500	14000	
600	14900	
700	15000	
800	15100	
900	15300	
1000	22000	
<b>Promedio Tiempo de Respuesta (ms)</b>	<b>12760</b>	<b>Deficiente</b>

➤ En esta la prueba realizada obtenemos un resultado **Deficiente** según los rangos de aprobación.

- Tiempo de Respuesta en Yii

Cantidad de Peticiones al Servidor Web	Tiempo de respuesta en milisegundos (ms)	Valor de Prueba
100	2000	
200	4000	
300	4020	
400	4030	
500	4500	
600	4700	
700	4800	



800	5000	
900	5900	
1000	9000	
<b>Promedio Tiempo de Respuesta (ms)</b>	<b>4795</b>	<b>Deficiente</b>

- En esta la prueba realizada obtenemos un resultado **Deficiente** según los rangos de aprobación.

### Segunda Prueba laptop Lenovo

- Tiempo de Respuesta en Laravel

Cantidad de Peticiones al Servidor Web	Tiempo de respuesta en milisegundos (ms)	VALOR DE PRUEBA
100	8000	
200	10000	
300	12000	
400	12500	
500	13000	
600	13500	
700	13800	
800	14000	



900	14800	
1000	30000	
<b>Promedio Tiempo de Respuesta (ms)</b>	<b>14160</b>	<b>Acceptable</b>

- En esta la prueba realizada obtenemos un resultado **Acceptable** según los rangos de aprobación.
- Tiempo de Respuesta en Yii

Cantidad de Peticiones al Servidor Web	Tiempo de respuesta en milisegundos (ms)	VALOR DE PRUEBA
100	9000	
200	11500	
300	11580	
400	11590	
500	12000	
600	13000	
700	15900	
800	16500	
900	18000	
1000	21500	
<b>Promedio Tiempo de Respuesta (ms)</b>	<b>14057</b>	<b>Acceptable</b>



➤ En esta la prueba realizada obtenemos un resultado **Aceptable** según los rangos de aprobación.

❖ **Consumo de Memoria RAM**

- Se obtendrá un resultado, luego de aplicar la fórmula establecida en la Operacionalización.
- La función se utilizó en los módulos de Mantenimiento que incluye Categoría, Marca, Productos, Clientes, Persona y Venta.
- Se realizaron las pruebas en cada Laptop, la cual ha resultado los siguientes resultados.
- La explicación a detalle las herramientas y funciones utilizadas se explica al finalizar el plan de pruebas propuesto.

**Primera Prueba en laptop Toshiba**

- Consumo de Memoria RAM en Laravel

MANTENIMIENTO	MEMORIA RAM INICIAL (bytes) – MRI	MEMORIA RAM FINAL (bytes)- MRF	MEMORIA_RAM_PROMEDIO (bytes) – MRP MRP = (MRI + MRF)/2	VALOR DE PRUEBA
Categoría	7340032	7340032	7340032	
Marca	7340032	7602176	7471104	
Producto	7602176	7864320	7733248	
Persona	7340032	7602176	7471104	
Trabajador	7340032	7340032	7340032	
<b>Promedio de Memoria RAM</b>			<b>7471104</b>	<b>Aceptable</b>





➤ En esta la prueba realizada obtenemos un resultado **Aceptable** según los rangos de aprobación.

- Consumo de Memoria RAM en Yii

MANTENIMIENTO	MEMORIA RAM INICIAL (bytes) – MRI	MEMORIA RAM FINAL (bytes)-MRF	MEMORIA_RAM_PROMEDIO (bytes) – MRP MRP = (MRI + MRF)/2	VALOR DE PRUEBA
Categoría	5201282	5215660	5208471	
Marca	5423900	5468480	5446190	
Producto	5561384	5607660	5584522	
Persona	5436312	5481840	5459076	
Venta	5501590	5546650	5524120	
<b>Promedio de Memoria RAM</b>			5444475	Deficiente

➤ En esta la prueba realizada obtenemos un resultado **Deficiente** según los rangos de aprobación.

### Segunda Prueba en laptop Lenovo

- Consumo de Memoria RAM en Laravel

Mantenimiento	MEMORIA RAM INICIAL (bytes) – MRI	MEMORIA RAM FINAL (bytes)-MRF	MEMORIA_RAM_PROMEDIO (bytes) – MRP MRP = (MRI + MRF)/2	VALOR DE PRUEBA
<b>Categoría</b>	8122144	8165992	8144068	
<b>Marca</b>	7876976	8170408	8023692	



<b>Producto</b>	7953288	8246616	8099952	
<b>Persona</b>	8150080	8193928	8172004	
<b>Venta</b>	7985328	8278672	8132000	
<b>Promedio de Memoria RAM</b>			8114343	Aceptable

➤ En esta la prueba realizada obtenemos un resultado **Aceptable** según los rangos de aprobación.

- Consumo de Memoria RAM en Yii

MANTENIMIENTO	MEMORIA RAM INICIAL (bytes) – MRI	MEMORIA RAM FINAL (bytes)- MRF	MEMORIA RAM PROMEDIO (bytes) – MRP $MRP = (MRI + MRF)/2$	VALOR DE PRUEBA
<b>Categoría</b>	6201284	6315660	6258472	
<b>Marca</b>	6423904	6468488	6446196	
<b>Producto</b>	6561384	6607680	6584532	
<b>Persona</b>	6436312	6481880	6459096	
<b>Venta</b>	6501584	6546680	6524132	
<b>Promedio de Memoria RAM</b>			6454485	Aceptable

➤ En esta la prueba realizada obtenemos un resultado **Aceptable** según los rangos de aprobación.

❖ **Tiempo de Carga**

- Vamos aplicar la fórmula establecida en la operacionalización
- Del sistema de Ventas se usó los módulos de Mantenimiento que incluye Categoría, Marca, Productos, Clientes, Persona y Venta.



- Se realizaran las pruebas correspondientes en cada Laptop, obteniendo los siguientes resultados.
- La explicación a detalle las herramientas y funciones utilizadas se explica al finalizar el plan de pruebas propuesto.

**Primera Prueba laptop Toshiba**

- Tiempo de Carga en Yii

Mantenimiento	TIEMPO CARGA INICIAL (microsegundos) – TCI	TIEMPO CARGA FINAL (microsegundos)- TCF	TIEMPO_CARGA_P ROMEDIO (microsegundos) – TCP TCP = (TCI + TCF)/2	VALOR DE PRUEBA
<b>Categoría</b>	1388098355.2400	1399098355.2420	1393598355,20041	
<b>Marca</b>	1398097111.2570	1399097111.1579	1398597111,20745	
<b>Producto</b>	1399097832.1550	1399097832.1563	1399097471,65645	
<b>Persona</b>	1399098100.0510	1399098100.0519	1399098100,05145	
<b>Venta</b>	1399098355.5400	1399098355.5410	1399098355,54005	
<b>Promedio de Tiempo de Carga</b>			1397897878,731252	Deficiente

➤ En esta la prueba realizada obtenemos un resultado **Deficiente** según los rangos de aprobación.

- Tiempo de Carga en Laravel

Mantenimiento	TIEMPO CARGA INICIAL (microsegundos) –TCI	TIEMPO CARGA FINAL (microsegundos) -TCF	TIEMPO_CARGA_PROMEDIO (microsegundos) – TCP TCP = (TCI + TCF)/2	VALOR DE PRUEBA
<b>Categoría</b>	1399094372.0090	1399094372.0379	1399094372,02345	



<b>Marca</b>	1399094131.2730	1399094131.3160	1399094131,2945	
<b>Producto</b>	1399094609.7700	1399094609.8600	1399094783,91	
<b>Persona</b>	1399094957.9900	1399094958.0500	1399094958,02	
<b>Venta</b>	1399095211.0255	1399095211.0859	1399095211,0557	
<b>Promedio de Tiempo de Carga</b>			1399094691,26073	Deficiente

- En esta la prueba realizada obtenemos un resultado **Deficiente** según los rangos de aprobación.

### Segunda Prueba laptop Lenovo

- Tiempo de Carga en Yii

Mantenimiento	TIEMPO CARGA INICIAL (microsegundos) –TCI	TIEMPO CARGA FINAL (microsegundos)-TCF	TIEMPO_CARGA_PROMEDIO (microsegundos) – TCP $TCP = (TCI + TCF)/2$	VALOR DE PRUEBA
<b>Categoría</b>	1499098355.3404	1499098355.3423	1499098355.34135	
<b>Marca</b>	1499097111.3575	1499097111.3587	1499097111.3581	
<b>Producto</b>	1499097832.2554	1499097832.2573	1499097832.25635	
<b>Persona</b>	1499098100.0519	1499098100.0529	1499098100.0524	
<b>Venta</b>	1499098355.6409	1499098355.6413	1499098355.6411	
<b>Promedio de Tiempo de Carga</b>			1229260150,92986	Deficiente

- En esta la prueba realizada obtenemos un resultado **Deficiente** según los rangos de aprobación.



- Tiempo de Carga en Laravel

Mantenimiento	TIEMPO CARGA INICIAL (microsegundos) -TCI	TIEMPO CARGA FINAL (microsegundos)- TCF	TIEMPO_CARGA_PRO MEDIO (microsegundos) – TCP TCP = (TCI + TCF)/2	VALOR DE PRUEBA
<b>Categoría</b>	1499094 372.0098	14990943 72.0385	1499094372,02415	
<b>Marca</b>	1499094 131.2732	14990941 31.3168	1499094131,295	
<b>Producto</b>	1499094 609.7711	14990946 09.8602	1499094609,81565	
<b>Persona</b>	1499094 957.9902	14990949 58.0509	1499094958,02055	
<b>Venta</b>	1499095 211.0266	14990952 11.0869	1499095211,05675	
<b>Promedio de Tiempo de Carga</b>			1499094656,44242	Acceptable

- En esta la prueba realizada obtenemos un resultado **Acceptable** según los rangos de aprobación.

#### 5.4.1.8 Realizar las conclusiones obtenidas y recomendaciones

	DESCRIPCION
✓	Se concluye el plan de pruebas con los resultados respectivos y que serán de referencias a investigaciones futuras.
✓	El uso del protocolo ha permitido tener un orden correcto en la realización del plan de pruebas.
✓	Además la cantidad de peticiones y el tipo de función han sido indicados para medir el indicador de rendimiento, la cual son respaldados por investigaciones confiables previas.
✓	A través de la implementación de un sistema con sus módulos respectivos ha permitido comparar los indicadores de rendimiento y determinar cuál es el mejor framework PHP.



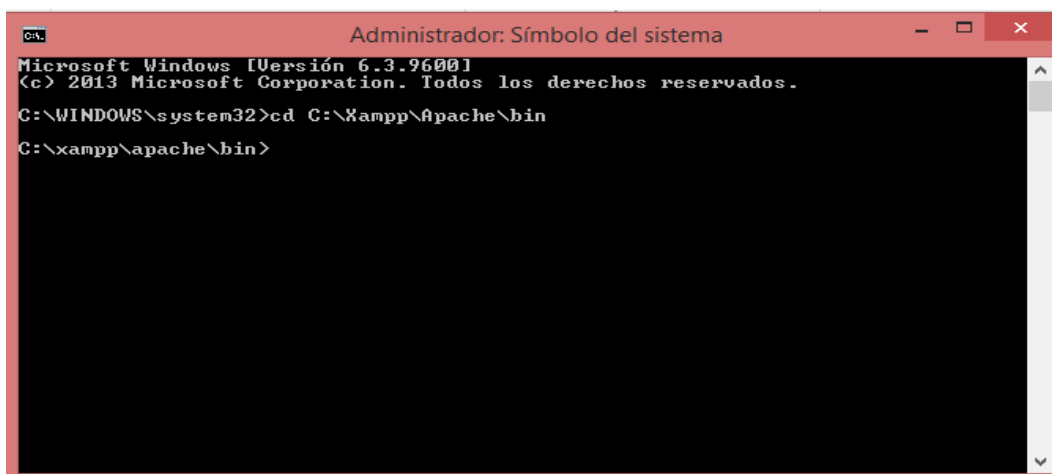
### 5.4.1 Evaluación de Framework Laravel en Tiempo de Respuesta

Para medir este indicador se utilizó un programa de línea de comandos de nombre Apache Bench(2.3), esta herramienta ab viene con la distribución estándar de Apache y al igual que el mismo servidor web Apache, que es un software libre y está distribuido bajo los términos de la Licencia Apache.

En primer lugar lo que se hizo es abrir una terminal de línea de comandos CMD, e ingresar al directorio donde se encuentra la carpeta Apache en un nuestro servidor local Tomcat.

Línea de Comando: `cd C:\Xampp\Apache\bin`

Figura 56. Apache en PC Local



Luego se ingresó el comando: "ab -g resultados/laravel/results.tsv -n 1000 -c 100 -C laravel\_session=eyJpdil6lnZrZzRYM2NGb1ZIMVQ1ZzhPZUItcUE9PSIslnZhbHVlIjoiZzR5SVdPd29xUXk3XC8wSkd5UmQ1cVdsN0hmNFAyVTIRZjN2a0tOeUYrUVJ oXC9VSE5leTdxWFicL1VwSDhsK1ZFam9aRDNJTTlpaVNNbnBSMmVqNjRhSGc 9PSIsIm1hYyI6IjExYzFiNjIwMjBhMjI3Njg4OWUyZjI0ZjdlYjY3ZjRlNmlyMmVmZTU 2OTk1OTYyYWNhNDIzOGU5ZGE4NDE3YzEifQ%3D%3D http://localhost/SistemaVentas-Tesis/laravel/sisVentas/public/ventas/venta"



Este comando forma parte de la herramienta Apache Bench y se compone de las siguientes partes:

Donde:

ab es Apache Bench

-g resultados/laravel/results.tsv guarda los resultados en el archivo results\_laravel.tsv

-n 1000 indica que se harán 1000 peticiones

-c 100 indica que se harán 100 peticiones concurrentes

- C

- laravel\_session=eyJpdil6InZrZzRYM2NGb1ZIMVQ1ZzhPZUItcUE9PSIsInZhbHVlIjoizR5SVdPd29xUXk3XC8wSkd5UmQ1cVdsN0hmNFAyVTIRZjN2a0tOeUYrUVJoXC9VSE5leTdxWFicL1VwSDhsK1ZFam9aRDNJTTlpaVNNbnBSMmVqNjRhSGc9PSIsIm1hYyI6IjExYzFiNjIwMjI3Njg0WUyZjI0ZjdYjY3ZjRINmlyMmVmZTU2OTk1OTYyYWNhNDIzOGU5ZGE4NDE3YzEifQ%3D%3D pasa el identificador de sesión en forma de cookie “http://localhost/SistemaVentas-Tesis/laravel/sisVentas/public/ventas/venta” es la URL que vamos a testear.

