

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



**DISEÑO E IMPLEMENTACIÓN DE UNA
APLICACIÓN WEB, PARA SU POSTERIOR ANÁLISIS
MEDIANTE DOS TIPOS DE SISTEMAS DE GESTIÓN
DE BASE DE DATOS**

TRABAJO FIN DE MÁSTER

Anabel Pilicita Garrido

2016

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

**DISEÑO E IMPLEMENTACIÓN DE UNA
APLICACIÓN WEB, PARA SU POSTERIOR ANÁLISIS
MEDIANTE DOS TIPOS DE SISTEMAS DE GESTIÓN
DE BASE DE DATOS**

Autor
Anabel Pilicita Garrido

Director
Enrique Barras Arias

Departamento de Ingeniería de Sistemas Telemáticos

2016

Resumen

Hoy en día el Internet tiene un gran impacto sobre ámbitos como educación, comunicación, gobierno y en la mayor parte de la humanidad por lo tanto, cualquier tipo de entidad u organización busca darse a conocer a través de este medio. Las páginas web proporcionan información y servicios de tal forma, que los sitios web resulten atractivos para captar más usuarios. Es por ello que en el presente trabajo se diseñó e implementó una aplicación web para una Asociación de Docentes la cual brindar información sobre actividades de interés para docentes como por ejemplo seminarios o conferencias promovidas por la asociación con el fin de incentivar la inscripción a las mismas solicitando previamente que los docentes se encuentren registrados mediante un formulario. También es posible compartir comentarios para conocer opiniones del resto de docentes con respecto alguna actividad en particular. La página web permite visualizar información específica de la asociación y también noticias de interés general procurando tener funcionalidades que se encuentran en una página web normal.

Por otra parte, el almacenamiento y administración de los datos requiere Sistemas de Gestión de Base de Datos acorde a las necesidades de la organización. En este caso particular se diseñó un esquema utilizando el modelo Entidad-Relación para crear las bases de datos tanto en MariaDB como en PostgreSQL, para luego poblar cada tabla de la base de datos con información relacionada con actividades, comentarios, sugerencias, inscripciones y docentes de la asociación.

El rendimiento de un sitio web es un factor a tener en cuenta para su buen funcionamiento. Por lo tanto, se utilizó la herramienta Apache Jmeter para realizar pruebas, específicamente de carga mediante la concurrencia de 10, 50, 100 y 150 usuarios en tres páginas web de la aplicación desarrollada.

En las pruebas se estableció un marco de trabajo sobre una computadora portátil donde todos los elementos durante el desarrollo de pruebas se mantuvieron y el único cambio realizado fue la conexión hacia las bases de datos. En primera instancia las páginas web fueron conectadas a una base de datos en MariaDB, a continuación se cambió la conexión a una base de datos en PostgreSQL. Finalmente para el análisis de resultados, se tomó los tiempos de acceso obtenidos a medida que el número de usuarios aumentó comparándolos en cada escenario propuesto.

Abstract

Nowadays, the Internet has a major impact on areas such as education, communication, government and most of mankind, therefore any entity or organization seeks to be known through this means providing information and services using web pages. So that websites are attractive to attract more users every day. That is why, in this paper was designed and implemented a web application for Teachers' Association which provide information on activities of interest to teachers, such as seminars or conferences promoted by the association in order to encourage enrollment to them requesting previously that teachers are registered through a form.

You can also share comments to know opinions of other teachers about any specific activity. The website allows you to view specific information of the association and general interest news also trying to have features found on any web page.

Moreover, storage and data management requires Management Systems Database according to the needs of the organization. In this particular case a scheme was designed using the Entity-Relationship model for creating databases both MariaDB and PostgreSQL, then populate each table in the database with information related to activities, comments, suggestions, inscriptions and teachers of the association. The performance of a website is a factor to consider for proper operation. Therefore the Apache JMeter tool for load testing specifically through attendance of 10, 50, 100 and 150 users in three web pages of the application developed was used.

Our framework was a laptop where all the elements during development and testing remained the only change made was the connection to the database was established. In the first instance web pages were connected to a database in MariaDB, then the connection is switched to a PostgreSQL database. Finally for the analysis of results, access times obtained was taken as the number of users increased compared in each proposed scenario.

Índice general

Resumen	v
Abstract.....	vi
Índice general.....	vii
Índice de figuras.....	xi
Índice de tablas.....	xiii
Siglas	xiv
CAPÍTULO 1.....	1
1 Introducción.....	1
1.1 Objetivos	2
1.1.1 Objetivos Generales.....	2
1.1.2 Objetivos Específicos.....	2
1.2 Alcance.....	3
1.2.1 Diseño e Implementación de la aplicación Web	3
1.2.2 Conexión con MariaDB y PostgreSQL.	3
1.3 Estructura del Documento	4
CAPÍTULO 2.....	5
2 Marco Teórico	5
2.1 Arquitectura Web.....	5
2.2 Diseño Web	7
2.2.1 Twiter Bootstrap	7
2.2.2 Comparativa.....	9
2.3 Programación Web.....	9
2.3.1 PHP.....	10
2.3.2 XAMPP	10
2.3.3 PHP MyAdmin.....	11

2.3.4	Comparativa.....	11
2.4	Bases de Datos SQL.....	12
2.4.1	MariaDB.....	13
2.4.2	PostgreSQL.....	13
2.4.3	Comparativa.....	14
2.5	Rendimiento de Aplicaciones Web.....	15
2.5.1	Apache JMeter.....	16
CAPÍTULO 3.....		17
3	Diseño e Implementación de la Aplicación.....	17
3.1	Diseño de las Páginas Web.....	18
3.1.1	Mapa del Sitio Web.....	19
3.2	Diseño de la Base de Datos.....	26
3.2.1	Modelo Entidad Relación.....	26
3.2.2	Creación de la Base de Datos.....	29
3.3	Implementación de la Aplicación.....	29
CAPÍTULO 4.....		31
4	Pruebas.....	31
4.1	Arquitectura.....	31
4.2	Almacenamiento de datos.....	32
4.2.1	Almacenamiento en MariaDB.....	33
4.2.2	Almacenamiento en PostgreSQL.....	37
4.3	Elementos de Pruebas.....	41
4.4	Plan de Pruebas en Jmeter.....	42
4.4.1	Configuración de parámetros en Jmeter.....	43
4.5	Resultados.....	45
4.5.1	Resultados con MariaDB.....	46
4.5.2	Resultados con PostgreSQL.....	49
4.6	Resultados en conjunto MariaDB y PostgreSQL.....	53
4.6.1	MariaDB y PostgreSQL actividades.php.....	53
4.6.2	MariaDB y PostgreSQL actividades5.php.....	55
4.6.3	MariaDB y PostgreSQL contacto.php.....	57

4.6.4	Consolidación de Resultados MariaDB y PostgreSQL.....	59
4.7	Discusión	61
CAPÍTULO 5.....		63
5	Conclusiones y líneas Futuras	63
5.1	Conclusiones	63
5.2	Líneas Futuras.....	64
Bibliografía		65

Índice de figuras

Figura 1. Comunicación Web	5
Figura 2. Arquitectura de tres capas.....	6
Figura 3. Arquitectura de tres capas.....	17
Figura 4. Capa de Presentación para el Usuario.....	18
Figura 5. Mapa web de la página Asociación de Docentes	19
Figura 6. Página Web de Inicio (index.php).....	20
Figura 7. Página Web de la Misión (mision.php)	20
Figura 8. Página Web de la Visión.....	21
Figura 9. Página Web Informativa de Actividades (actividades.php)	21
Figura 10. Página Web de Contacto (contacto.php)	22
Figura 11. Página Web para el registro de usuarios.....	22
Figura 12. Página Web para Inicio de Sesión de usuarios registrados (login.php) ...	23
Figura 13. Página principal de Usuario (pag_usuario.php).....	23
Figura 14. Página web para actualizar información de usuarios (actualizar.php)	24
Figura 15. Página de actividades inscritas por el usuario (búsqueda.php)	24
Figura 16. Página de Información de Actividades (informacion.php)	25
Figura 17. Página de Comentarios de Usuarios Registrados	25
Figura 18. Modelo Entidad Relación de la Asociación de Docentes.....	26
Figura 19. Modelo ER en WorkBench de la Asociación de Docentes	27
Figura 20. Proyecto alojado en GitHub (https://github.com/Anabel-Pilicita/Trabajo).....	30
Figura 21. Arquitectura para Pruebas en un equipo	31
Figura 22. Página web cargar_datos.php.....	33
Figura 23. Panel de control de phpMyAdmin	34
Figura 24. Tabla Docente en MariaDB.....	34
Figura 25. Tabla Comentario en MariaDB.....	35
Figura 26. Tabla Contacto en MariaDB	35
Figura 27. Tabla Departamento en MariaDB.....	36
Figura 28. Tabla Inscripción en MariaDB	36
Figura 29. Tabla Actividades en MariaDB.....	37
Figura 30. Panel de Control de pgAdmin III.....	38
Figura 31. Tabla Departamento en PostgreSQL.....	38
Figura 32. Tabla Docente en PostgreSQL.....	39
Figura 33. Tabla Comentario en PostgreSQL	39