-

Figura 57. Peticiones al Servidor

```
C:\xampp\apache\bin>ab -g resultados/laravel/results.tsv -n 1000 -c 100 -C laravel_session=eyJpdiI6InZrZzRYM2NGb1Z1MUQ1ZzhpZU1tcUE9PSIsInZhbHVlIjoiazR5SUdPd29xUk3Xc8wSkd5UmQ1cUdsN0hmNFAyUT1RZjN2a0t0eUyrUUJoXC9USE5leTdxWF1cL1UwSDhsK1ZFam9aRDNJTTlpaUNNbnBSMmUqNjRhSGc9PSIsIm1hYyI6IjExYzFiNjIwMjBhMjI3NjgxOWUyZjI0Zjd1YjY3ZjRlNmIyMmUmZTU2OTk1OTYyYWNhNDIzOGU5ZGE4NDE3YzEifQ%3D%3D http://localhost/Sistema
```

Figura 58. Transcurriendo las Peticiones

```
C:\xampp\apache\bin>ab -g resultados/laravel/results.tsv -n 1000 -c 100 -C laravel_session=eyJpdiI6InZrZzRYM2NGb1Z1MUQ1ZzhpZU1tcUE9PSIsInZhbHVlIjoiazR5SUdPd29xUk3Xc8wSkd5UmQ1cUdsN0hmNFAyUT1RZjN2a0t0eUyrUUJoXC9USE5leTdxWF1cL1UwSDhsK1ZFam9aRDNJTTlpaUNNbnBSMmUqNjRhSGc9PSIsIm1hYyI6IjExYzFiNjIwMjBhMjI3NjgxOWUyZjI0Zjd1YjY3ZjRlNmIyMmUmZTU2OTk1OTYyYWNhNDIzOGU5ZGE4NDE3YzEifQ%3D%3D http://localhost/SistemaVentas-Tesis/laravel/sisVentas/public/ventas/venta
This is ApacheBench, Version 2.3 <$Revision: 1757674 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
```

Mientras se ejecuta las peticiones al servidor podemos ir observando el avance del proceso completado de cien en cien peticiones.

Al final del proceso de peticiones al servidor que contiene nuestro aplicativo web se mostró una reseña con los resultados obtenidos.





Figura 59. Resultados Obtenidos con 1000 peticiones

```

Benchmarking localhost <be patient>
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      apache/2.4.25
Server Hostname:     localhost
Server Port:         80

Document Path:       /SistemaVentas-Tesis/laravel/sisVentas/public/ventas/venta
Document Length:     30557 bytes

Concurrency Level:    100
Time taken for tests: 125.376 seconds
Complete requests:    1000
Failed requests:      0
  (Connect: 0, Receive: 0, Length: 88, Exceptions: 0)
Non-2xx responses:    0
Total transferred:    31539144 bytes
HTML transferred:     30528551 bytes
Requests per second: 7.98 [#/sec] <mean>
Time per request:     12537.553 [ms] <mean>
Time per request:     125.376 [ms] <mean, across all concurrent requests>
Transfer rate:        245.66 [Kbytes/sec] received

Connection Times (ms)
  Connect:    min      0      0      2.8      1      55
  Processing: 2631 12275 3350.3 12364 30405
  Waiting:    2630 12238 3350.0 12364 30405
  Total:      2632 12275 3350.0 12365 30405

Percentage of the requests served within a certain time (ms)
 50%    12365
 66%    12945
 75%    13187
 80%    13487
 90%    14579
 95%    15218
 98%    15299.7
 99%    2505.7
100%    30405 <longest request>
    
```

Lo más relevante de estos resultados es lo siguiente:

- a) Requests per second: se refiere a las peticiones atendidas por segundo durante la prueba realizada.
- b) Time per request (mean): es decir el tiempo medio en que el servidor ha tardado en atender a un grupo de peticiones de manera concurrente.
- c) Time per request (mean, across all concurrent requests): se refiere al tiempo medio que el servidor ha tardado en atender una petición de forma individual.

Adicional a los resultados obtenidos mediante la herramienta Apache Bench se utilizó la herramienta Gnuplot el cual es un software en línea de comandos que permite dibujar graficas de funciones a través de fórmulas que las definen.

Se generó el siguiente grafico creando en la ruta del servidor Apache de Xampp:  
**Ruta: C:\xampp\apache\bin** un archivo “plot.p”



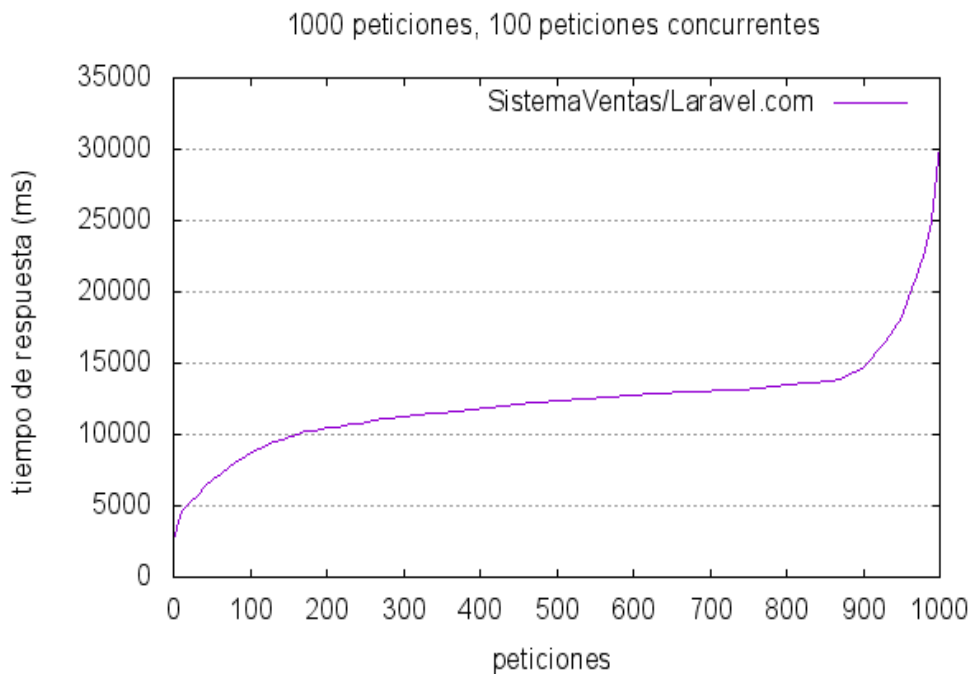
El archivo "plot.p" que se creó contiene las siguientes líneas de código:

```
# output as png image
set terminal png size 600
# save file to "out.png"
set output "results.png"
# graph title
set title "1000 peticiones, 100 peticiones concurrentes"
# nicer aspect ratio for image size
set size ratio 0.6
# y-axis grid
set grid y
# x-axis label
set xlabel "peticiones"
# y-axis label
set ylabel "tiempo de respuesta (ms)"
plot "results_laravel.tsv" using 9 smooth sbezier with lines title
"SistemaVentas/laravel.com"
```

Luego con la herramienta Gnuplot se abrió el archivo antes creado el cual genero la siguiente grafica en dos dimensiones, en el eje X, la cantidad de peticiones las cuales fueron 1000 peticiones por segundo más 100 peticiones concurrentes por segundo; y el eje Y, el tiempo de respuesta a dichas peticiones medido en milisegundos (ms).



Figura 60. Cantidad y Tiempo de Peticiones



En la figura mostrada anteriormente se pudo observar un incremento moderado del tiempo de respuesta cada cien (100) peticiones realizadas al servidor web y una constante del tiempo de respuesta entre las 200 y 900 peticiones realizadas.

Tabla 7. Cantidad de Peticiones y Tiempo de Respuesta

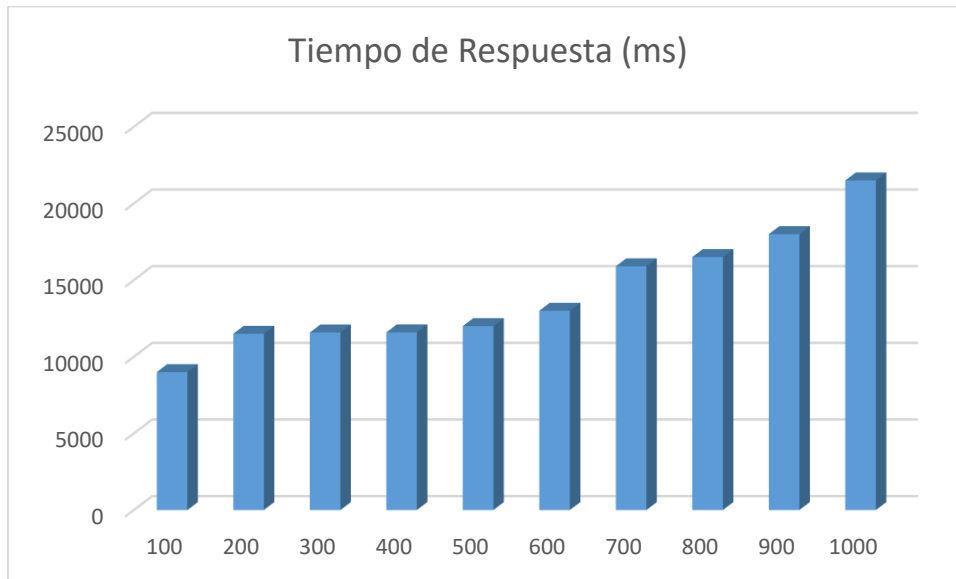
Cantidad de Peticiones al Servidor Web	Tiempo de respuesta en milisegundos (ms) $TR=(TI+TF)/10$
100	8000
200	10000
300	12000



400	12500
500	13000
600	13500
700	13800
800	14000
900	14800
1000	30000
<b><i>Promedio Tiempo de Respuesta (ms)</i></b>	<b>14160</b>

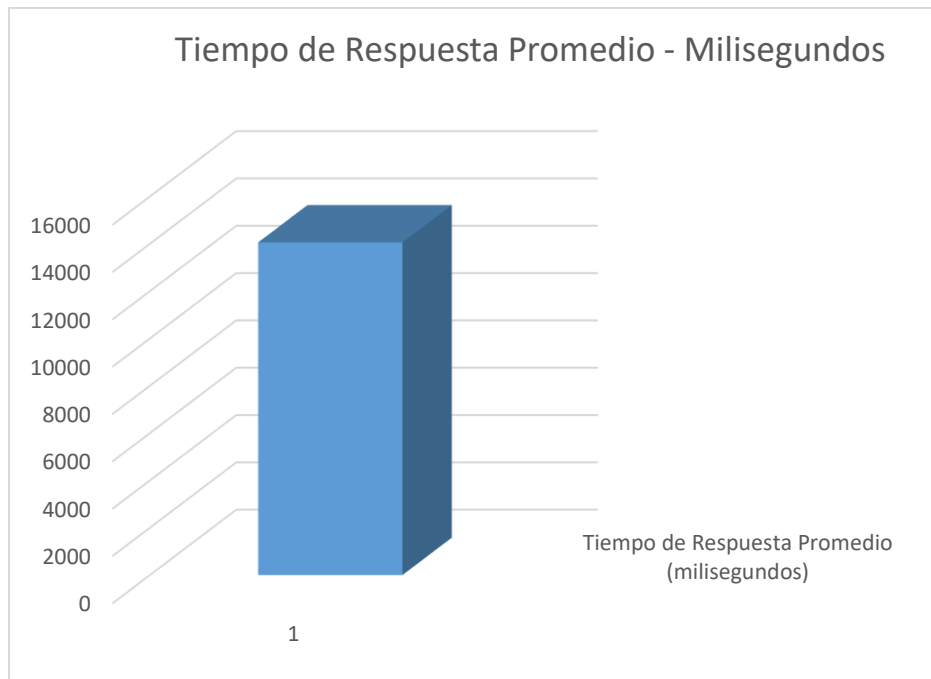


Gráfico 7. Promedio de Tiempo de Respuesta cada 1000 peticiones



Tenemos un Promedio de 14160 (ms) con 1000 peticiones a prueba.

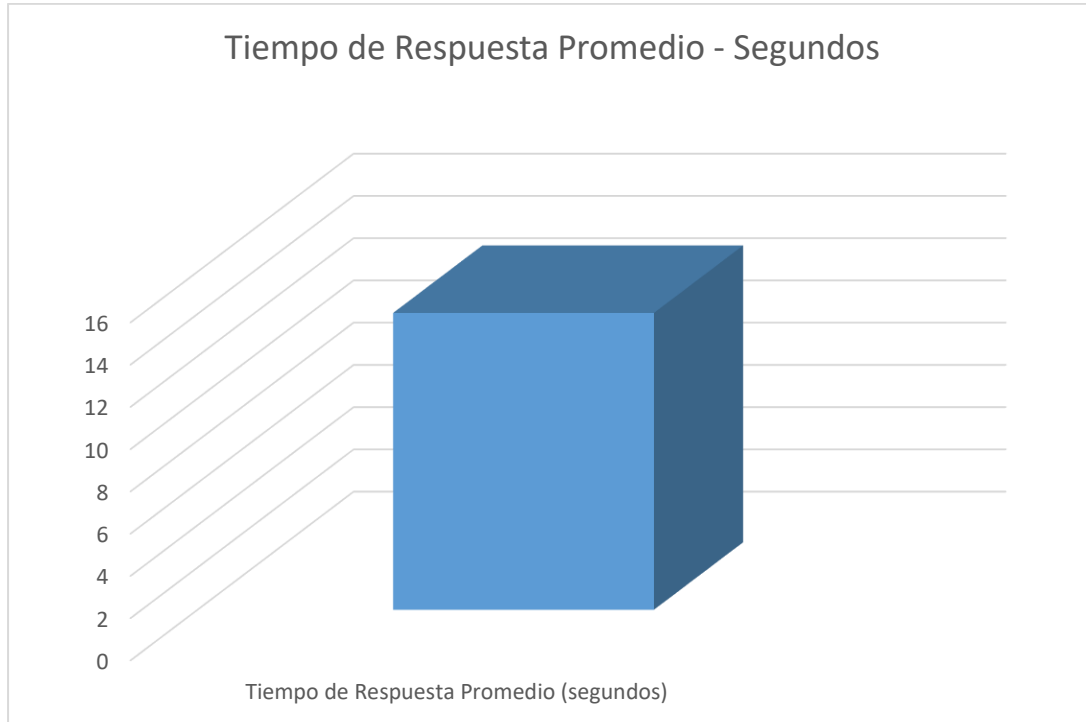
Gráfico 8. Promedio en Tiempo de Respuesta en Milisegundos



Tenemos un Promedio de 14160 (ms)



Gráfico 9.Promedio de Tiempo de Respuesta en Segundos



Tiene un promedio de 14.16 (s).

#### 5.4.2 Evaluación de Framework Laravel en Consumo de Memoria RAM

Para medir este indicador se utilizó una función de php que tiene por nombre, `memory_get_usage()`, la cual devuelve la cantidad de memoria asignada a php en bytes que es asignado a un script de php. La función se utilizó en los módulos de Mantenimiento que incluye Categoría, Marca, Productos, Clientes, Proveedores y Trabajadores, en los cuales se evaluó la función `index`, puesto que es aquella donde se hace un listado de todos los registros ingresados en la base de datos. Para la cual se creó la tabla “Pruebas” que presenta los siguientes atributos: `idprueba`, `memoria_ram_inicial`, `memoria_ram_final`, `memoria_ram_total`.