Figura 34. Tabla Actividades en PostgreSQL.....	40
Figura 35. Tabla Contacto en PostgreSQL	40
Figura 36. Tabla Inscripcion en PostgreSQL	41
Figura 37. Variables Definidas por el Usuario en Jmeter	43
Figura 38. Valores por Defecto para Petición Web.....	43
Figura 39. Valores para los Usuarios Conectados	44
Figura 40. Petición HTTP hacia la página actividades.php.....	44
Figura 41. Petición HTTP hacia la página actividades5.php.....	45
Figura 42. Petición http hacia la página contacto.php	45
Figura 43. Reporte de Resumen	46
Figura 44. Gráfico actividades.php en MariaDB.....	47
Figura 45. Gráfico actividades5.php en MariaDB.....	48
Figura 46. Gráfico contacto.php en MariaDB.....	49
Figura 47. Gráfico actividades.php en PostgreSQL.....	50
Figura 48. Gráfico actividades5.php en PostgreSQL.....	51
Figura 49. Gráfico contacto.php en PostgreSQL.....	52
Figura 50. Gráfico 10 usuarios actividades.php PostgreSQL y MariaDB.....	53
Figura 51. Gráfico 50 usuarios actividades.php PostgreSQL y MariaDB.....	54
Figura 52. Gráfico 100 usuarios actividades.php PostgreSQL y MariaDB.....	54
Figura 53. Gráfico 150 usuarios actividades.php PostgreSQL y MariaDB.....	55
Figura 54. Gráfico 10 usuarios actividades5.php PostgreSQL y MariaDB.....	55
Figura 55. Gráfico 50 usuarios actividades5.php PostgreSQL y MariaDB.....	56
Figura 56. Gráfico 100 usuarios actividades5.php PostgreSQL y MariaDB.....	56
Figura 57. Gráfico 150 usuarios actividades5.php PostgreSQL y MariaDB.....	57
Figura 58. Gráfico 10 usuarios contacto.php PostgreSQL y MariaDB.....	57
Figura 59. Gráfico 150 usuarios contacto.php PostgreSQL y MariaDB.....	58
Figura 60. Gráfico 100 usuarios contacto.php PostgreSQL y MariaDB.....	58
Figura 61. Gráfico 150 usuarios contacto.php PostgreSQL y MariaDB.....	59
Figura 62. MariaDB vs PostgreSQL actividades.php	59
Figura 63. MariaDB vs PostgreSQL actividades5.php	60
Figura 64. MariaDB vs PostgreSQL contacto.php	60

Índice de tablas

Tabla 1. Comparativa entre Bootstrap y Foundation.....	9
Tabla 2. Ventajas y Desventajas de Diferentes Lenguajes de Programación Web.....	11
Tabla 3. Ventajas y Desventajas entre Base de Datos SQL	15
Tabla 4. Datos Almacenados en MariaDB y PostgreSQL	33
Tabla 5. Características técnicas del equipo donde se realizaron las pruebas.....	41
Tabla 6. Versiones de las herramientas usadas para pruebas.....	41
Tabla 7. Plan de Pruebas para MariaDB y PostgreSQL en la página web actividades.php	42
Tabla 8. Plan de Pruebas para MariaDB y PostgreSQL en la página web actividades5.php	42
Tabla 9. Plan de Pruebas para MariaDB y PostgreSQL en la página web contacto.php.....	42
Tabla 10. Resultados escenarios 1-4 MariaDB.....	46
Tabla 11. Resultados escenarios 5-8 MariaDB.....	47
Tabla 12. Resultados escenarios 9-12 MariaDB.....	48
Tabla 13. Resultados escenarios 1-4 PostgreSQL.....	50
Tabla 14. Resultados escenarios 5-8 PostgreSQL.....	51
Tabla 15. Resultados escenarios 9-12 PostgreSQL.....	52

Siglas

HTML5	<i>HyperText Markup Language</i>
CSS	<i>Cascading style sheets</i>
PHP	<i>Hypertext Preprocessor</i>
SQL	<i>Structured Query Language</i>
SGBD	<i>Sistema de Gestión de Base de Datos</i>
DBMS	<i>Database Management System</i>

CAPÍTULO 1

1 Introducción

El auge del Internet y de la web ha promovido el desarrollo de páginas web que son de vital importancia para cualquier tipo de organización o entidad. Independientemente del tipo de sitio web el objetivo es captar al mayor número de personas de forma sencilla a través del internet con el fin de incrementar ya sea el valor económico, o aumentar el prestigio de una organización. Adicionalmente, hoy en día es común el uso de aplicaciones y tecnologías web debido a su fácil implementación y uso por parte de cualquier tipo de entidades o particulares; inclusive información sensible es manejada a través de aplicaciones web. Por lo tanto, el saber diseñar e implementar sitios web promueve la creatividad y las capacidades de programación de los desarrolladores web.

Por otra parte, las organizaciones o entidades también manejan grandes volúmenes de información de personas vinculadas a las mismas; información relevante tanto para los usuarios como para las organizaciones. La creación de bases de datos permite tener una estructura bajo la cual los datos son almacenados, incluyendo restricciones y posibles relaciones entre las tablas que la conforman.

Entre las características que tienen las de bases de datos están las siguientes: independencia cuando los datos se organizan y se gestionan independientemente de los programas que se vayan a usar, la gestión de concurrencia de acceso, integridad de datos, persistencia, recuperación frente a fallos, seguridad, rendimiento, lenguajes de definición y manipulación de datos todo esto se logra a través de un Sistema de Gestión de Base de Datos que oculta la complejidad de la gestión de los datos al desarrollador.

A lo ya mencionado, podemos añadir que el crecimiento de las actividades productivas han aumentado y también son mayores las necesidades de computo; así que uno de los principales retos en base de datos es la estructuración y gestión de grandes volúmenes de datos con el fin de tener una capacidad de almacenamiento, acelerar al acceso a los mismos mediante consultas de forma que si existe un crecimiento de usuarios, las bases de datos deben ser capaces de lidiar con esto; es decir deben poder escalar frente a cambios y frente al crecimiento.

En la actualidad nos encontramos frente a de bases de datos SQL que hacen referencia a base de datos relacionales y bases de datos NoSQL que se caracterizan por tener un esquema dinámico.

El presente trabajo contiene el desarrollo de una aplicación web para una Asociación de Docentes. Para lo cual, se utilizó frameworks, teniendo como resultado páginas web con una estética más agradable y visualmente más profesionales. También se diseñó e implementó el esquema de la base de datos para almacenar información relacionada con la Asociación de Docentes en MariaDB y PostgreSQL. Para enlazar la capa de almacenamiento de datos con las peticiones realizadas por el usuario a través de la interfaz web se utilizó PHP un lenguaje de programación web.

La calidad percibida por los usuarios al cargar una página web está relacionada con el tiempo de esperan. Es por ello, que se utilizó la herramienta Jmeter para pruebas de carga utilizando tres páginas web. Primero, estas fueron conectadas a una base de datos en MariaDB, y luego se realizó las mismas pruebas en otra base de datos en PostgreSQL. Al obtener los resultados se comparó los tiempos obtenidos en cada caso para ponerlos a consideración en una discusión.

1.1 Objetivos

Los objetivos a cumplir son explicados a continuación.

1.1.1 Objetivos Generales

- Diseñar e implementar una aplicación web para una Asociación de Docentes.
- Analizar el rendimiento en páginas web al utilizar dos Sistemas de Gestión de Base de Datos.

1.1.2 Objetivos Específicos

- Manejar Bootstrap para el diseño de páginas web.
- Diseñar e Implementar una aplicación web que contenga páginas web estáticas que sean informativas para los usuarios. Así como páginas web dinámicas que permitan a los usuarios registrarse, inscribirse y comentar en actividades ofrecidas por la asociación.
- Diseñar un modelo Entidad-Relación para crear una base de datos en MariaDB y PostgreSQL para almacenar información de la Asociación de Docentes.
- Generar datos para poblar las tablas en MariaDB y PostgreSQL.
- Crear un plan de Pruebas en Apache Jmeter para diferentes escenarios.