• **Mantenimiento – Producto**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 38 y línea 114 del código php, el resultado se almacena en la base de datos en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

$$\text{Cálculo de: Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$

Figura 61. Insertando Código de Prueba en Tabla Producto.

```

34
35 <div>
36 <?php
37
38 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
39
40 ?>
41 </div>
92 </table>
93 </div>
94
95 <?php
96 define("LIMITE", 0.2); //200ms
97
98 function medirTiempo($tiempoInicial){
103 }
104
105 $inicio = microtime(true);
106
107 echo medirTiempo($inicio);
108 ?>
109
110 <?php timequery(); ?>
111
112 <?php
113
114 echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
115
116 ?>
117 </div>
118 </div>
119
120 <!--
121

```

• **Mantenimiento – Categoría**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 58 y línea 105 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.



Calculo de:  $Memoria\_ram\_promedio = (memoria\_ram\_inicial + memoria\_ram\_final)/2$

Figura 62 .Codigo de Prueba en la Tabla Categoría

```

55      <div>
56      <?php
57      |
58      echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
59
60      ?>
61      </div>
62      </table>
63      </div>
64      <?php
65      define("LIMITE", 0.2); //200ms
66
67      function medirTiempo($tiempoInicial){
68      }
69
70      $inicio = microtime(true);
71
72      echo medirTiempo($inicio);
73      ?>
74
75      <?php timequery(); ?>
76
77      <?php
78
79      echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
80
81      ?>
82      </div>
83      </div>
84      </div>
85      <!--

```

- **Mantenimiento – Marca**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 22 y línea 34 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$Memoria\_ram\_promedio = (memoria\_ram\_inicial + memoria\_ram\_final)/2$$





Figura 63. Código de Prueba en la Tabla Marca

```

31 <span class="card-title">Listado de Marcas</span><br>
32
33 <div class="row">
34 <div>
35 <?php
36
37 echo ' <pre>Memoria Ram Consumida Inicial: ' . memory_get_usage(). "</pre>";
38
39 ?>
40 </div>
80 </table>
81 </div>
82
83 <?php
84 define("LIMITE", 0.2); //200ms
85
86 function medirTiempo($tiempoInicial){
87     }
88
89 $inicio = microtime(true);
90
91 echo medirTiempo($inicio);
92 ?>
93
94 <?php timequery(); ?>
95
96 <?php
97
98 echo ' <pre>Memoria Ram Consumida Final: ' . memory_get_usage(). "</pre>";
99
100 ?>
101
102
103
104
105

```

• **Mantenimiento – Clientes y Proveedores**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 37 y línea 117 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$



Figura 64 .Codigo de Prueba en la Tabla Persona

```

31 <span class="card-title">Listado de Personas</span><br>
32 <!--@include('mantenimiento.persona.search')-->
33 <div class="row">
34 <div>
35 <?php
36
37 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
38
39 ?>
40 </div>
95 </table>
96 </div>
97
98 <?php
99 define("LIMITE", 0.2); //200ms
100
101 function medirTiempo($tiempoInicial){
102 }
103
104 $inicio = microtime(true);
105
106 echo medirTiempo($inicio);
107 ?>
108
109 <?php timequery(); ?>
110
111 <?php
112
113 echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
114
115 ?>
116
117
118
119
120

```

• **Mantenimiento - Ventas**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 37 y línea 117 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$



Figura 65.Codigo de Prueba en la Tabla Venta

```

31      <span class="card-title">Listado de Ventas</span><br>
32
33      <div class="row">
34      <div>
35      <?php
36
37      echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
38
39      ?>
40      </div>
95      </table>
96      </div>
97
98      <?php
99      define("LIMITE", 0.2); //200ms
100
101      function medirTiempo($tiempoInicial){
106      }
107
108      $inicio = microtime(true);
109
110      echo medirTiempo($inicio);
111      ?>
112
113      <?php timequery(); ?>
114
115      <?php
116
117      echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
118
119      ?>
122      </div>
123      </div>
    
```

Tabla 8. Uso de Memoria RAM en Tablas del Sistema

Tablas	MEMORIA RAM INICIAL (bytes) – MRI	MEMORIA RAM FINAL (bytes) – MRF	MEMORIA_RAM_PROMEDIO (bytes) – MRP MRP = (MRI + MRF)/2
<b>Categoría</b>	8122144	8165992	8144068
<b>Marca</b>	7876976	8170408	8023692
<b>Producto</b>	7953288	8246616	8099952
<b>Persona</b>	8150080	8193928	8172004



<b>Venta</b>	7985328	8278 672	8132000
--------------	---------	-------------	---------

MEMORIA_RAM_TOTAL	Resultado (bytes)				
	Categoría	Marca	Producto	Persona	Venta
<b>Cantidad de memoria RAM asignada a php en los formularios evaluados.</b>	8144068	8023692	8099952	8172004	8132000
<b>TOTAL</b>	8114343,2 bytes <> 8,1143432 Megabyte (MB)				

Tabla 9.Total de memoria RAM asignado a formularios

Gráfico 10.Total de Memoria RAM en tablas hechas con Laravel

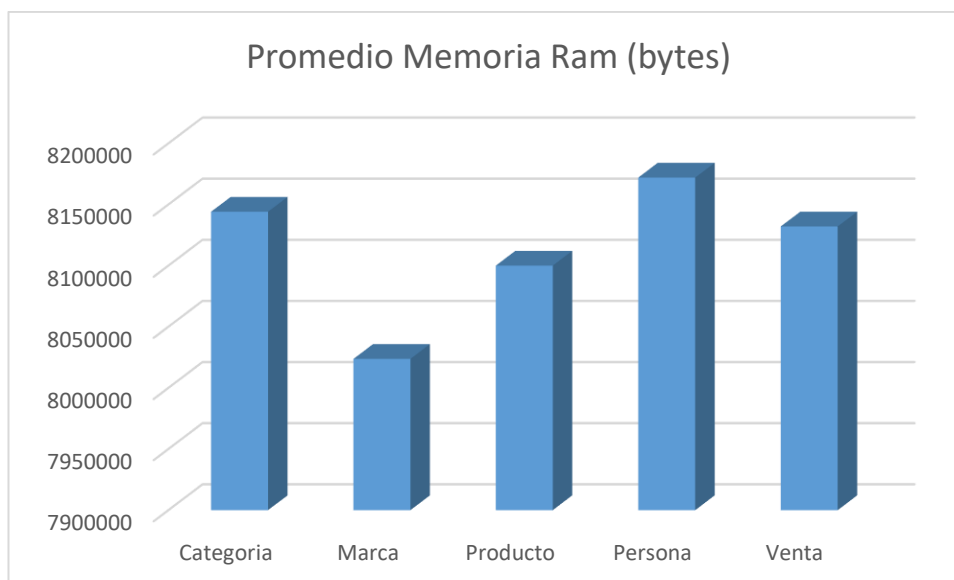
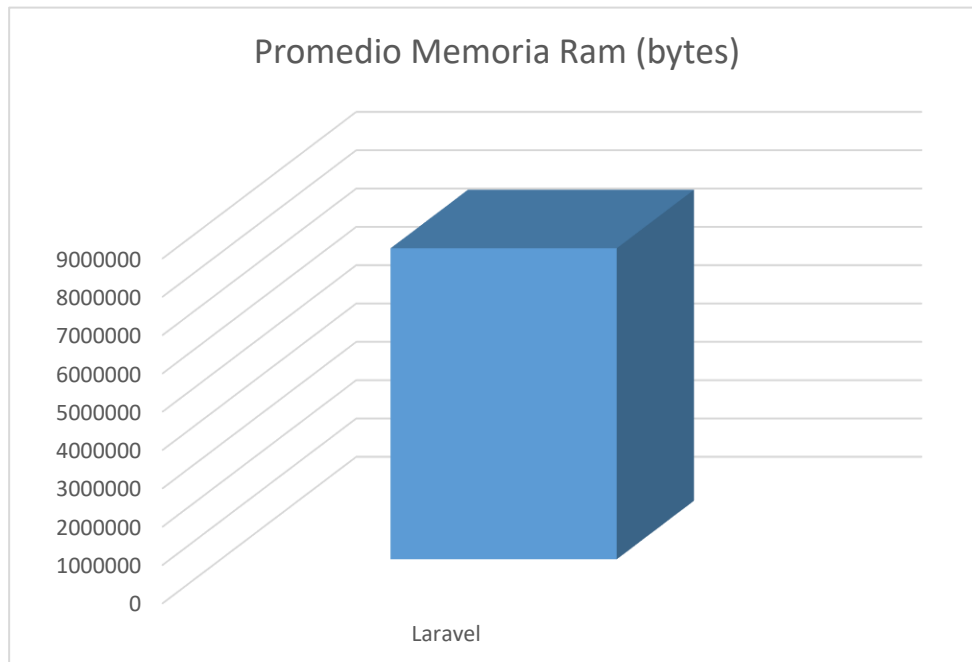
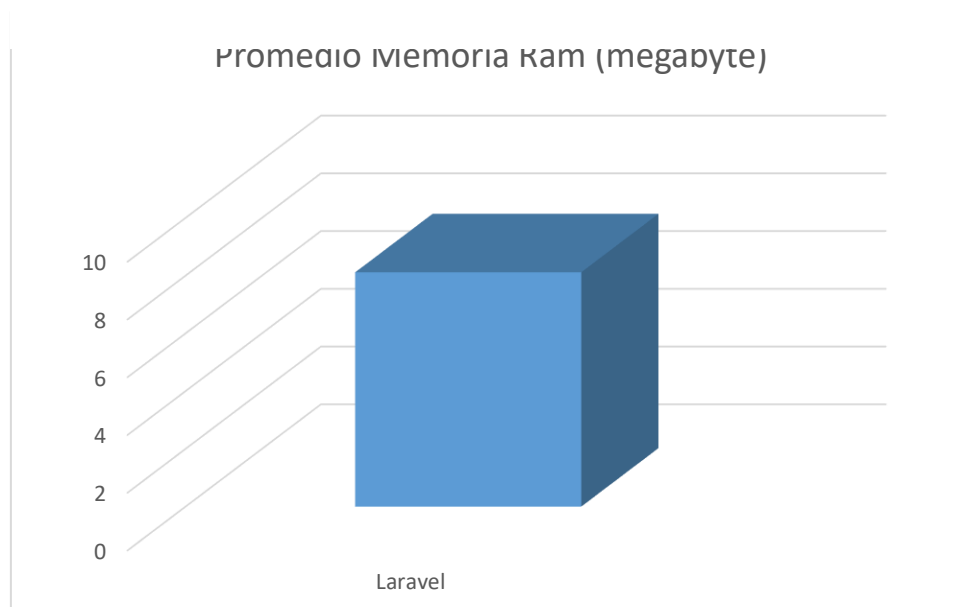


Gráfico 11. Promedio de Memoria RAM total consumida por el Framework Laravel (bytes)



Como resultado en Consumo de memoria RAM con el framework Laravel se obtiene 8114343,2 bytes

Gráfico 12. Promedio de Memoria RAM en Laravel



Como resultado en Consumo de memoria RAM con el framework Laravel se obtiene 8,1143432 megabytes



### 5.4.3 Evaluación de Framework Laravel en Tiempo de Carga

Para medir este indicador se utilizó una función de php que tiene por nombre, `microtime()`, la cual devuelve el tiempo en microsegundos que es asignado a un script de php. La función se utilizó en los módulos de Mantenimiento que incluye Categoría, Marca, Productos, Persona, Venta, en los cuales se evaluó la función `index`, puesto que es aquella donde se hace un listado de todos los registros ingresados en la base de datos. Para la cual se creó la tabla "Pruebas\_Carga" que presenta los siguientes atributos: `idpruebacarga`, `framework`, `tabla`, `tiempo_carga_inicial`, `tiempo_carga_final`, `tiempo_carga_promedio`

#### • Mantenimiento – Categoría

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 41 y línea 105 del código php, el resultado se almacena en la base de datos específicamente en la tabla `pruebas`, la cual contiene los atributos de: `idpruebacarga`, `framework`, `tabla`, `tiempo_carga_inicial`, `tiempo_carga_final`, `tiempo_carga_promedio`.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$

Figura 66 .Prueba de Carga a la tabla Categoría

```

37
38     <div>
39     <?php
40     $startTime = microtime(true);
41     echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
42     ?>
43     </div>
44
45     <div class="row">
46
47     <div>
48     <?php
49
50     echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage(). "</pre>";
51     </div>
52
53     </tbody>
54     </table>
55
56     </div>
57
58     <div>
59     <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
60     </div>
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88

```

• **Mantenimiento – Marca**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 36 y línea 92 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$



Figura 67 .Prueba de Carga a la tabla Marca

```

34 <?php
35 $startTime = microtime(true);
36 echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
37 ?>
38 </div>
39
40
41 <div class="row">
42 <div>
43 <?php
44
45 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage() . "</pre>";
46 </div>
47 </div>
48 </table>
49
50 </div>
51
52 <div>
53 <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
54 </div>
55
56 define("LIMITE", 0.2); //200ms
57
58 function medirTiempo($tiempoInicial){
59
60 }
61
62 $inicio = microtime(true);
63
64 echo medirTiempo($inicio);
65 ?>
66
67 <?php timequery(); ?>
68
69 <?php

```

• **Mantenimiento – Producto**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 36 y línea 105 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$





Figura 62 Prueba de Carga a la tabla Producto

```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
49
100
101
102
103
104
105
106
107
108
109
110
111
--
<div id="VerListado">
  <div class="card">
    <div class="card-content">
      <span class="card-title">Listado de Productos</span><br>
    </div>
    <?php
      $startTime = microtime(true);
      echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
      ?>
    </div>
    <div class="row">
      <div>
        <?php
          echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage() . "</pre>";
        </div>
      </div>
    </table>
  </div>
  <div>
    <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
  </div>
  <?php
    define("LIMITE", 0.2); //200ms
  </?php
  
```

• **Mantenimiento – Persona**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 38 y línea 108 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$



Figura 63. Prueba de Carga a la tabla Persona

```

34
35
36
37 <?php
38 $startTime = microtime(true);
39 echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
40 ?>
41 </div>
42
43 <div class="row">
44 <div>
45 <?php
46
47 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage() . "</pre>";
48 ?>
49 </div>
50 </div> </table>
51
52 <div>
53 <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
54 </div>
55
56 <?php
57 define("LIMITE", 0.2); //200ms
58 function medirTiempo($tiempoInicial){
59 }
60
61 $inicio = microtime(true);

```

• **Mantenimiento – Venta**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 36 y línea 107 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$



Figura 64. Prueba de Carga a la tabla Venta

```

31 <span class="card-title">Listado de Ventas</span><br>
32
33 <div>
34 <?php
35 $startTime = microtime(true);
36 echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
37 ?>
38 </div>
39
40 <div class="row">
41 <?php
42
43
44 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage() . "</pre>";
45 </div>
46
47 </table>
48
49 <div>
50 <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
51 </div>
52
53 <?php
54 define("LIMITE", 0.2); //200ms
55
56 function medirTiempo($tiempoInicial){
57     $final = microtime(true);
58     $total = $final - $tiempoInicial;
59     $faltante = LIMITE - $total;
60     return '<pre>Tiempo de ejecución: ' . $total . " segundos\n\r quedan disr