- Realizar pruebas de carga y concurrencia en las páginas web de la Asociación de Docentes.
- Analizar los resultados obtenidos al conectar la aplicación a MariaDB y luego a PostgreSQL para realizar una comparación de resultados.

1.2 Alcance

El presente trabajo engloba dos etapas para su entrega final:

1.2.1 Diseño e Implementación de la aplicación Web

Se diseñó e implementó una página web para una asociación de profesores con las siguientes funcionalidades:

- Muestra información con respecto a la misión, visión de la asociación de docentes. Adicionalmente permite visualizar noticias relacionadas con docentes que puede ser de su interés.
- Muestra información con respecto a la localización de la asociación; permitiendo dejar comentarios.
- Permite registrar a nuevos usuarios.
- Los usuarios que ya se encuentran registrados pueden iniciar sesión con el fin de inscribirse a las actividades que promueve la asociación. Entre las actividades que se destacan son seminarios, conferencias o de carácter deportivo.
- Los usuarios registrados pueden ver comentarios referentes a las actividades y también pueden dejar comentarios propios.
- Es posible cambiar información de usuario para que la misma sea actualizada; y los usuarios registrados pueden visualizar las actividades en las cuales están inscritos.

1.2.2 Conexión con MariaDB y PostgreSQL.

- Se diseñó las bases de datos utilizando el modelo Entidad-Relación.
- Se creó una página web que permite generar datos para poblar las tablas de las bases de datos.
- Se utilizó tres páginas web para la conexión con las bases de datos en MariaDB y PostgreSQL.
- Se analizó los resultados obtenidos tras realizar pruebas de carga utilizando la herramienta Jmeter.

1.3 Estructura del Documento

A continuación, se hace una descripción del contenido de cada uno de los capítulos que componen el presente Trabajo de Fin de Master.

En el Capítulo 1 se realiza una introducción al desarrollo del trabajo de fin de master colocándolo en contexto, indicando objetivos y alcance obtenido.

En el capítulo 2 se expone todas las tecnologías utilizadas para el presente trabajo mediante una descripción de las mismas. Concretamente se detalla el tipo de framework y lenguaje de programación web usado la aplicación web. Así como también, los componentes usados para tener un servidor web local que permita la conexión con las bases de datos SQL seleccionadas para este trabajo. Finalmente se describe la herramienta que permitirá el análisis de las bases de datos Jmeter.

En el capítulo 3 contiene el diseño e implementación de la aplicación web para una asociación de docentes. Se aborda de forma detallada el desarrollo de la aplicación web. En primer lugar se realiza una descripción general del sistema. Seguidamente expone el modelo de datos en que se basa dicho sistema. Por último se detalla el despliegue del sistema en la nube.

En el capítulo 4 se realizará las diferentes pruebas de carga hacia tres páginas web primero conectándolas a MariaDB y luego a PostgreSQL. Los resultados obtenidos son puestos a consideración para una discusión sobre los mismos.

En el capítulo 5 contiene las conclusiones obtenidas luego de la finalización del Trabajo de Fin de Master y posibles líneas futuras.

CAPÍTULO 2

2 Marco Teórico

Para crear una aplicación web se toman en consideración diferentes aspectos para su diseño e implementación, por lo tanto en este capítulo se explica la metodología que permitió el desarrollo de la aplicación web para la Asociación de Docentes. También incluye herramientas y lenguajes utilizados. Finalmente se explica las características principales de la herramienta que permite realiza pruebas de carga hacia un sitio web.

2.1 Arquitectura Web

En la arquitectura web el modelo de comunicación entre cliente y servidor se establece mediante una conexión a Internet. El intercambio de información desde el navegador al Servidor se realiza mediante el uso del protocolo HTTP protocolo para el intercambio de información en Web.

El esquema de comunicación sigue el siguiente formato:

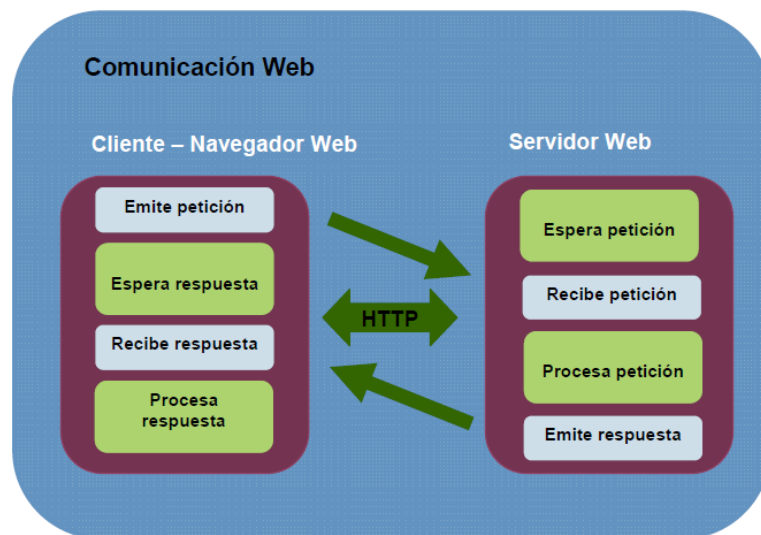


Figura 1. Comunicación Web

El usuario interactúa con las aplicaciones Web, a través del navegador. Como consecuencia de la actividad del usuario, se envían peticiones al servidor, donde se aloja la aplicación web y se usa una base de datos que almacena toda la información relacionada con la misma. El servidor procesa la petición y devuelve la respuesta al navegador que la presenta al usuario.

Por lo tanto, el sistema se distribuye en tres componentes:

- El navegador, que presenta la interfaz al usuario.
- La aplicación, que se encarga de realizar las operaciones de la lógica de la aplicación.
- La base de datos, donde la información relacionada con la aplicación es almacenada.

Esta distribución se conoce como el modelo o arquitectura de tres capas como se visualiza en la Figura 2.

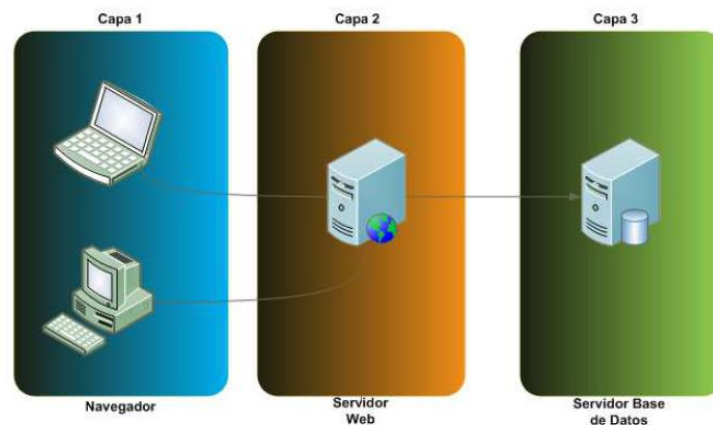


Figura 2. Arquitectura de tres capas

Cada capa tiene su funcionalidad la misma que se explica a continuación.

Nivel de presentación (capa cliente): se encarga de generar la interfaz de usuario. Captura datos de usuario, envía los datos a la capa intermedia y presenta los resultados que procedan de esta.

Nivel de negocio (capa intermedia): capa que constituye el verdadero núcleo de la aplicación Web. Contiene toda la lógica que modela los procesos de negocio y es donde se realiza todo el procesamiento necesario para atender a las peticiones del usuario. Se encarga del procesamiento de los datos del usuario y de la generación y envío de las respuestas a la capa cliente.

Nivel de administración de datos: se encarga de hacer persistente toda la información. Suministra y almacena información requerida por el nivel de negocio.

Luego de tener como referencia esta arquitectura, tenemos claro que se debe estructurar la misma con herramientas que permitan su desarrollo por capas por lo tanto, se define a continuación el diseño web asociado con el nivel de presentación según el modelo ya explicado.

2.2 Diseño Web

El diseño de cualquier sitio web es muy importante. El hecho de proporcionar una página web atractiva al usuario que genere confianza y credibilidad proporciona que aumenten las visitas al sitio web. Aspectos como gama de colores, contrastes, efectos, fondos hoy en día son manejados por el uso de frameworks que facilitado el diseño de las páginas web.

Al definir framework por simple traducción lo asociamos a un marco de trabajo; profundizando un poco más y asociándolo con el desarrollo de software es una estructura conceptual y tecnológica de soporte definido, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto [1].