```

Tabla 10. Tiempo de Carga en Tablas del Sistema

Tablas	TIEMPO CARGA INICIAL (microsegundos) –TCI	TIEMPO CARGA FINAL (microsegundos)-TCF	TIEMPO_CARGA_PROM EDIO (microsegundos) – TCP TCP = (TCI + TCF)/2
<b>Categoría</b>	1499094372.0098	1499094372.0385	1499094372,02415
<b>Marca</b>	1499094131.2732	1499094131.3168	1499094131,295
<b>Producto</b>	1499094609.7711	1499094609.8602	1499094609,81565
<b>Persona</b>	1499094957.9902	1499094958.0509	1499094958,02055
<b>Venta</b>	1499095211.0266	1499095211.0869	1499095211,05675



TIEMPO_CARGA_TOTAL	Resultado (microsegundos)				
	Categoría	Marca	Producto	Persona	Venta
Tiempo en microsegundos asignada a php en los formularios evaluados.	1499094372.02415	1499094131.295	1499094609.81565	1499094958.02055	1499095211.05675
<b>TOTAL</b>	149909,46564 Microsegundos ( $\mu$ s) <> 1,4990946564 Segundos (s)				

Tabla 11.Total de Tiempo de Carga asignado a formularios

Gráfico 13.Total de Tiempo de Carga en tablas hechas con Laravel

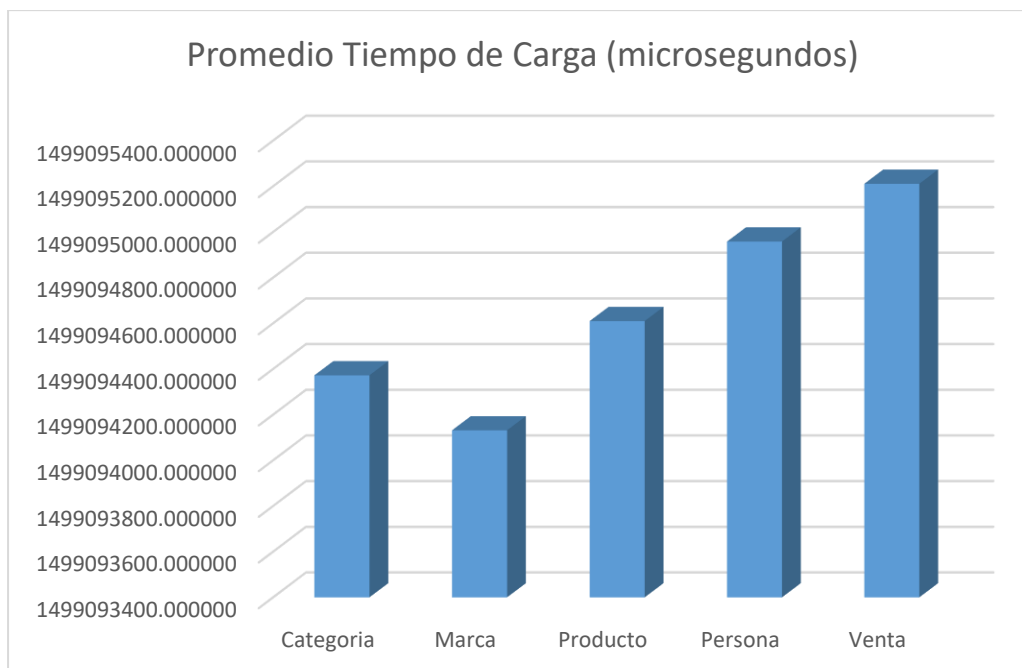
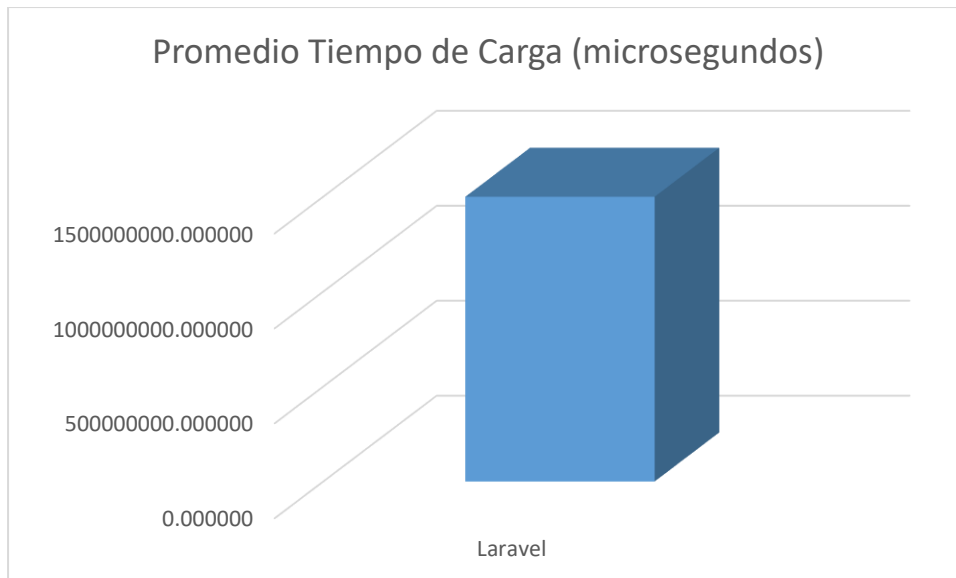


Gráfico 14. Promedio de Tiempo de Carga total consumida por el Framework Laravel (microsegundos)



Como resultado en Tiempo de Carga con el framework Laravel se obtiene 149909,46564 μs

#### 5.4.4 Evaluación de Framework Yii en Tiempo de Respuesta

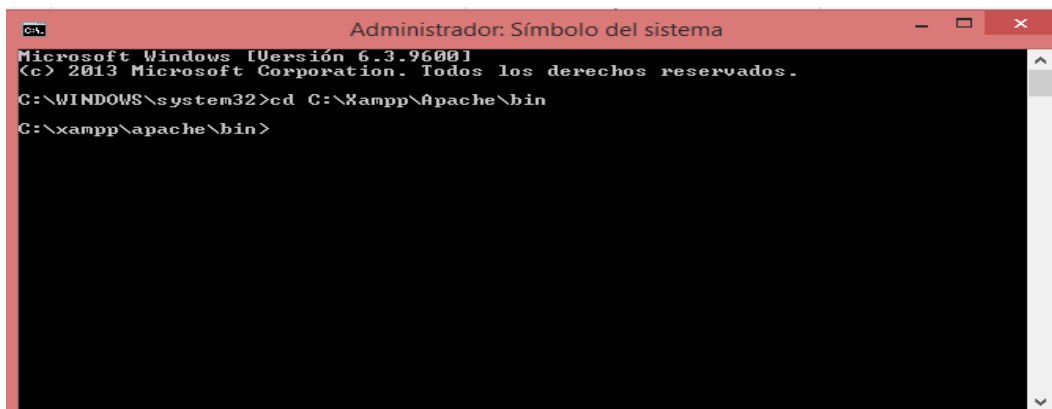
Para medir este indicador se utilizó un programa de línea de comandos de nombre Apache Bench, esta herramienta ab viene con la distribución estándar de Apache y al igual que el mismo servidor web Apache, que se define como un software libre, y distribuido bajo los términos de la Licencia Apache.

En primer lugar lo que se hizo es abrir una terminal de línea de comandos CMD, e ingresar al directorio donde se encuentra la carpeta Apache en un nuestro servidor local Tomcat.

Línea de Comando: `cd C:\Xampp\Apache\bin`



Figura 65.Ruta de Apache



Luego se ingresó el comando: “ab -g resultados/yii/results.tsv -n 1000 -c 100 -C PHPSESSID=blhen3u2gffhbq3fbr3pi6aom5 http://localhost/SistemaVentas-Tesis/Yii/sisventas/web/index.php?r=ingreso%2Fviewingreso “

Este comando forma parte de la herramienta Apache Bench y se compone de las siguientes partes:

Donde:

ab es Apache Bench

-g resultados/yii/results.tsv guarda los resultados en el archivo results.tsv

-n 1000 indica que se harán 1000 peticiones

-c 100 indica que se harán 100 peticiones concurrentes

-C PHPSESSID = blhen3u2gffhbq3fbr3pi6aom5 pasa el identificador de sesión en forma de cookie

http://localhost/SistemaVentas-

Tesis/Yii/sisventas/web/index.php?r=ingreso%2Fviewingreso, es la URL que vamos a testear.

Figura 66.URL a Testear

```
C:\xampp\apache\bin>ab -g resultados/yii/results.tsv -n 1000 -c 100 -C PHPSESSID
=blhen3u2gffhbq3fbr3pi6aom5 http://localhost/SistemaVentas-Tesis/Yii/sisventas/w
eb/index.php?r=ingreso%2Fviewingreso
This is ApacheBench, Version 2.3 <$Revision: 1757674 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
```

Figura 67.Prueba con Peticiones al Servidor

```
=blhen3u2gffhbq3fbr3pi6aom5 http://localhost/SistemaVentas-Tesis/Yii/sisventas/w
eb/index.php?r=ingreso%2Fviewingreso
This is ApacheBench, Version 2.3 <$Revision: 1757674 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
```

Mientras se ejecuta las peticiones al servidor podemos ir observando el avance del proceso completado de cien en cien peticiones.



Figura 68. Prueba 1000 Peticiones

```

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests

Server Software:      Apache/2.4.25
Server Hostname:     localhost
Server Port:         80

Document Path:       /SistemaVentas-Tesis/Yii/sisventas/web/index.php?r=ingreso%2Fviewingreso
Document Length:     27762 bytes

Concurrency Level:   100
Time taken for tests: 138.790 seconds
Complete requests:   1000
Failed requests:     77
  (Connect: 0, Receive: 0, Length: 77, Exceptions: 0)
Total transferred:   28348571 bytes
HTML transferred:    27761923 bytes
Requests per second: 7.21 [#/sec] (mean)
Time per request:    13878.963 [ms] (mean)
Time per request:    138.790 [ms] (mean, across all concurrent requests)
Transfer rate:       199.47 [Kbytes/sec] received

Connection Times (ms)
  min      mean[+/-sd] median    max
Connect:    0      0      2.3      0
Processing: 2106   13433 3391.5   12193 21588
Waiting:    1933   13404 3389.5   12168 21555
Total:      2106   13434 3391.3   12193 21589

Percentage of the requests served within a certain time (ms)
 50%    12193
 66%    14927
 75%    16613
 80%    16830
 90%    17962
 95%    19519
 98%    20696
 99%    21132
100%    21589 (longest request)
    
```

Al final del proceso de peticiones al servidor que contiene nuestro aplicativo web se mostró una reseña con los resultados obtenidos.

Lo más relevante de estos resultados es lo siguiente:

- a) Requests per second: se refiere a las peticiones atendidas por segundo durante la prueba realizada.
- b) Time per request (mean): es decir el tiempo medio en que el servidor ha tardado en atender a un grupo de peticiones de manera concurrente.
- c) Time per request (mean, across all concurrent requests): se refiere al tiempo medio que el servidor ha tardado en atender una petición de forma individual.





Adicional a los resultados obtenidos mediante la herramienta Apache Bench se utilizó la herramienta Gnuplot el cual es un software en línea de comandos que permite dibujar graficas de funciones a través de fórmulas que las definen.

Se generó el siguiente grafico creando un archivo "plot.p" en la ruta del servidor Apache de Xampp:

Ruta: **C:\xampp\apache\bin**

El archivo "plot.p" que se creó contiene las siguientes líneas de código:

```
# output as png image
set terminal png size 600
# save file to "out.png"
set output "results_yii.png"
# graph title
set title "1000 peticiones, 100 peticiones concurrentes"

# nicer aspect ratio for image size
set size ratio 0.6
# y-axis grid
set grid y
# x-axis label

set xlabel "peticiones"
# y-axis label

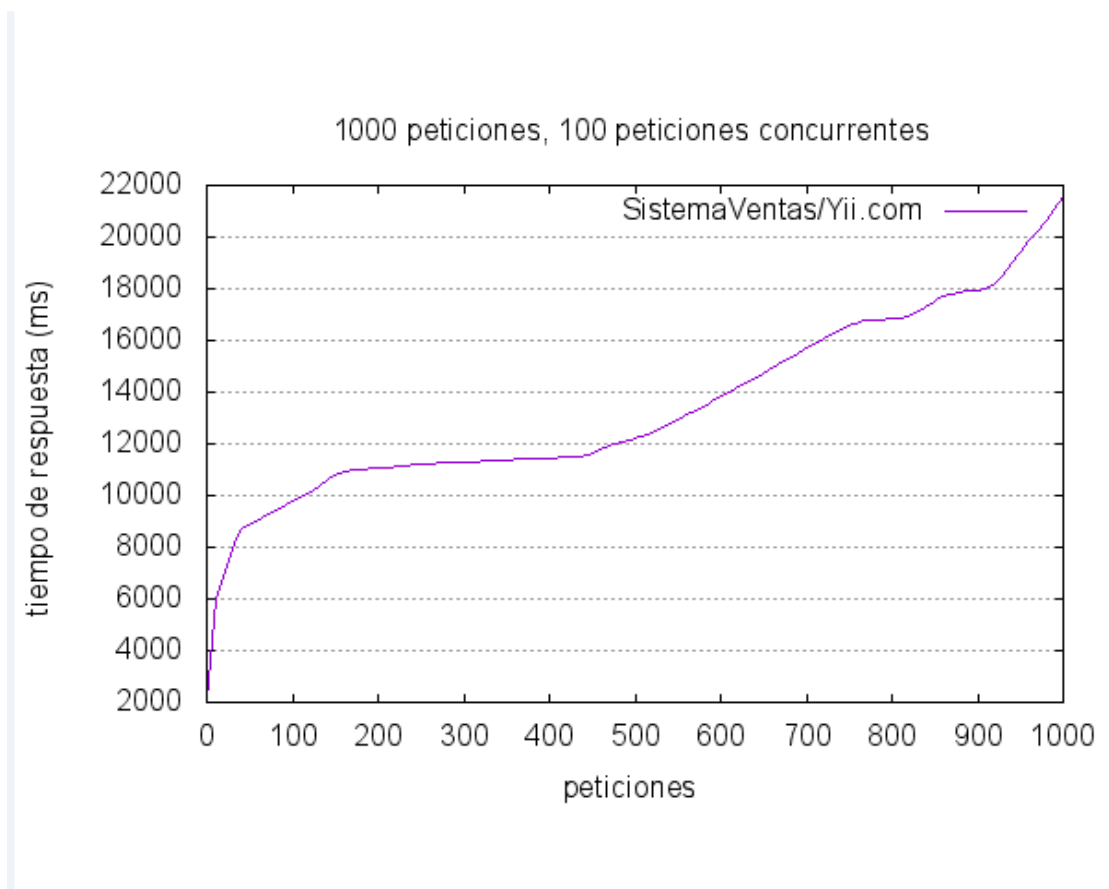
set ylabel "tiempo de respuesta (ms)"

plot "results_yii.tsv" using 9 smooth sbezier with lines title
"SistemaVentas/Yii.com"
```



Luego con la herramienta Gnuplot se abrió el archivo antes creado el cual genero la siguiente grafica en dos dimensiones, en el eje X, la cantidad de peticiones las cuales fueron 1000 peticiones por segundo más 100 peticiones concurrentes por segundo; y el eje Y, el tiempo de respuesta a dichas peticiones medido en milisegundos (ms).

Figura 69. Tiempo de Respuesta y Peticiones



En la cual se pudo observar un incremento del tiempo de respuesta de peticiones realizadas al servidor web en las 200 primeras peticiones y una constante del tiempo de respuesta entre las 200 y 500 peticiones realizada, luego un incremento acelerado del tiempo de respuesta entre las 500 y 1000 peticiones.



Tabla 12. Promedio de Tiempo de Respuesta

<b>Cantidad de Peticiones al Servidor Web</b>	<b>Tiempo de respuesta en milisegundos (ms)</b> <b>TR=(TI+TF)/10</b>
100	9000
200	11500
300	11580
400	11590
500	12000
600	13000
700	15900
800	16500
900	18000
1000	21500
<b>Promedio Tiempo de Respuesta (ms)</b>	<b>14057</b>

Tenemos un Promedio de 14057 (ms) que equivale a 14.05 (s), con 1000 peticiones a prueba.



Gráfico 15. Promedio de Tiempo de Respuesta cada 100 peticiones

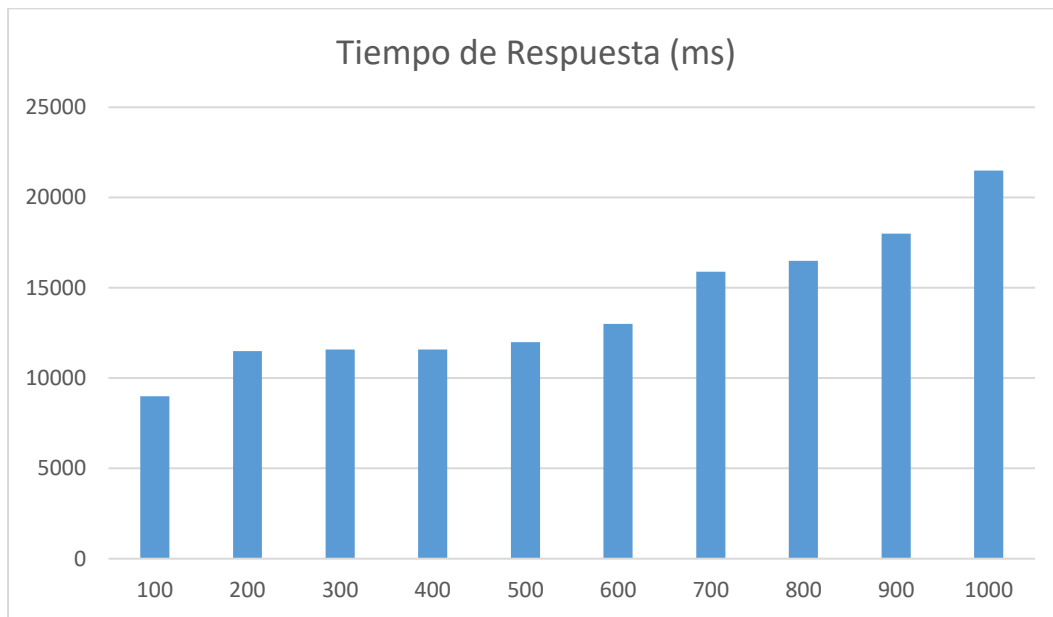
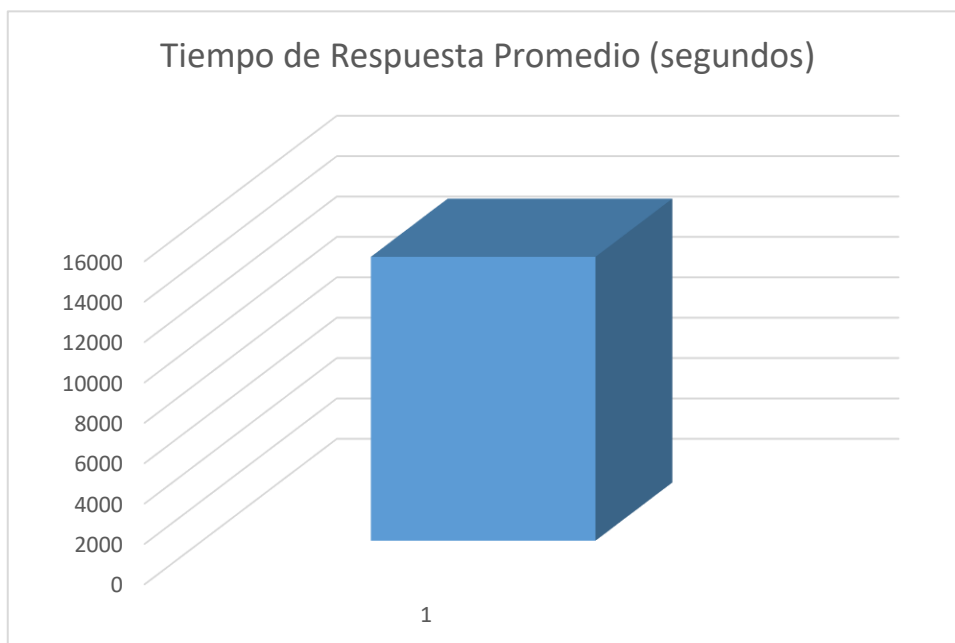


Gráfico 16. Promedio de Tiempo de Respuesta



Como resultado en Tiempo de Respuesta con el framework Yii se obtiene 14.05 segundos



#### 5.4.5 Evaluación de Framework Yii en Consumo de Memoria RAM

Para medir este indicador se utilizó una función de php que tiene por nombre, `memory_get_usage ()`, la cual devuelve la cantidad de memoria asignada a php en bytes que es asignado a un script de php. La función se utilizó en los módulos de Mantenimiento que incluye Categoría, Marca, Productos, Clientes, Proveedores y Trabajadores, en los cuales se evaluó la función `index`, puesto que es aquella donde se hace un listado de todos los registros ingresados en la base de datos. Para la cual se creó la tabla “Pruebas” que presenta los siguientes atributos: `idprueba`, `memoria_ram_inicial`, `memoria_ram_final`, `memoria_ram_total`.

##### • Mantenimiento – Categoría

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 87 y línea 122 del código php, el resultado se almacena en la base de datos específicamente en la tabla `pruebas`, la cual contiene los atributos de: `memoria_ram_inicial`, `memoria_ram_final`, `memoria_ram_promedio`.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$

Figura 70. Memoria RAM de tabla Categoría

```

viewmarca.php x viewproducto.php x viewpersona.php x viewventa.php x
49 <div>
50 <?php
51
52         echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
53
54     ?>
55 </div>
56
57 <table class="table table-bordered">
58 </table>
59
60 <?= LinkPager::widget([
61     "pagination" => $pages,
62 ])
63 ?>
64
65 <?php
66
67     ?>
68
69     <?php timequery(); ?>
70
71     <?php
72
73         echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
74
75     ?>
76

```

• **Mantenimiento – Marca**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 59 y línea 129 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$



Figura 71. Prueba de memoria RAM a la tabla Marca

```

55     <div>
56
57         <?php
58
59         echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
60
61         ?>
62     </div>
63     <tbody>
101    </tbody>
102 </tr>
103 </table>
104
105 <?= LinkPager::widget([
106     "pagination" => $pages,
107 ])
108 ?>
109
110 <?php
123
124
125     <?php timequery(); ?>
126
127     <?php
128
129     echo '<pre>Memoria Ram Consumida: ' . memory_get_usage()."</pre>";
130
131     ?>

```

• **Mantenimiento – Producto**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 53 y línea 155 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$



Figura 72. Prueba de memoria RAM a la tabla Producto

```

46
47
48 <h3>Lista de productos</h3>
49
50 <div>
51 <?php
52
53         echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
54
55     ?>
56 </div>
57
58 <table class="table table-bordered">
59     <tr>
128     </tr>
129 </table>
130
131 <?= LinkPager::widget([
132     "pagination" => $pages,
133 ])
134 ?>
135
136 <?php
149         ?>
150
151         <?php timequery(); ?>
152
153         <?php
154
155         echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
156
157     ?>
    
```

- **Mantenimiento – Persona**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 53 y línea 155 del código php, el resultado se almacena específicamente en la base de datos en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$





Figura 73. Prueba de memoria RAM a la tabla Persona

```

52
53         echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
54
55     ?>
56 </div>
57
58 <table class="table table-bordered">
59     <tr>
128     </tr>
129 </table>
130
131 <?= LinkPager::widget([
132     "pagination" => $pages,
133 ])
134 ?>
135
136 <?php
149     ?>
150
151     <?php timequery(); ?>
152
153     <?php
154
155     echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
156
157     ?>

```

• **Mantenimiento - Venta**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 52 y línea 124 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: memoria\_ram\_inicial, memoria\_ram\_final, memoria\_ram\_promedio.

Calculo de:

$$\text{Memoria\_ram\_promedio} = (\text{memoria\_ram\_inicial} + \text{memoria\_ram\_final})/2$$



Figura 74. Prueba de memoria RAM a la tabla Venta

```

viewmarca.php x viewproducto.php x viewpersona.php x viewventa.php x
43 <?php $f->end() ?>
44
45 <h3><? $search ?></h3>
46
47 <h3>Listado de ventas</h3>
48
49 <div>
50 <?php
51
52 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
53
54 ?>
55 </div>
56
57 <table class="table table-bordered">
58 </table>
59
100 <?= LinkPager::widget([
101     "pagination" => $pages,
102 ])
103 ?>
104
105 <?php
106
107 ?>
108
109
110 <?php timequery(); ?>
111
112
113 <?php
114
115 echo '<pre>Memoria Ram Consumida Final: ' . memory_get_usage()."</pre>";
116
117 ?>
118
119
120
121
122
123
124
125
126

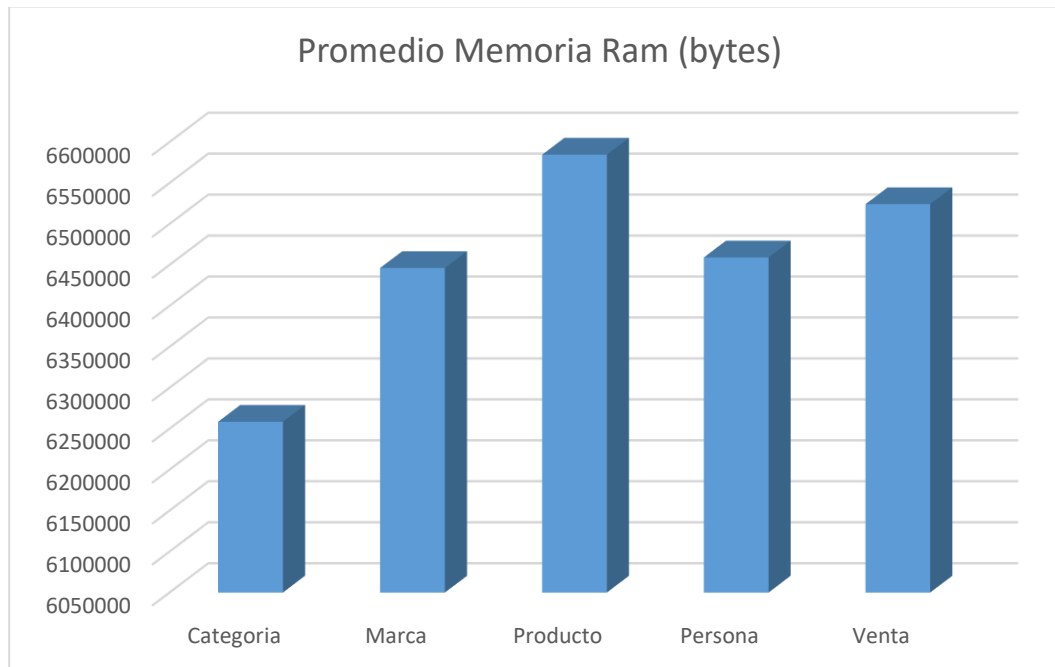
```

Tabla 13. Consumo de Memoria RAM por cada Tabla

MANTENIMIENTO	MEMORIA RAM INICIAL (bytes) – MRI	MEMORIA RAM FINAL (bytes)- MRF	MEMORIA_RAM_PROMEDIO (bytes) – MRP MRP = (MRI + MRF)/2
<b>Categoría</b>	6201284	6315660	6258472
<b>Marca</b>	6423904	6468488	6446196
<b>Producto</b>	6561384	6607680	6584532
<b>Persona</b>	6436312	6481880	6459096
<b>Venta</b>	6501584	6546680	6524132



Gráfico 17.Promedio Uso de Memoria RAM en tablas hechas con Yii

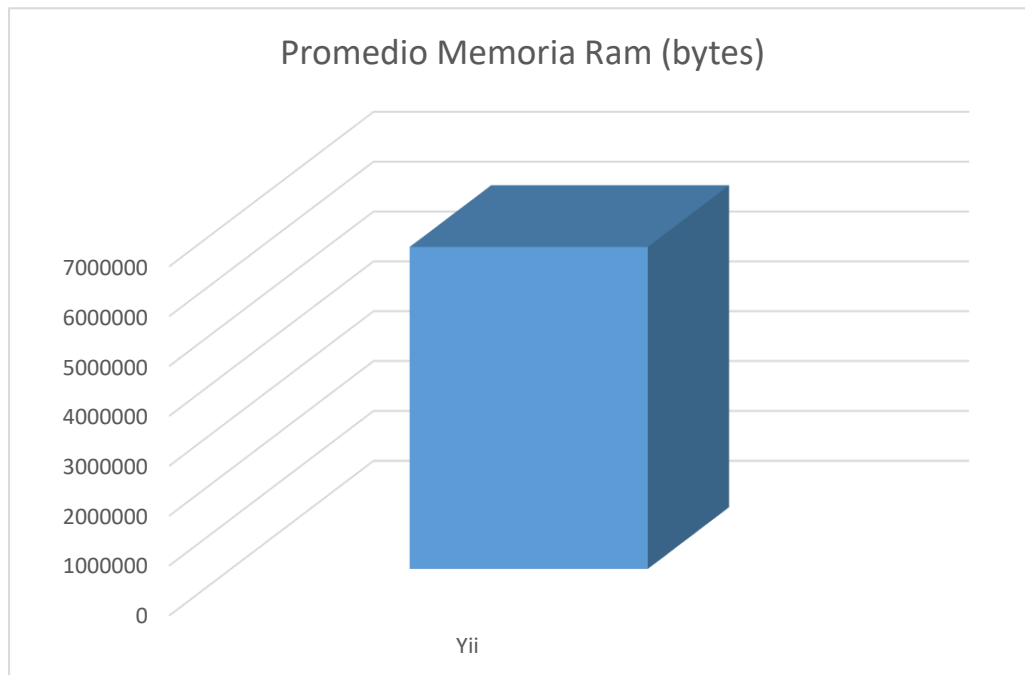


	Resultados (bytes)				
	Categoría	Marca	Producto	Persona	Venta
<b>MEMORIA_RAM_PROMEDIO.</b>	6258472	6446196	6584532	6459096	6524132
<b>TOTAL</b>	6454485,6 bytes <> 6,4544856 Megabyte (MB)				