En el ámbito de programación web los frameworks facilitar el desarrollo de aplicaciones; ya que, contiene herramientas, librerías cuyo objetivo es tener funciones repetitivas en módulos para ser reutilizados. De cierta forma permite a la persona que se encuentra programando olvidarse de tareas que resultan repetitivas y concentrarse en otros elementos del diseño con el fin de mejorar la presentación de la aplicación web. Por lo tanto, es una herramienta que facilita la reutilización de código.

Hoy en día nos encontramos con un sin número de frameworks y la combinación de HTML5 y CSS permite crear diseños web profesionales que pueden ser personalizados rápidamente. Este tipo de frameworks han proporcionado un nivel de diseño más eficiente. Para el desarrollo del presente trabajo se ha utilizado Twiter Bootstrap por lo cual se detalla mejor a continuación.

2.2.1 Twiter Bootstrap

Bootstrap fue desarrollado por Mark Otto y Jacob Thornton de Twitter [2]. En el año del 2011 fue liberado por Twitter como código abierto y en el año de 2012 se convirtió en el proyecto de desarrollo más difundido por GitHub.

Bootstrap es un framework que permite el diseño de sitios y aplicaciones web. Entre sus principales características tenemos:

- La última versión disponible es la 4.0.0-alfa, lanzada en el 2015.
- Contiene plantillas de diseño con tipografía, mensajes de alerta, barras de progreso, formularios, botones, cuadros, menús, listas de navegación, etiquetas horizontales y verticales, ruta de navegación, paginación entre otros elementos de diseño basados en HTML5 y CSS.
- Es compatible con la mayoría de los navegadores web.
- Tiene extensiones que permite trabajar con JavaScript.

- Es de tipo código abierto y se encuentra disponible en GitHub.
- Soporta diseño sensible; por lo tanto, las páginas se ajustan dinámicamente al tipo de equipo usado ya sea un teléfono inteligente, tablet u ordenador.

Bootstrap está conformado por conjunto librerías como normalice, jquery, less, entre otras; por medio de las cuales es posible utilizar efectos y funciones que si las hiciéramos nosotros mismos tardaríamos mucho. Otro aspecto importante es que tiene un sistema de rejilla basado en 12 columnas fluidas que permite acoplar a la página web en función del dispositivo desde el que accedamos. Además las imágenes se irán escalando al tamaño correcto en función si lo estamos viendo desde un móvil o un ordenador de sobremesa, podremos ocultar imágenes, contenedores o lo que queramos de manera sencilla o incluso incluir efectos personalizados en función del dispositivo.

Bootstrap proporciona un conjunto de hojas de estilo que proveen definiciones básicas de estilo para todos los componentes de HTML. Esto otorga una uniformidad al navegador y al sistema de anchura, da una apariencia moderna para el formateo de los elementos de texto, tablas y formularios.

Los componentes de JavaScript para Bootstrap están basados en la librería jquery de JavaScript. Los mismos que proporcionan elementos adicionales de interfaz de usuario como diálogos, carruseles y opciones de autocompletar.

Dentro de sus principales ventajas tenemos:

- Tiene grandes contribuciones por parte de la comunidad de GitHub.
- El diseño es rápido e intuitivo ya que cuenta con componentes ya elaborados.
- Mantiene parámetros de diseño uniforme y organizado.
- Compatibilidad con lenguajes de programación.
- La curva de aprendizaje es liviana.
- Los componentes incluidos en el framework hacen que no se programe código de bajo nivel.
- Los diseños se adaptan automáticamente a las diferentes resoluciones de diferentes dispositivos

Por otra parte, también tiene una serie de desventajas que se ponen a consideración:

- Por defecto ya tiene anchos, márgenes y altos de línea esto con respecto a la rejilla que está basado en 12 columnas.
- Es complicado cambiar de versión cuando se han realizado modificaciones profundas.

2.2.2 Comparativa

En la actualidad existen varias opciones para maquetar web por lo que a continuación se visualiza una tabla comparativa entre dos de las principales herramientas para el diseño web de hoy en día como es Foundation y Twitter Bootstrap; esta última ha sido la que hemos utilizado para el desarrollo del trabajo.

Principales Características	Framework Bootstrap 4.0.0	Framework Foundation
Multi-Navegador	Móvil - Chrome - Android and iOS. - Firefox - Android. - Safari - iOS. -Android Browser & WebView - Android Escritorio -Chrome - Mac/Win -Firefox - Mac/Win - IE9+ - Win (IE9 limitations) - Opera - Mac/Win - Safari - Mac	- Chrome - Firefox - Safari - Ópera - Safari Mobile - IE Mobile - IE 9+ - Navegador de Android 2.3+
Diseño web Adaptable	✓	✓
Código Abierto	✓	✓
Integración con otros Frameworks	✓	Limitada

Tabla 1. Comparativa entre Bootstrap y Foundation

2.3 Programación Web

Las aplicaciones web tienen mucho éxito, principalmente porque solo requieren un navegador web independientemente del sistema operativo y no se necesita instalar ningún software en los equipos de los miles de usuarios que las utilizan. Por lo tanto, una aplicación web puede ser ejecutada en múltiples plataformas.

Actualmente existen diferentes lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. A medida que crecen las demandas de los usuarios se han dado soluciones a través de lenguajes de programación para la web dinámicos; los mismos que permiten interactuar con los usuarios al mismo tiempo que se hace uso de sistemas de bases de datos. A continuación se detalla más a cerca del lenguaje usado en el desarrollo del presente proyecto.

2.3.1 PHP

Fue creado originalmente por Rasmus Lerdorf en 1995; sin embargo la implementación principal de PHP es producida ahora por el grupo PHP. [3]. Es un lenguaje para el desarrollo web de contenido dinámico que está enfocado principalmente en la programación de scripts del lado del servidor. El código es interpretado por un servidor web que tiene un módulo que procesa código PHP y genera una página web como resultado.

Es considerado como una de los lenguajes más flexibles, potentes y de alto rendimiento hasta la actualidad. Entre las principales características de este lenguaje tenemos las siguientes:

- Puede ser implementado en la mayoría de servidores web.
- Es multiplataforma por lo que puede ser ejecutado en Linux, Mac y Windows
- Permite la conexión con varios motores de bases de datos como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, SQLite o MongoDB.
- Código PHP puede incorporarse directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.
- Es un software de licencia libre.
- El código fuente escrito en PHP es invisible al navegador web ya que el servidor se encarga de ejecutar el código y enviar el resultado al navegador.
- Está considerado como un lenguaje de fácil aprendizaje.

Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Es importante contar con un servidor que soporte PHP y pueda tratar a todos los ficheros con extensión .php; es por ellos que se ha incluido el uso de XAMPP.

2.3.2 XAMPP

Esta herramienta de desarrollo fue diseñada por Baiker y Anthony Cooperation [4]. Es un sistema de gestión de base de datos MySQL, el servidor web Apache y los intérpretes de lenguajes PHP y Perl. Por lo tanto es uno de los entornos más utilizados para el desarrollo con PHP. Entre sus principales características tenemos:

- Es una herramienta multiplataforma disponible para Windows, Linux, Solaris y Mac OS X.
- Es de software libre entorno que permite el desarrollo con PHP.
- Desde la versión 5.6.15 se cambió la base de datos MySQL a MariaDB.
- Su instalación y uso resulta ser muy fácil sin ningún tipo de inconveniente.
- También incluye otros módulos como OpenSSL y phpMyAdmin.
- Existen dos tipos de versiones la completa como una versión portátil.

2.3.3 PHP MyAdmin

El proyecto phpMyAdmin es un miembro del Software Freedom Conservancy una organización sin fines de lucro que ayuda a promover, mejorar, desarrollar y defender proyectos de software de código abierto gratuito y libre [5].

El propósito de esta herramienta es la de administra una base de datos a través de la web. Es compatible con operaciones usadas en MySQL y MariaDB. Entre sus principales características está:

- Proporciona una interfaz web intuitiva.
- Soporta consultas, borrar bases de datos, vistas de tablas, campos e índices, modificar bases de datos, tablas, campos e índices.
- Administrar cuentas y privilegios de usuarios.
- Soporta importación y exportación de datos en los siguientes formatos:
formatos: CSV, SQL, XML, PDF, ISO / IEC 26300
- Permite la creación de consultas complejas.

2.3.4 Comparativa

A continuación se presenta una comparativa entre distintos lenguajes de programación para la web.

Lenguajes	Ventajas	Desventajas
PHP	<ul style="list-style-type: none">- Muy fácil de aprender.- Se caracteriza por ser un lenguaje muy rápido.- Soporta en cierta medida la orientación a objeto, clases y herencias.- Capacidad de conexión con la mayoría de los manejadores de base de datos- Posee documentación en su página oficial.	<ul style="list-style-type: none">- Se necesita instalar un servidor web.- Todo el trabajo lo realiza el servidor. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
ASP	<ul style="list-style-type: none">- Usa Visual Basic Script, siendo fácil para los usuarios.- Comunicación óptima con SQL Server.	<ul style="list-style-type: none">- Se necesita escribir mucho código para realizar funciones sencillas.- Tecnología propietaria.- Hospedaje de sitios web costosos.
JSP	<ul style="list-style-type: none">- Ejecución rápida de servlets.- Crear páginas del lado del servidor.- Multiplataforma.- Código bien estructurado.- Integridad con los módulos de Java.	<ul style="list-style-type: none">- Complejidad de aprendizaje.

Tabla 2. Ventajas y Desventajas de Diferentes Lenguajes de Programación Web

2.4 Bases de Datos SQL

Las bases de datos son parte importante de cualquier desarrollo ya que es donde se almacena datos; como se mostró en un inicio en el modelo de tres capas; esta sección muestra información específica de bases de datos SQL.

Una base de datos es una colección de datos relacionados. Por datos hacemos referencia a hechos conocidos que pueden registrarse y que tienen un significado explícito. Adicionalmente, una base de datos puede tener cualquier tamaño y complejidad; puede crearse y mantenerse bien mediante un sistema de gestión de base de datos (SGBD).

Un sistema de gestión de base de datos es una colección de programas que permiten a los usuarios crear y mantener una base de datos. La definición de una base de datos consiste en especificar los tipos de datos, la estructura y restricciones para los datos que se van a almacenar en la misma. La manipulación de la base de datos incluye funciones tales como consultar la base de datos para recuperar unos datos específicos, actualizar base de datos entre otras operaciones que se lo realiza mediante el lenguaje SQL.

Como se mencionó anteriormente para realizar operaciones sobre las bases de datos, ya sea, definirla, crear tablas, realizar modificaciones. Es usado el lenguaje SQL que permite encontrar la información exacta de una manera rápida, repetible y confiable. De esta manera es posible consultar fácilmente y extraer datos significativos a partir de grandes bases de datos.

En concreto un sistema de gestión de base de datos es un software que oculta la complejidad de gestión de los datos al desarrollador. Entre sus características principales tenemos:

- Independencia: Los datos se gestionan independientemente de los programas que los vayan a utilizar teniendo así una independencia lógica y de los ficheros en los que se almacenen obteniendo una independencia física.
- Gestión de concurrencia de acceso.
- Integridad de datos: Mediante transacciones ACID y BASE.
- Persistencia.
- Recuperación frente a fallos
- Seguridad
- Rendimiento
- Lenguajes de Definición y Manipulación de Datos.

Básicamente un gestor de base de datos relacional almacena los datos en tablas, cada una de las cuales está formada por filas que hacen referencia a los registros; al mismo tiempo se tiene columnas relacionadas con los campos.

Se conoce como SQL al lenguaje que se utiliza para consultar o modificar la información almacenada en las bases de datos relacionales. Una base de datos relacional es la que utiliza el modelo relacional para definir las relaciones, condiciones y restricciones de las tablas donde que almacenan la información.

Las bases de datos SQL son bases de datos relacionales que almacenan información de naturaleza heterogénea y que guarda relaciones que pueden ser representadas formalmente según el modelo relacional. Este modelo pretende representar los objetos de la vida real mediante tablas, relaciones y restricciones entre éstas. Las bases de datos relacionales han evolucionado mucho desde sus comienzos, de forma que hoy en día se puede hablar de las bases de datos Objeto Relacional, las cuales no solo almacenan datos simples, sino que pueden almacenar objetos, documentos estructurados y realizar consultas sobre ellos.

2.4.1 MariaDB

MariaDB es uno de los SGBD más populares del mundo. Desarrollado por Michael Widenius fundador de MySQL y la comunidad de desarrolladores de software libre [7]. Se trata de un remplazo para MySQL proporcionando mejoras una de las más destacables es que proporciona mayor escalabilidad y es más robusta.

Entre sus principales características tenemos:

- Ofrece una interfaz SQL para acceder a los datos.
- Ha sido desarrollado como software de código abierto.
- Nuevos motores de almacenamiento como Aria y XtraDB.
- Tiene alta compatibilidad con MySQL ya que posee las mismas bibliotecas y APIs.
- Es utilizado por Wikipedia, Facebook y Google.

2.4.2 PostgreSQL

Es considerada como una de las bases de datos más potentes; proporcionando estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y frente a una alta concurrencia de usuarios.

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

Entre sus principales características encontramos:

- Es una base de datos ACID.

- Es de código abierto.
- Es posible realizar copias de seguridad en caliente.
- Permite múltiples métodos de autenticación.
- Permite el acceso encriptado vía SSL.
- Una gran comunidad que impulsa PostgreSQL y también tiene documentación que facilita su estudio.
- Disponible para Linux, UNIX y Windows.

2.4.3 Comparativa

Existen muchos tipos de bases de datos en función del modo en que almacenan y acceden a la información que guardan: relacional o jerárquica, en red, orientada a objetos, entre otras. Ejemplos de gestores de bases de datos relacionales hay muchos como: MySQL, SQLite, Oracle, Informix, SyBase, PostgreSQL, MariaDB. A continuación se muestra una pequeña comparación entre alguno de ellos.

DBMS	Ventajas	Desventajas
MariaDB	<ul style="list-style-type: none"> - Fácil de aprender y utilizar multiplataforma. - Código abierto - Fácil configurar. - Tiempos de respuesta bajos. 	<ul style="list-style-type: none"> - El soporte es muy básico. - No soporta algunas conversiones de datos. - Los privilegios de las tablas no se borran de forma automática
Oracle	<ul style="list-style-type: none"> - Costoso - Oracle ofrece soporte técnico - Permite la gestión de múltiples bases de datos 	<ul style="list-style-type: none"> - Una mala configuración ofrece resultados desfavorables
PostgreSQL	<ul style="list-style-type: none"> - Menos costoso - Estabilidad - Gran capacidad de almacenamiento 	<ul style="list-style-type: none"> - Lento en actualizaciones.
SQLite	<ul style="list-style-type: none"> - Multiplataforma. - Muchos lenguajes de programación tienen soporte o módulos para SQLite. 	<ul style="list-style-type: none"> - Su límite es de 2 terabytes su base de datos - En algunas versiones los tipos de datos los asigna a los valores individuales y no a columnas esto en ocasiones no permite ser portable a otras bases de datos.
InterBase	<ul style="list-style-type: none"> - Para Microsoft Windows y Linux - Permite hacer copias de seguridad en caliente 	<ul style="list-style-type: none"> - No permite realizar particiones

	- Tiene cercanía al estándar SQL	
Microsoft SQL Server	<ul style="list-style-type: none"> - Soporte de transacciones. - Estabilidad. - Seguridad. - Soporte de procedimientos almacenados. - Entorno grafico 	<ul style="list-style-type: none"> - Utiliza muchos recursos computaciones como memoria RAM. - Es de paga

Tabla 3. Ventajas y Desventajas entre Base de Datos SQL

2.5 Rendimiento de Aplicaciones Web

El rendimiento de un sitio o aplicación web debería ser siempre un factor a tener muy en cuenta para su buen funcionamiento. La estabilidad de un sitio web es clave para una buena experiencia de usuario y depende de una rápida carga de las páginas servidas, una correcta visualización del contenido y una buena respuesta ante múltiples accesos concurrentes[8].

Por lo tanto las pruebas de rendimiento sirven para varios propósitos por lo que a continuación se presentan dos tipos de pruebas [9].

- *Pruebas de carga:* Este tipo de pruebas es usada generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones determinada por un número de usuarios que acceden concurrentemente y las transacciones que realizan durante el tiempo que dura la prueba de carga. Adicionalmente este tipo de pruebas permiten mostrar cuellos de botella que existe en la aplicación.
- *Prueba de estrés:* Esta prueba se utiliza normalmente para romper la aplicación; ya sea doblando el número de usuarios para que sean agregados a la aplicación y se ejecute una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

En las pruebas de rendimiento, a menudo es crucial aunque difícil de conseguir que las condiciones de prueba sean similares a las esperadas en el uso real. Esto es, sin embargo, casi imposible en la práctica. La razón es que los sistemas en producción tienen un carácter aleatorio de la carga de trabajo y aunque en las pruebas se intente dar lo mejor de sí para imitar el volumen de trabajo que pueda tener el entorno de producción, es imposible reproducir exactamente la variabilidad de ese trabajo, salvo en el sistema más simple.