Tabla 14.Total de Memoria RAM

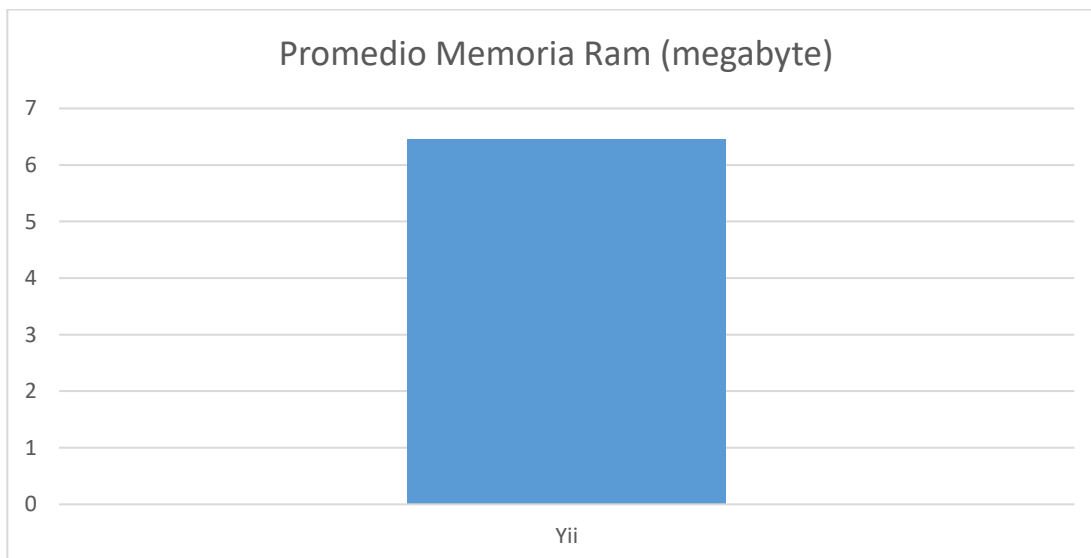


Gráfico 18.Promedio Memoria RAM



Como resultado en uso de Memoria RAM con el framework Laravel se obtiene 6454485,6 bytes

Gráfico 19.Promedio Memoria RAM



Como resultado en uso de Memoria RAM con el framework Yii se obtiene 6,4544856 Megabyte

#### 5.4.6 Evaluación de Framework Yii en Tiempo de Carga

Para medir este indicador se utilizó una función de php que tiene por nombre, `microtime()`, la cual devuelve el tiempo en microsegundos que es asignado a un script de php. La función se utilizó en los módulos de Mantenimiento que incluye Categoría, Marca, Productos, Persona, Venta, en los cuales se evaluó la función `index`, puesto que es aquella donde se hace un listado de todos los registros ingresados en la base de datos. Para la cual se creó la tabla "Pruebas\_Carga" que presenta los siguientes atributos: `idpruebacarga`, `framework`, `tabla`, `tiempo_carga_inicial`, `tiempo_carga_final`, `tiempo_carga_promedio`

##### • Mantenimiento – Categoría

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 41 y línea 105 del código php, el resultado se almacena en la base de datos específicamente en la tabla `pruebas`, la cual contiene los atributos de: `idpruebacarga`, `framework`, `tabla`, `tiempo_carga_inicial`, `tiempo_carga_final`, `tiempo_carga_promedio`.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$

Figura 75. Prueba de Carga a la tabla Categoría

```

37
38     <div>
39     <?php
40     $startTime = microtime(true);
41     echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
42     ?>
43     </div>
44
45     <div class="row">
46
47     <div>
48     <?php
49
50     echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage(). "</pre>";
51     </div>
52
53     </tbody>
54     </table>
55
56     </div>
57
58     <div>
59     <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
60     </div>
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108

```

• **Mantenimiento – Marca**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 52 y línea 112 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$



Figura 76. Prueba de Carga a la tabla Marca

```

49 <div>
50 <?php
51 $startTime = microtime(true);
52 echo '<pre>Tiempo de carga Inicial: ' . $startTime . "</pre>";
53 ?>
54 </div>
55
56 <table class="table table-bordered">
57 <tr>
58 <th>Marca</th>
59 <th>Descripción</th>
60 <th>Condición</th>
61 <th>Opciones</th>
62 <div>
63
64 <?php
65 echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage() . "</pre>";
66 ?>
67
68 </div>
69 </tbody>
70 </tr>
108 </table>
109
110 </div>
111 <div>
112 <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
113 </div>
114
115 <?= LinkPager::widget([
116     "pagination" => $pages,]
117 )

```

• **Mantenimiento – Producto**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 61 y línea 138 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$



Figura 77. Prueba de Carga a la tabla Producto

```

46
47
48 <h3>Lista de productos</h3>
49
50 <div>
51 <?php
52
53         echo '<pre>Memoria Ram Consumida Inicial: ' . memory_get_usage()."</pre>";
54
55     ?>
56 </div>
57
58 <div>
59
60     <?php
61     $startTime = microtime(true);
62     echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
63     ?>
64 </div>
65 <table class="table table-bordered">
66 </table>
136 </table>
137 <div>
138     <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; ?>
139 </div>
140
141 <?= LinkPager::widget([
142     "pagination" => $pages,
143 ])
144 ?>
145
146 <?php
147     define("LIMITE", 0.2); //200ms
    
```

• **Mantenimiento – Persona**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 52 y línea 126 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$





Figura 78. Prueba de Carga a la tabla Persona

```

46
47 <h3>Lista de Personas</h3>
48
49 <div>
50 <?php
51     $startTime = microtime(true);
52     echo '<pre>Tiempo de carga Inicial: ' . $startTime . "</pre>";
53 >
54 </div>
55
56 <table class="table table-bordered">
57 <tr>
58 </tr>
59 </table>
60
61 <div>
62 <?php echo '<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"; >
63 </div>
64
65 <?= LinkPager::widget([
66     "pagination" => $pages,
67 ])
68 >
69
70 <?php
71     define("LIMITE", 0.2); //200ms
72
73     function medirTiempo($tiempoInicial){
74         $final = microtime(true);
75         $total = $final - $tiempoInicial;
76         $faltante = LIMITE - $total;
77         return '<pre>Tiempo de ejecución: ' . $total . " segundos\n\r quedan dis:

```

• **Mantenimiento – Venta**

Aquí se aplicó la función antes y después de ejecutar la consulta que muestra la lista completa de productos y la observamos en la línea 60 y línea 107 del código php, el resultado se almacena en la base de datos específicamente en la tabla pruebas, la cual contiene los atributos de: idpruebacarga, framework, tabla, tiempo\_carga\_inicial, tiempo\_carga\_final, tiempo\_carga\_promedio.

Calculo de:

$$\text{tiempo\_carga\_promedio} = (\text{tiempo\_carga\_inicial} + \text{tiempo\_carga\_final})/2$$



Figura 79. Prueba de Carga a la tabla Venta

```

55 </div>
56
57 <div>
58     <?php
59         $startTime = microtime(true);
60         echo ('<pre>Tiempo de carga Inicial: ' . $startTime) . "</pre>";
61     >
62 </div>
63
64 <table class="table table-bordered">
65     <tr>
104     </tr>
105 </table>
106 <div>
107     <?php echo ('<pre>Tiempo de carga Final: ' . microtime(true) . "</pre>"); >
108 </div>
109
110 <?= LinkPager::widget([
111     "pagination" => $pages,
112 ])
113 ?>
114
115 <?php
116     define("LIMITE", 0.2); //200ms
117
118     function medirTiempo($tiempoInicial){
119         $final = microtime(true);
120         $total = $final - $tiempoInicial;
121         $faltante = LIMITE - $total;
122         return ('<pre>Tiempo de ejecución: ' . $total . " segundos\n\r quedan dis
123
124

```

Tabla 15. Tiempo de Carga en Tablas del Sistema

Tablas	TIEMPO CARGA INICIAL (microsegundos) –TCI	TIEMPO CARGA FINAL (microsegundos)-TCF	TIEMPO_CARGA_PROMEDIO (microsegundos) – TCP TCP = (TCI + TCF)/2
<b>Categoría</b>	1499098355.3404	1499098355.3423	1499098355.34135
<b>Marca</b>	1499097111.3575	1499097111.3587	1499097111.3581
<b>Producto</b>	1499097832.2554	1499097832.2573	1499097832.25635
<b>Persona</b>	1499098100.0519	1499098100.0529	1499098100.0524
<b>Venta</b>	1499098355.6409	1499098355.6413	1499098355.6411



Tabla 16.Total de Tiempo de Carga asignado a formularios

TIEMPO_CARGA_TOTAL	Resultado (microsegundos)				
	Categoría	Marca	Producto	Persona	Venta
Tiempo en microsegundos asignada a php en los formularios evaluados.	1499098355,34135	1499097111,3581	1499097832,25635	1499098100,0524	1499098355,6411
<b>TOTAL</b>	149909,79509 Microsegundos ( $\mu$ s) <> 1,4990979509 Segundos (s)				

Gráfico 20.Total de Tiempo de Carga en tablas hechas con Yii

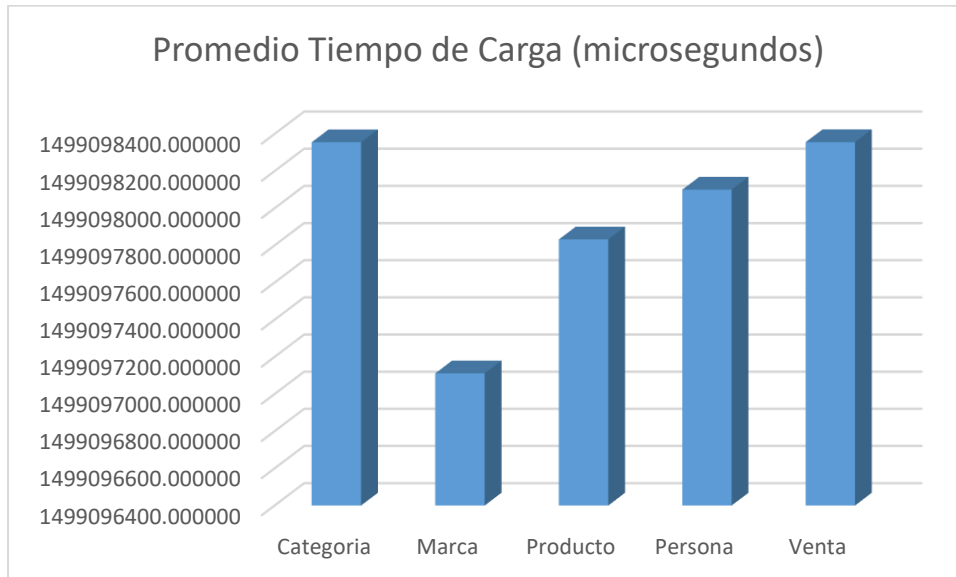
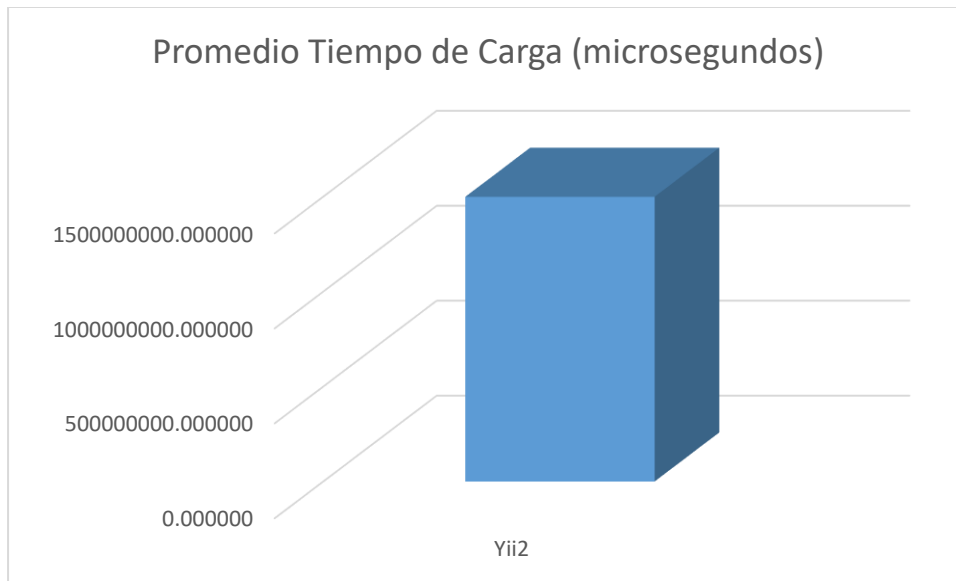


Gráfico 21. Promedio de Tiempo de Carga total consumida por el Framework Yii (microsegundos)



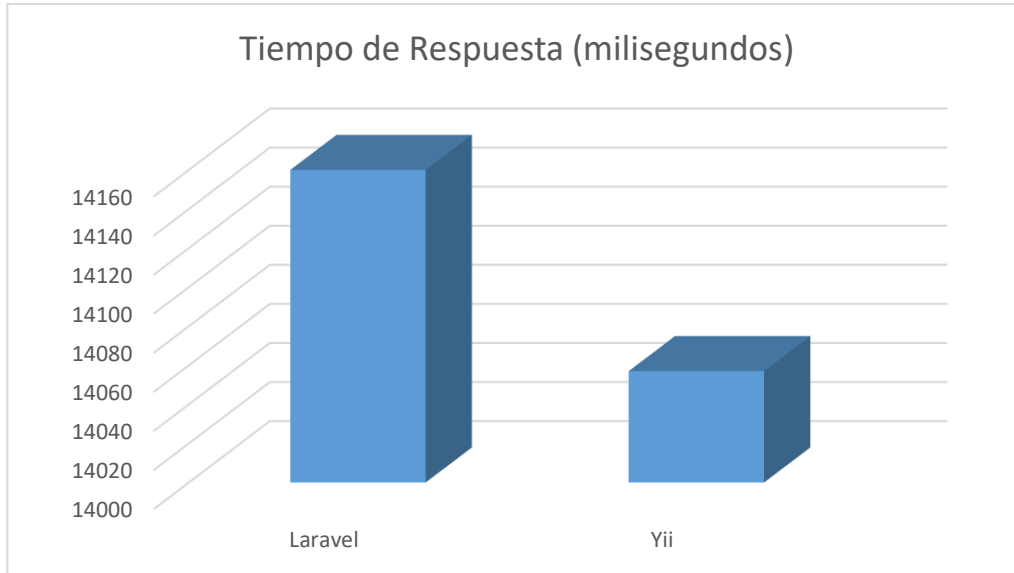
Como resultado en Tiempo de Carga con el framework Yii se obtiene 149909,79509  $\mu$ s.