2.5.1 Apache JMeter

En muchas ocasiones luego de desarrollar e implementar una aplicación web los desarrolladores tienden a olvidarse un poco de los requerimientos funcionales y uno de ellos es el rendimiento en un entorno real donde muchos usuarios utilizan la aplicación simultáneamente se requiere conocer si esta sigue funcionando correctamente y no existe una sobrecarga en la misma.

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web .

Entre sus principales características encontramos:

- Puede ser utilizado para probar el rendimiento en recursos estáticos y dinámicos:
 - Estáticos: SOAP/ REST
 - Dinámicos: PHP, Java, ASP.NET.
- Puede ser usado para pruebas unitarias para conexiones de base de datos con JDBC, FTP, LDAP y servicios web.
- Se puede utilizar para realizar un análisis gráfico de rendimiento o para probar su comportamiento.
- JMeter soporta aserciones para asegurarse que los datos recibidos son correctos, por cookies de hilos, configuración de variables y una variedad de reportes.
- Permite simular, con el nivel de concurrencia que se desee, las interacciones de uno o de varios usuarios virtuales.
- Tiene una interfaz que permite crear planes de prueba y controlar la ejecución de las mismas.
- Permite la ejecución de un pruebas localmente o desde línea de comandos.

CAPÍTULO 3

3 Diseño e Implementación de la Aplicación

Para el desarrollo de la aplicación web en conjunto se tomó como guía la arquitectura en tres capas como se visualiza en la Figura 3. El objetivo es separar en partes el desarrollo de la aplicación. Con el fin de tener claro la funcionalidad de cada subsistema. Por lo tanto, cada parte es explicada a continuación.

- Capa 1: La interfaz web que el usuario visualiza, está relacionada con el diseño de las páginas web. Para esto se utilizó la herramienta Bootstrap, en la siguiente sección se muestra en detalle cómo están compuestas las páginas web.
- Capa 2: La parte que maneja la lógica de la aplicación, permite procesar las peticiones y dependiendo de las solicitudes accede a la base de datos para obtener la información y mostrársela a los usuarios. El lenguaje utilizado fue PHP, código incrustado en páginas web para realizar consultas en las bases de datos.
- Capa 3: El almacenamiento de la información. Se diseñó el esquema de la base de datos usando el modelo Entidad Relación. Luego, se creó las bases de datos en MariaDB y PostgreSQL.

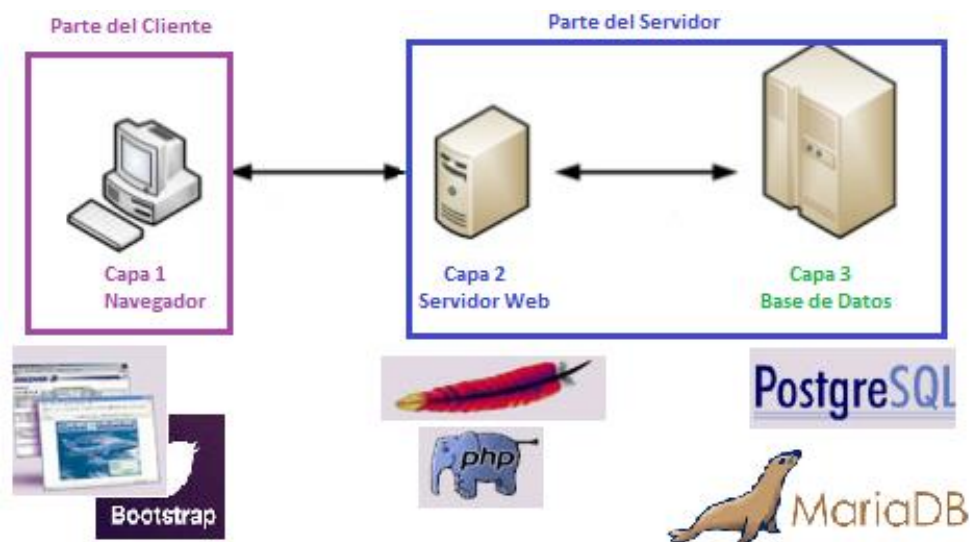


Figura 3. Arquitectura de tres capas

Al final en la implementación es cuando cada subcomponente debe funcionar en conjunto y de forma correcta.

En concreto la aplicación web fue creada para una Asociación de Docentes e implementada en un ordenador portátil personal. El portal web tiene las siguientes funcionalidades:

- Ofrece una página principal que facilita el acceso al resto del portal web.
- Muestra la misión y visión de la asociación, para tener conocimiento a que se dedican.
- Muestra noticias de interés para los docentes.
- Muestra las actividades que son impulsadas por la asociación ya sean seminario o conferencias.
- Ofrece información de contacto, también tiene formulario para que los usuarios establezcan contacto.
- Permite registrar a los usuarios mediante un formulario con esto ellos podrán inscribirse en las actividades que sean de su interés.
- Los usuarios registrados pueden leer y dejar comentarios en las actividades; así como actualizar sus datos personales.

Como ya se mencionó el portal web será usado por 2 tipos de usuarios, usuarios generales y usuarios registrados; los usuarios registrados deben identificarse mediante un usuario y contraseña. Todos los datos relacionados con los docentes, actividades, comentarios y sugerencias son almacenados en las base de datos.

3.1 Diseño de las Páginas Web

La capa de presentación, muestra la interfaz con la que el usuario se comunica con la aplicación, recoge y valida la información enviada por los usuarios y presenta los datos con el formato adecuado.



Figura 4. Capa de Presentación para el Usuario

Para el diseño de las páginas web se utilizó Bootstrap. En la aplicación web contamos con páginas web estáticas que tiene información y también páginas web

dinámicas en donde se incrusta código PHP el mismo que es ejecutado en el servidor web.

3.1.1 Mapa del Sitio Web

El mapa web de la Asociación de Docentes es un esquema conceptual que muestra el acceso a cada página web. Como se puede ver en el siguiente diagrama desde la página Index.php se puede acceder a todas las páginas del portal, excepto las páginas que solo son accesibles después de que un usuario ha sido registrado.

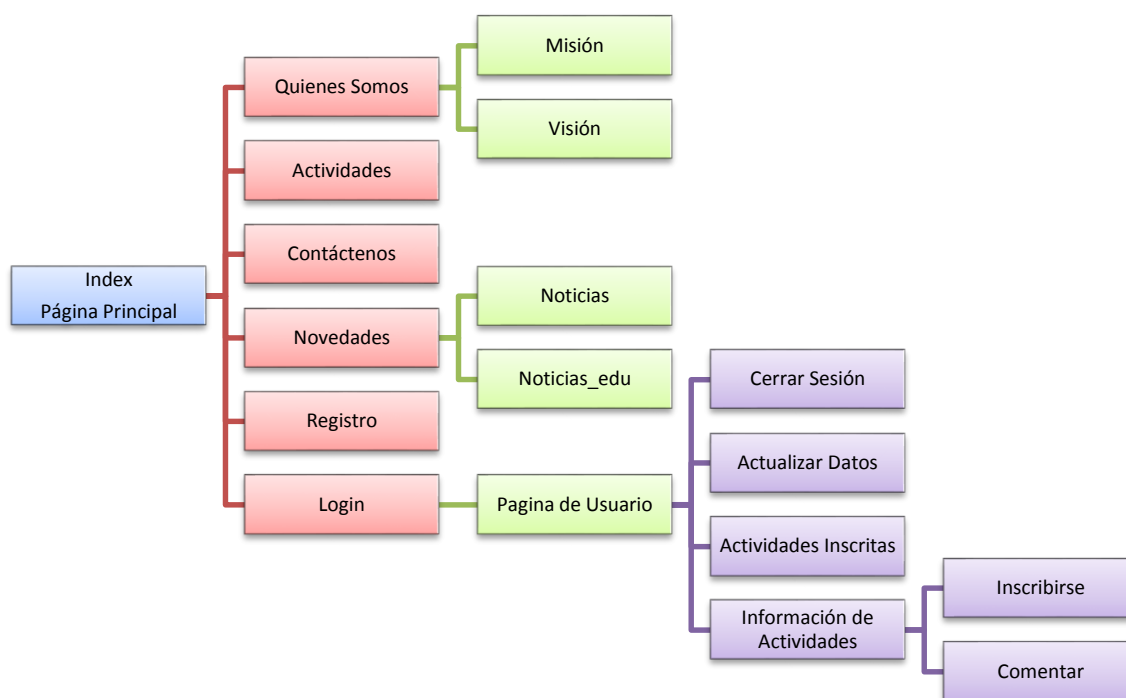


Figura 5. Mapa web de la página Asociación de Docentes

A continuación se presenta el diseño y se explica la funcionalidad de cada una de ellas.