5.5 Identificar según cada indicador medido, que framework resulto mejor.

5.5.1 Indicador Tiempo de Respuesta

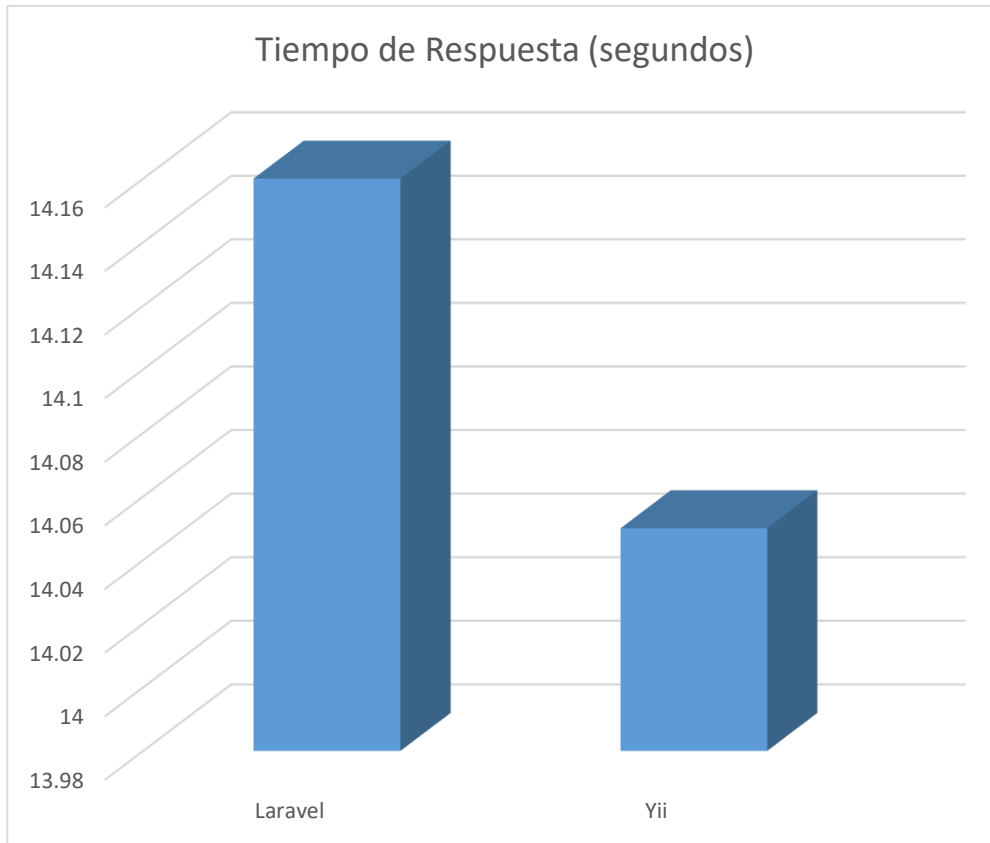
Gráfico 22.Resultado en Tiempo de Respuesta en milisegundos



Como resultado en Tiempo de Respuesta el framework Laravel obtuvo 14160 ms y el framework Yii obtuvo 14057 ms



Gráfico 23.Resultado en Tiempo de Respuesta en segundos



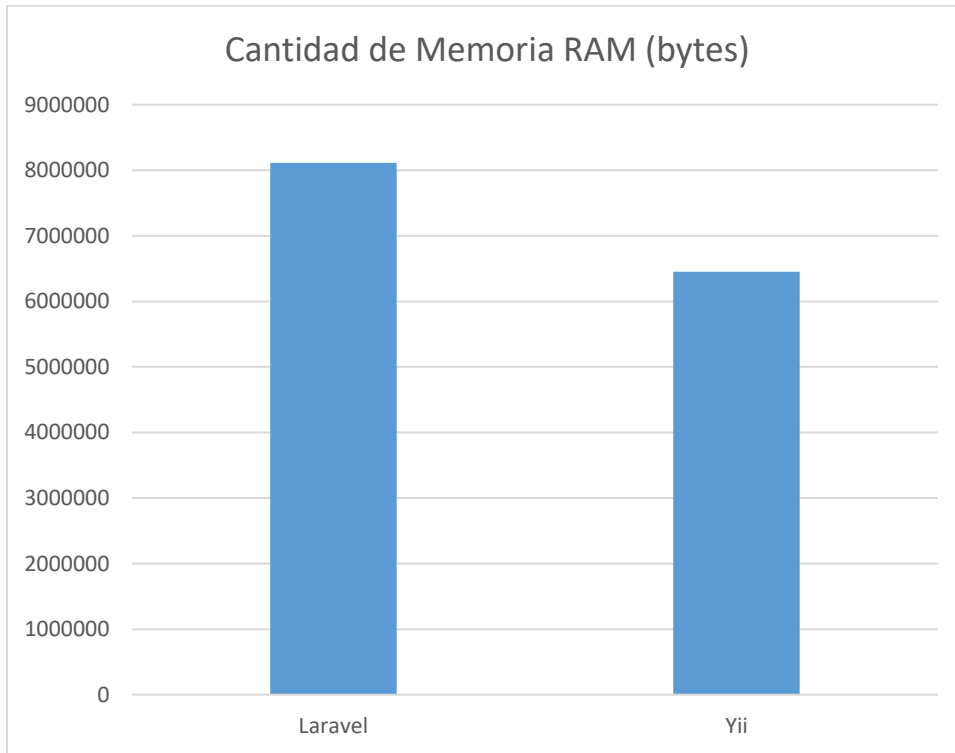
Como resultado en Tiempo de Respuesta con el framework Laravel se obtuvo 14.16 (s), y con el framework Yii se obtuvo 14.05(s).

El mejor Framework en Consumo de Memoria RAM es Yii.



### 5.5.2 Indicador Consumo de Memoria RAM

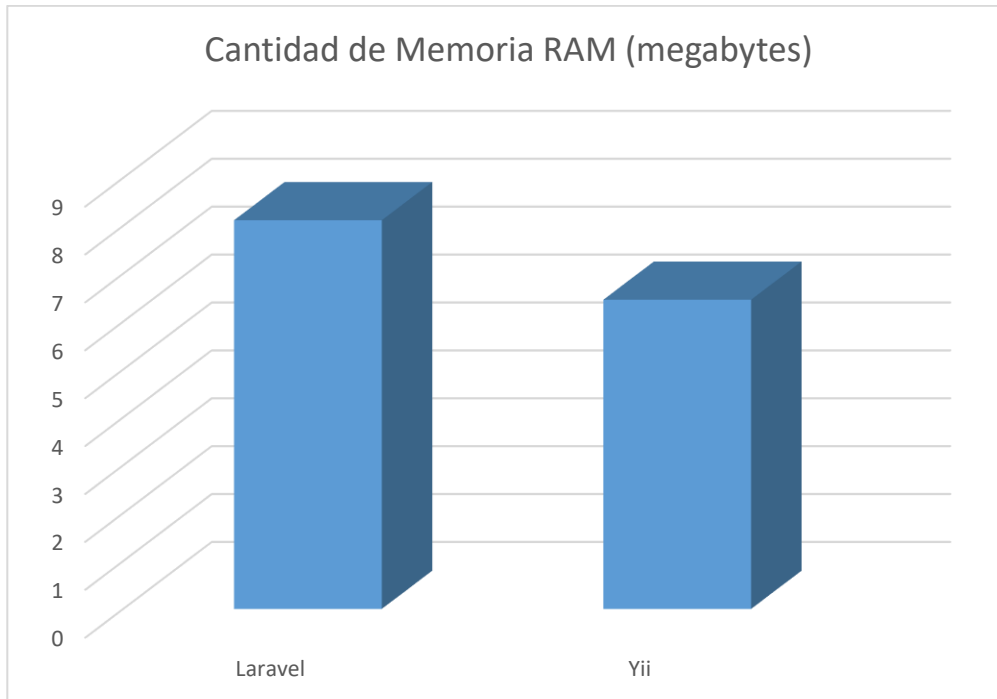
Gráfico 24.Resultado en Consumo de Memoria RAM en bytes



Como resultado en Uso de Memoria RAM el Framework Laravel obtuvo 8114343,2 B y con el framework Yii se obtiene 8114343,2 B.



Gráfico 25.Resultado en Consumo de Memoria RAM en megabytes

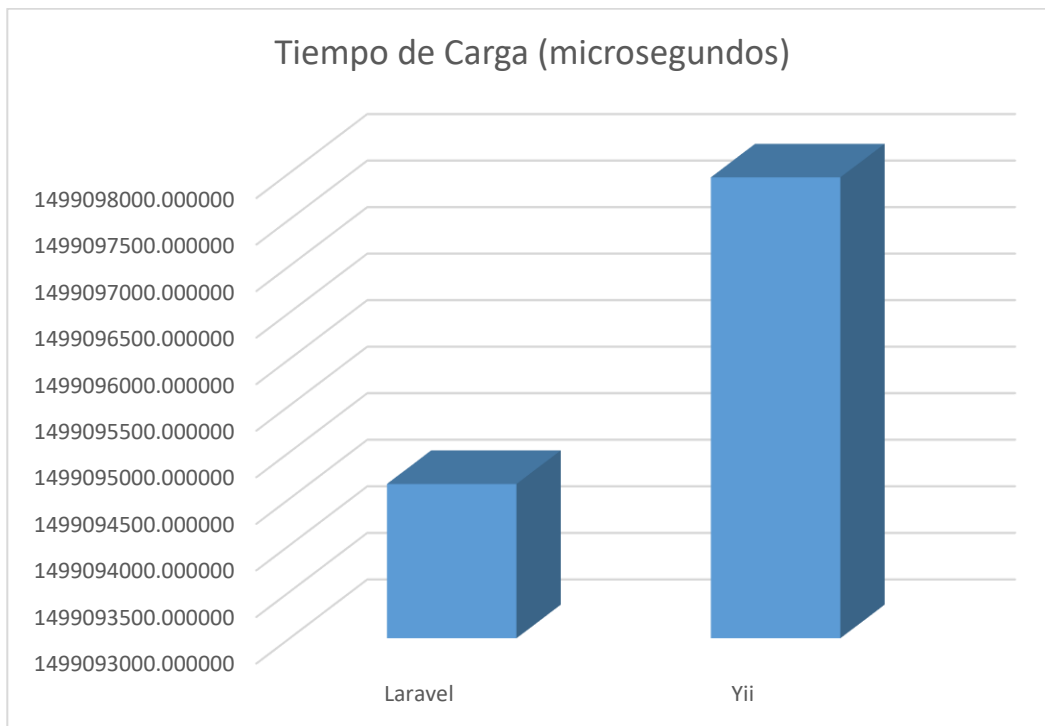


Como resultado en Uso de Memoria RAM el Framework Laravel obtuvo 8,1143432 Mb y con el framework Yii se obtiene 6,4544856 Mb  
El mejor framework en Tiempo de Respuesta es Yii.





### 5.5.3 Indicador Tiempo de Carga



Como resultado en Tiempo de Carga con el framework Laravel 1499094656,44242  $\mu$ s se obtuvo y con el framework Yii se obtuvo 1499097950,92986  $\mu$ s. El mejor Framework en Tiempo de Carga es Laravel



## CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

- a) Se tomó el criterio de popularidad y se realizó un análisis bibliográfico de estudios previos la cual nos permiten fundamentar la elección de los framework Yii y Laravel.
- b) La elección de los indicadores como: Tiempo de Respuesta, Consumo de Memoria RAM y Tiempo de Carga fueron la principal regla de medición ha permitido conocer que framework se desempeña mejor en términos de Rendimiento.
- c) Se implementó un sistema de ventas, el cual permitió de forma práctica comprobar y conocer la amplia información y la forma en que se utiliza cada framework PHP se desempeña , mediante el cual se analizaron tres indicadores de rendimiento.
- d) Se concluye que el framework Yii es mejor en el indicador de Tiempo de Respuesta ya que utiliza un menor tiempo con un resultado de 14.05 segundos, frente a un 14.16 segundos del framework Laravel.
- e) Además en el indicador de Consumo de Memoria RAM el framework Yii hace un menor consumo con un resultado de 6,45MB frente 8,11 MB de Laravel.
- f) Y en el indicador Tiempo de Carga el framework Laravel muestra un mejor resultado con un 149909,46564 microsegundos ( $\mu$ s), frente 149909,79509 microsegundos ( $\mu$ s) del framework Yii.



## RECOMENDACIONES

- a) Se recomienda realizar pruebas de rendimiento en diferentes proyectos de desarrollo web ya que permite evaluar y determinar la eficiencia de las herramientas utilizadas, como lenguaje de programación, framework y otros
- b) Para un estudio a futuro se recomienda realizar un análisis comparativo de otros framework PHP, usando el factor rendimiento.
- c) Se aconseja utilizar el framework Yii para proyectos de desarrollo web que requieran un mejor desempeño en el factor rendimiento.
- d) Se recomienda realizar un plan de pruebas, ya que permitirá un protocolo eficiente en la realización de las pruebas.
- e) Se recomienda para estudios futuros realizar una comparación del patrón MVC versus otros patrones, teniendo en cuenta el factor rendimiento.



## Referencias

- Acosta, S. C. (2015). Analisis Comparativo de los Frameworks para el Desarrollo de Aplicaciones Web en Java.
- Amazon. (24 de Abril de 2017). *Amazon*. Obtenido de Amazon: [www.amazon.com](http://www.amazon.com)
- Arcos Chalán, J. S., & Chicaiza Inguillay, D. E. (2016). Análisis Comparativo de los Frameworks Laravel Y Codelgniter para la Implementación del Sistema de Gestión de Concursos de Méritos Y Oposición en la Universidad Nacional de Chimborazo.
- Bahit, Eugenia. (2013). *POO y MVC PHP*.  
*Built With*. (2017). Obtenido de <https://trends.builtwith.com/>
- Carmen Martínez, C. (s.f.). Sistema de Gestion de Base de Datos Realcionales Difusas Multiproposito.Una Ontologia para la Representacion del Conocimiento Difuso.
- Dzone. (2017). Obtenido de <https://dzone.com/articles/php>
- Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web. (s.f.).
- F. Sierra, J. A. (2013). *Estudio y análisis de los framework en php basados en el modelo vista controlador para el desarrollo de software orientado a la web*.
- Francisco Javier Díaz, C. M. (2015). Metodologías para la evaluación de herramientas Free/Open Source para pruebas de software.
- Grupo CODESI - Training & Certificación. (2016).
- Gutierrez, J. J. (2017). *Vicerrectorado de Investigacion - Universidad de Sevilla*. Obtenido de Area: Departamento de Lenguajes y Sistemas Informaticos: <https://investigacion.us.es/>
- José Samaniego, L. M. (2015). Estudio comparativo de productividad de frameworks PHP orientados a objetos para desarrollar el Sistema de Seguimiento de Incidentes de la Infraestructura de Red en la ESPOCH. 10.
- Kujawa, L. (2017). Obtenido de <https://systemsarchitect.net>
- Laravel. (2 de Mayo de 2017). Obtenido de <https://laravel.com/>
- Laravel. (2017). <https://styde.net/instalacion-de-composer-y-laravel-en-ubuntu-14-lts-paso-a-paso/>.
- Liliana, Z. H., & Ochoa Iglesias, C. L. (2014). Estudio de Frameworks para PHP e Integración a una Herramienta IDE: Aplicado al Portal Web de la Comunidad Linux de la ESPOCH.
- Lopez Saavedra, E. (2009). Catalyst: Framework para el desarrollo de aplicaciones Web.
- Manuel, P. D., & Ildefonso, M. P. (2011). *Programación en PHP a través de ejemplos*. Creative Commons Reconocimiento-CompartirIgual 2.5 España.