- ***Página Principal (index.php)***

Como vemos en la Figura 6 la página principal da la bienvenida al sitio web. Es un punto de partida que permite enlazarnos con el resto de páginas, mediante la cual los usuarios empiezan a interactuar con el portal web.

ASOCIACIÓN DE DOCENTES

ETSIT UPM ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

Inicio Quienes Somos + Actividades Contáctenos Registro Login



Novedades

Huawei gana más de 5.000 millones de euros en 2015, un 32% más

Huawei obtuvo un beneficio neto de 36.910 millones de yuanes (5.005 millones de euros) al cierre de 2015, lo que representa una mejora del 32,4% con respecto al año anterior, según ha informado la empresa este viernes.

[Leer más »](#)

mSchools reclama el uso responsable de los dispositivos móviles en la educación

En cuatro días de conferencias, ponencias y presentaciones en el Mobile World Congress 2015 sobre la industria móvil no ha faltado la aplicación de la tecnología móvil a la educación. mSchools, un proyecto que tiene como objetivo ayudar a alumnado y docentes a integrar tecnologías digitales en el aula de manera eficaz, ha organizado en el último día del MWC el seminario "Changing Education Together".

[Leer más »](#)

Motivación

"Solo vencióndote vencerás"

- Eloy Alfaro

"Un profesor es el que te enseña, un maestro es del que aprendes"

- Simón Bolívar

Asociación de Docentes
 Escuela Técnica Superior de Ingenieros de Telecomunicación
 Universidad Politécnica de Madrid
 Avenida Complutense nº 30, "Ciudad Universitaria", 28040 - Madrid (España). Tel: +34 91 549 57 00. Fax: +34 91 543 96 52

Figura 6. Página Web de Inicio (index.php)

- **Quienes Somos (mision.php):** Esta página web es meramente informativa sobre aspectos que son manejados por una entidad como es el caso de la misión.

ASOCIACIÓN DE DOCENTES

ETSIT UPM ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

Inicio Quienes Somos + Actividades Contáctenos Registro Login



Misión

La Asociación de Docentes de la ETSIT tiene el propósito de representar los intereses de los docentes, velar por el desarrollo de sus asociados, en los ámbitos: académico, social, cultural, profesional y económico bajo los principios de democracia, unión y respeto; promoviendo una participación activa y permanente de todos sus miembros a fin de establecer nexos de cooperación y colaboración con la Comunidad Politécnica, la Sociedad española y otras organizaciones del exterior, que nos permita promover el bienestar de los profesores

Asociación de Docentes
 Escuela Técnica Superior de Ingenieros de Telecomunicación
 Universidad Politécnica de Madrid
 Avenida Complutense nº 30, "Ciudad Universitaria", 28040 - Madrid (España). Tel: +34 91 549 57 00. Fax: +34 91 543 96 52

Figura 7. Página Web de la Misión (mision.php)

- **Quienes Somos (vision.php):** Es una página web estática que muestra información de la asociación relacionada con la visión de la misma. Este tipo de páginas fueron creadas para mantener aspectos que generalmente los sitios web manejan.



Figura 8. Página Web de la Visión

- **Actividades (actividades.php):** Esta página web muestra todas las actividades que la asociación promueve ya sean seminarios, conferencias, entre otras. El objetivo es promover actividades que despierten el interés por parte de docentes en diferentes ámbitos. Cabe recalcar que esta página en particular es informativa.



Figura 9. Página Web Informativa de Actividades (actividades.php)

- **Contacto (contacto.php):** Esta página web permite conocer la localización de la asociación y ofrece un formulario de contacto para que cualquier usuario que visite el sitio pueda dejar algún tipo de sugerencia o comentario y así establecer una comunicación.



Figura 10. Página Web de Contacto (contacto.php)

- **Registro de Usuarios (registro.php):** Los docentes que quieren formar parte de la asociación se registran mediante un formulario ingresando datos personales. Luego es posible iniciar sesión y pueden acceder a más opciones que se detallan a continuación.



Figura 11. Página Web para el registro de usuarios

- **Inicio de Sesión (login.php):** Esta página web permite a los usuarios registrados iniciar sesión. Presenta un formulario de acceso en el cual se debe ingresar el nombre de usuario y la contraseña correspondiente.



Figura 12. Página Web para Inicio de Sesión de usuarios registrados (login.php)

- **Página de Usuario (pag_usuario.php):** Luego de iniciar sesión los usuarios ingresan a su perfil donde pueden acceder a varias opciones. Inicialmente el acceso está habilitado para obtener más información de actividades. Adicionalmente el resto de enlaces permiten actualizar datos y visualizar las actividades en las cuales está inscrito el usuario registrado.



Figura 13. Página principal de Usuario (pag_usuario.php)

- **Actualización de Datos (actualizar.php):** Es importante mantener la información lo más actualizada posible. Por lo tanto, los usuarios cuando lo requieran pueden ingresar a esta página web para actualizar sus datos personales.

Figura 14. Página web para actualizar información de usuarios (actualizar.php)

- **Actividades Inscritas (busqueda.php):** Luego de inscribirse en alguna actividad en particular. Los usuarios pueden consultar una lista de actividades en las que se encuentran inscritos mediante el acceso a esta página web.

Figura 15. Página de actividades inscritas por el usuario (búsqueda.php)

- **Página de Actividades y Comentarios (informacion.php):** Luego de acceder a una actividad de interés para un usuario, la página web siguiente muestra a detalle la actividad seleccionada con la posibilidad de inscribirse a la misma. También esta página web muestra los comentarios que los diferentes usuarios han dejado en una actividad.



Figura 16. Página de Información de Actividades (informacion.php)

- **Comentarios de Docentes registrados (comentario.php):** Mediante esta página web los docentes registrados pueden agregar comentarios en alguna actividad de esta forma se crea una interacción entre los diferentes miembros de la asociación.



Figura 17. Página de Comentarios de Usuarios Registrados

3.2 Diseño de la Base de Datos

Es necesario tener una base de datos que almacene toda la información relacionada con la Asociación de Docentes. Por lo tanto, el siguiente paso es el diseño y creación de la base de datos. El diseño implica la definición de la estructura que va a tener la base de datos, que se deberá realizar teniendo en cuenta principalmente el tipo de datos que se va almacenar y el modelo de base de datos elegido.

Para el diseño se optó por un modelo conceptual, uno de los más conocidos es el modelo Entidad-Relación. Donde la entidad representa a un objeto o concepto del mundo real sobre el cual se recolecta información. La entidad está compuesta por una serie de atributos o características que la definen. Con respecto a la relación permite establecer una asociación entre las entidades; principalmente define la relación que existe entre las entidades.

3.2.1 Modelo Entidad Relación

El resultado de la etapa de diseño conceptual se expresa a través de un modelo ER, el mismo que se visualiza en la Figura 18. El modelo está compuesto por entidades, atributos y relaciones como ya se mencionó anteriormente.

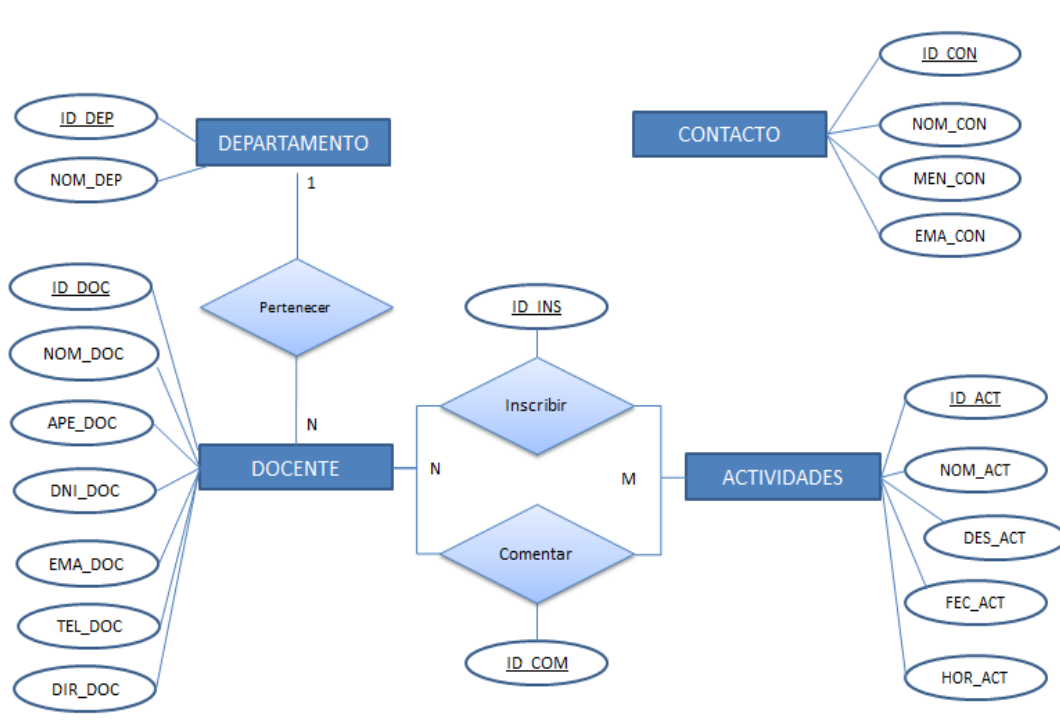


Figura 18. Modelo Entidad Relación de la Asociación de Docentes

Mediante el modelo ER es posible establecer relaciones entre las entidades y se establece la cardinalidad. En el caso de DEPARTAMENTO y DOCENTE existe una relación (1: N), donde a N docente solo puede pertenecer a un departamento. Por otra parte, entre DOCENTE y ACTIVIDADES existe una relación (N:M); en la cual, N

docentes pueden inscribirse y dejar comentarios en muchas actividades. Por el tipo de relación ya mencionado se desprenden las tablas INSCRIPCIÓN y COMENTARIO. Finalmente, se tiene la entidad CONTACTO que no tiene relación con otras entidades del modelo.

Luego de realizar el modelo ER se utilizó la herramienta MySQL WorkBench. Donde se obtuvo como resultado final la estructura de las tablas de la base de datos y las relaciones entre ellas incluyendo claves primarias como claves foráneas. Finalmente nuestro modelo está compuesto por 6 tablas que se describen a continuación.

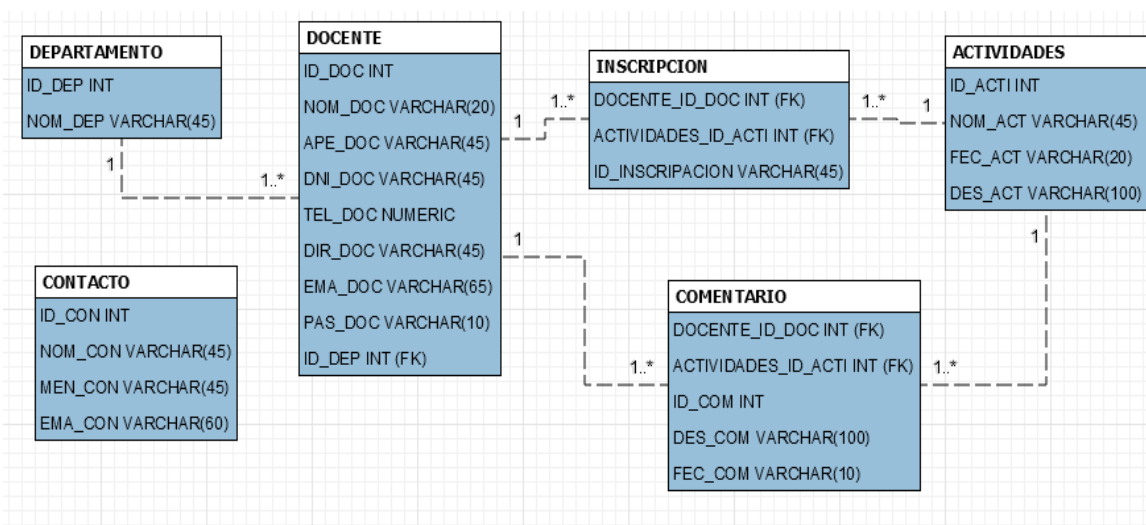


Figura 19. Modelo ER en WorkBench de la Asociación de Docentes

Tabla Departamento: Representa a los departamentos que la escuela a la cual pertenecen los docentes. Los atributos que tiene siguientes:

ID_DEP: Identificador de departamento.

NOM_DEP: Nombre del departamento.

Tabla Docente: Representa el conjunto de docentes registrados en la Asociación de Docentes. Esta tabla viene caracterizada por los siguientes atributos:

ID_DOC: Identificador del docente.

NOM_DOC: Nombre del docente.

APE_DOC: Apellido del docente.

DNI_DOC: Documento de identificación.

TEL_DOC: Número de contacto del docente.

DIR_DOC: Dirección de domicilio.

EMA_DOC: Correo electrónico.

PAS_DOC: Clave de acceso.

Tabla Actividades: Representa todas las actividades que promueve la Asociación de Docentes. Cuenta con los siguientes atributos.

ID_ACT: Identificador de la actividad.

NOM_ACT: Nombre de la actividad.

FEC_ACT: Contiene la fecha y hora en la cual se realizará cualquier tipo de actividad ya sea conferencias, seminarios entre otros.

DES_ACT: Describe brevemente a la actividad.

Tabla Inscripción: Almacena todas las inscripciones a las actividades en la que los docentes se registran para asistir. Viene caracterizado por los siguientes atributos:

ID_INS: Identificador de una inscripción.

ID_DOC: Identificador del docente.

ID_ACT: Identificador de la actividad.

Tabla Comentario: Son todos los comentarios que los distintos usuarios registrados dejan en las actividades. Cuenta con los siguientes atributos.

ID_COM: Identificador del comentario.

DES_COM: Es el comentario dejado por un docente.

FEC_COM: Fecha en la cual se dejó el comentario.

ID_DOC: Identificador del docente:

ID_ACT: Identificador de la actividad.

Tabla Contacto: Contiene todos los mensajes enviados por las personas que acceden a la página web y requieren establecer contacto. Sus atributos son los siguientes.

ID_CON: Identificador del contacto.

NOM_CON: Nombre del usuario que ha dejado el mensaje.

EMA_CON: Email de la persona quien dejó el mensaje.

MEN_CON: Mensaje que ha dejado el usuario.

Cada tabla tiene una clave primaria, es decir, un identificador único de cada registro. Pero también existen claves foráneas o claves externas, en los casos donde para establecer una relación entre dos tablas es necesario incluir, en forma de columna, en una de ellas la clave primaria de la otra.

3.2.2 Creación de la Base de Datos

Una vez elaborado el diseño de la base de datos se debe implementar usando un Gestor de Base de Datos. Para el presente trabajo se seleccionó a MariaDB y PostgreSQL para que crear las base de datos con el diseño del apartado anterior. Se optó por MariaDB ya que en la actualidad es la solución impulsada por Oracle y compatible con MySQL que busca dar continuidad a la misma y también es posible encontrarla dentro de la herramienta Xampp.

Por otra parte, PostgreSQL es una solución de software libre que tiene el respaldo de la comunidad de desarrolladores a nivel mundial que día a día dan seguimiento a nuevos hallazgos para mejorar el servicio.

Por lo tanto, se crearon tres bases de datos con el mismo diseño pero ubicadas en diferentes Sistemas de Gestión de Base de Datos como veremos a continuación.

- *db_asociacion*→ Se encuentra alojada en *MariaDB* y fue utilizada para pruebas de funcionamiento de la aplicación en conjunto.
- *db_datos*→ Está ubicada en *MariaDB* y fue creada para almacenar grandes cantidades de datos para realizar pruebas de rendimiento.
- *db_datosp*→ Está ubicada en *PostgreSQL* y fue creada para almacenar grandes cantidades de datos para realizar pruebas de rendimiento.

3.3 Implementación de la Aplicación

La capa de negocio es en nivel donde se implementa la funcionalidad de la aplicación web. Esta capa sirve de enlace entre los niveles de presentación y de persistencia de los datos. En el desarrollo de la aplicación se trabajó con PHP lenguaje que fue incrustado en las páginas web para colocar las funciones de lógicas que permitieron procesar las peticiones y dependiendo de las solicitudes accede a la base de datos para obtener la información y mostrársela a los usuarios.

Todo el código desarrollado a lo largo del trabajo está ubicado en <https://github.com/Anabel-Pilicita/Trabajo>, de esta manera es como se puede visualizar el desarrollo del trabajo. Se utilizó GitHub un servicio para alojar repositorios de software. Por lo tanto, el código está disponible como open source para un posible uso futuro por parte de cualquier persona.

Anabel-Pilicita / Trabajo Unwatch 1 Star 0 Fork 0
 <> Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings
 No description or website provided. — Edit
 7 commits 1 branch 0 releases 1 contributor
 Branch: master New pull request Create new file Upload files Find file Clone or download
 Anabel-Pilicita committed on GitHub Update README.md Latest commit 194415a 5 days ago

Jmeter	Plan de pruebas	9 days ago
css	Páginas Web	9 days ago
fonts	Páginas