- María del Pilar Salas Zárate, G. A. (2015). *Analyzing best practices on Web development frameworks: The lift approach.*
- Medhi Achour, & Jakrub Vrana. (2016). Obtenido de <http://php.net/manual/es/index.php>
- Miranda Chinlli, M., & Rochina Rochina, C. G. (2015). *Análisis de los Frameworks Yii y Zend para mejorar la Usabilidad y la Compatibilidad del Sitio Web de la Cooperativa de Ahorros y Crédito Minga Ltda.*
- Miranda, M., & Rochina, C. (2015). *Análisis de los Frameworks Yii y Zend para mejorar la Usabilidad y la Compatibilidad del Sitio Web de la Cooperativa de Ahorros y Crédito Minga Ltda.*
- MySQL. (2017). Obtenido de <https://www.mysql.com/>
- php.net. (2017). <http://php.net/manual/es/intro-what-is.php>.
- Php.net. (2017). Obtenido de <http://php.net/download-logos.php>
- Pilar, S. M., Giner, H., Rafael, V., Alejandro, R., & Luis, L. (2015). *Analyzing best practices on Web development frameworks.*
- Prokofyeva, N., & Boltunova, V. (2017). *Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Prokofyeva, Natalya.*
- Prokofyevaa, N., & Boltunovaa, V. (2017). *Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems. Prokofyeva, Natalya.*
- Robles, J. R. (2011). *Comparativa entre el Desarrollo Web usando el Framework Jbossseam y el Desarrollo Tradicional.*
- Salas Zárate, M., Alor Hernández, G., Valencia García, R., Rodríguez Mazahua, L., Rodríguez González, A., & López Cuadrado, J. L. (2015). *Analyzing best practices on Web development frameworks: The lift approach.*
- Samaniego Larrea, M. J. (2015). *Estudio comparativo de productividad de frameworks PHP orientados a objetos para desarrollar el Sistema de Seguimiento de Incidentes de la Infraestructura de Red en la ESPOCH.*
- Sánchez Osejo, E. R., & Vera Cárdenas, I. E. (2011). *Análisis del Rendimiento de Frameworks PHP para desarrollar Aplicaciones Web óptimas, Caso Práctico: Portal Academia Linux ESPOCH.*
- T. Chao, J., R. Parker, K., & Davey, B. (2013). *Navigating the Framework Jungle for Teaching Web Application Development.*
- T. Chao, J., R. Parker, K., & Davey, B. (2013). *Navigating the Framework Jungle for Teaching Web Application Development.*
- Victor Robles. (2013). Obtenido de <http://victorroblesweb.es/2013/11/18/tutorial-mvc-en-php-nativo/>
- w3techs. (2017). Obtenido de <https://w3techs.com/>
- Yii. (5 de Mayo de 2017). Obtenido de <http://www.yiiframework.com/>



Zabala Hidalgo Ximena Liliana, O. I. (2008). *Estudio de Frameworks para PHP e Integración a una Herramienta IDE: Aplicado al Portal Web de la Comunidad Linux de la ESPOCH.*

Zend. (2017). *Zend* . Obtenido de Zend: <http://www.zend.com>



**Anexos**  
**Anexo 1**



Solicitud

Estimado (a) señor (a): Gino Paul Gonzales Custodio

Motiva la presente el solicitar su valiosa colaboración en la revisión del instrumento anexo 1, el cual tiene como objeto obtener la validación de los indicadores, que se aplicará para la fundamentación y desarrollo del tema, denominado “ANÁLISIS COMPARATIVO DE FRAMEWORKS PHP PARA MEDIR EL RENDIMIENTO.”

Acudo a usted, debido a sus conocimientos y experiencias en la materia, los cuales aportarían una útil y completa información para la culminación exitosa de este trabajo de investigación.

Gracias por su valioso aporte y participación.



**GUÍA DE JUICIO DE EXPERTOS**

**1. Identificación del Experto**

**Nombre y Apellidos:** Gonzales Custodio Gino Paul

**Centro laboral:** Centro de Sistemas de Información –Gerencia Regional de Salud

**Título profesional:** Ingeniero de Sistemas

**Grado** Maestría Concluida

**Mención** Administración y Dirección en Tecnologías de la Información

**Institución donde lo obtuvo:** Universidad Nacional de Trujillo

**Otros estudios**

- Diploma SCJP SUN ,Certified Java
- Diplomado de Desarrollo de Sistemas de Información con UML RUP
- Diplomado Gerencia y Administración de Proyectos con MS Project

**2. Instrucciones**

Estimado(a) especialista, a continuación se muestra un conjunto de indicadores, el cual tiene que evaluar con criterio ético y estrictez científica, la validez del instrumento propuesto

Para evaluar dicho instrumento, se escribirá el número correspondiente en cada indicador.

Nivel	Puntaje
Bueno	3
Regular	2
Pésimo	1



3. Anexos  
Anexo 1

CUADRO COMPARATIVO DE VALOR BASADO EN CARACTERÍSTICAS

FRAMEWORK	LARAVEL	CODEIGNITER	FUEL.PHP	CAKEPHP	SYMFONY	ZEND	PHALCON	VII	SLIM	PHPME
INDICADORES										
COMPATIBILIDAD DE BASE DE DATOS	3	2	3	3	3	3	3	3	1	3
SEGURIDAD	3	2	3	3	2	3	2	3	1	2
VALIDACIÓN INTEGRADA	3	1	1	1	1	1	1	3	1	1



SISTEMA DE PLANTILLAS RÁPIDO Y FLEXIBLES	3	1	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	
COMPATIBILIDAD PHP	1	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3
DOCUMENTACION	3	1	1	2	2	2	2	2	2	2	2	2	2	3	1	1	1	1	2
ROUTE DE URL FLEXIBLE	3	2	2	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1
SCAFFOLDING	3	1	1	2	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1
MAPEO DE OBJETOS RELACIONAL . ORM	3	1	1	1	2	2	2	2	1	1	1	1	1	3	1	1	1	1	1
FACH. LECTURA DE CODIGO	3	3	3	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1
<b>TOTAL</b>	<b>28</b>	<b>16</b>	<b>18</b>	<b>17</b>	<b>17</b>	<b>16</b>	<b>15</b>	<b>17</b>	<b>15</b>	<b>16</b>	<b>15</b>	<b>17</b>	<b>17</b>	<b>29</b>	<b>11</b>	<b>11</b>	<b>16</b>	<b>16</b>	<b>16</b>

Nota: Índice de validación del juicio de experto puntajes más alto Yii=29 y Laravel=28

4. Niveles de validación

Nivel Pésimo	Nivel Regular	Nivel Bueno
1	2	3
Nivel que demuestra que el framework contiene pocas características y da una probabilidad para no ser usado.	Nivel que demuestra que el framework implementa algunas características, y careciendo de otras relevantes para la programación.	Nivel que indica que el framework posee las mayores y mejores características y está apto para su aplicación.
<b>Interpretación:</b> Cuanto más se acerque el coeficiente uno (1), habrá la probabilidad de no ser tomado en cuenta.		

5. Conclusión general de la validación y sugerencias

Se concluye que de acuerdo a los resultados obtenidos según el CUADRO COMPARATIVO DE VALOR BASADO EN CARACTERISTICAS los frameworks a utilizar en el desarrollo del caso práctico: Sistema de Ventas – SISVENTAS, serán Yii y Laravel al haber obtenido los máximos puntajes demostrando un nivel Excelente.

6. Constancia de Juicio de experto

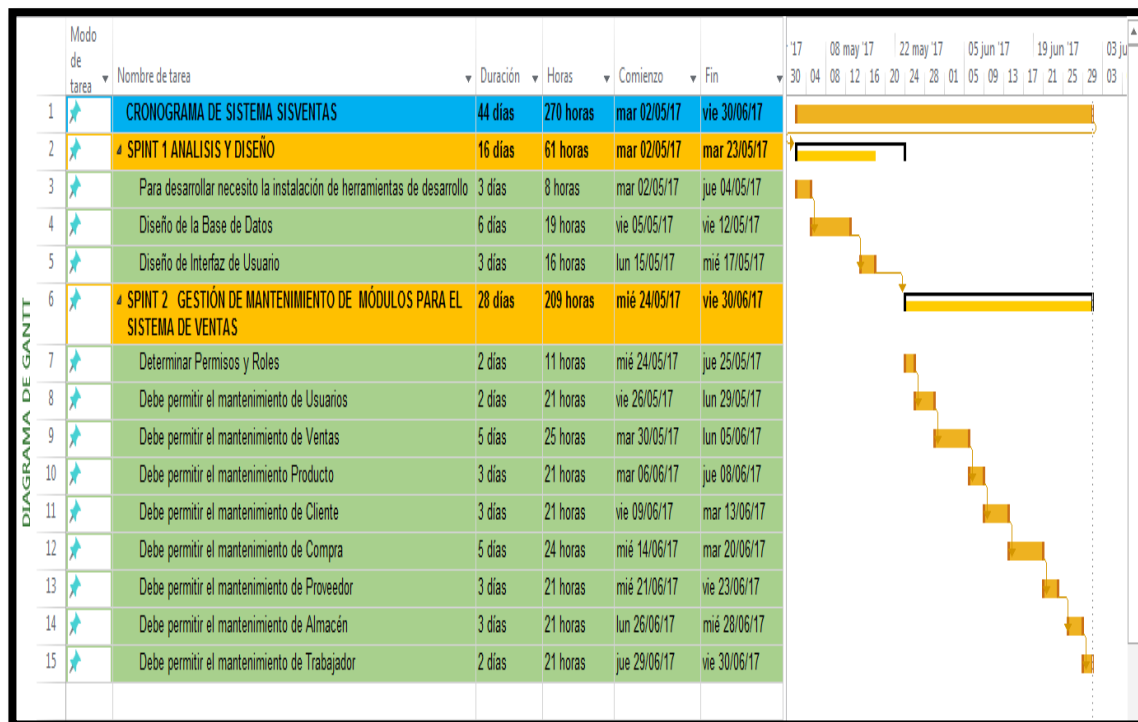
El que suscribe, Gino Paul Gonzales Custodio identificado con DNI. N° 45937483 Certifico que realicé el juicio del experto al instrumento diseñado por los tesisas Llatas Yovera Luis Humberto y Rojas Herrera Alex Jhampier.

  
 GINO PAUL GONZALES CUSTODIO  
 INGENIERO DE SISTEMAS  
 Reg. C.I.P. 149321

Ing. Gonzales Custodio Gino Paul  
 DNI. N° 45937483



Anexo 2 .Cronograma de Desarrollo de Sistemas de Ventas - SISVENTAS



**Anexo 3. Historias de Usuario**

<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HT-01	<b>Usuario:</b> Programador(SCRUM Team )
<b>Nombre historia:</b> Instalación de herramientas de desarrollo	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Se debe realizar la instalación de herramientas de desarrollo para realizar el Sistema en este caso se instalara lo siguiente:                             <ul style="list-style-type: none"> <li>a) Editor de texto SublimeText 3</li> <li>b) Apache</li> <li>c) Composer</li> <li>d) MySQL</li> <li>e) PHP</li> <li>f) Framework Yii 2</li> <li>g) Framework Laravel</li> </ul> </li> </ul>	
<b>Observaciones:</b> Procurar Instalar versiones actuales	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HT-03	<b>Usuario:</b> Programador(SCRUM Team )
<b>Nombre historia:</b> Diseño de Interfaces	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Se debe tener en diseño las posibles interfaces, las cuales deben estar alineadas de la forma más clara para el posterior uso del sistema de ventas.</li> </ul>	
<b>Observaciones:</b> Utilizar los mismo colores previos al finalizar	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-4	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Determinar Permisos y Roles	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá permitir crear ciertos permisos, según el tipo de usuarios</li> <li>➤ Los tipos serán: Administrador y Trabajador</li> </ul>	
<b>Observaciones:</b> Para la creación de la clave ,esta se deberá almacenar de forma encriptado	





<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-5	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Debe permitir el mantenimiento de Usuarios	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá permitir realizar el mantenimiento a la opción Usuario es decir permitir</li> <li>➤ Crear</li> <li>➤ Modificar</li> <li>➤ Eliminar</li> <li>➤ Actualizar</li> </ul>	
<b>Observaciones:</b> Restringir el acceso a módulos según tipo de usuario	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-6	<b>Usuario:</b> Administrador / Trabajador
<b>Nombre historia:</b> Debe permitir el mantenimiento de Ventas	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá permitir realizar la Venta de Productos, en este caso electrodomésticos</li> <li>➤ Deberá permitir realizar el mantenimiento al módulo Ventas es decir permitir</li> <li>➤ Crear</li> <li>➤ Modificar</li> <li>➤ Eliminar</li> <li>➤ Actualizar</li> </ul>	
<b>Observaciones:</b> Mostrar un módulo claro y de forma entendible al usuario del sistema	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-7	<b>Usuario:</b> Administrador /Trabajador
<b>Nombre historia:</b> Debe permitir el mantenimiento Producto	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá almacenar cada producto</li> <li>➤ Deberá permitir realizar el mantenimiento al módulo de Productos es decir permitir</li> <li>➤ Crear</li> <li>➤ Modificar</li> <li>➤ Eliminar</li> <li>➤ Actualizar</li> </ul>	
<b>Observaciones:</b> Mostrar un módulo claro y de forma entendible al usuario del sistema	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-8	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Debe permitir el mantenimiento de Cliente	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá almacenar los datos de cada cliente</li> <li>➤ Deberá permitir realizar el mantenimiento al módulo de Cliente es decir permitir :</li> <li>➤ Crear</li> <li>➤ Modificar</li> <li>➤ Eliminar</li> <li>➤ Actualizar</li> </ul>	
<b>Observaciones:</b> debe enlazarse con el módulo ventas	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-9	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Debe permitir el mantenimiento de Compra	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá permitir el proceso de Compra de Productos</li> <li>➤ Deberá permitir realizar el mantenimiento al módulo de Compras es decir permitir :</li> <li>➤ Crear</li> <li>➤ Modificar</li> <li>➤ Eliminar</li> <li>➤ Actualizar</li> </ul>	
<b>Observaciones:</b> Presentar interfaz entendible	



<b>SISTEMA DE VENTAS (SISVENTAS)</b>	
<b>Número:</b> HU-10	<b>Usuario:</b> Administrador
<b>Nombre historia:</b> Debe permitir el mantenimiento de Proveedor	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> Alta
<b>Programadores Responsables:</b> Alex Rojas y Luis Llatas	
<p><b>Descripción:</b></p> <ul style="list-style-type: none"> <li>➤ Deberá registrar datos del Proveedor de los Productos</li> <li>➤ Deberá permitir realizar el mantenimiento al módulo de Proveedor es decir permitir :                             <ul style="list-style-type: none"> <li>➤ Crear</li> <li>➤ Modificar</li> <li>➤ Eliminar</li> <li>➤ Actualizar</li> </ul> </li> </ul>	
<b>Observaciones:</b> Presentar interfaz entendible	

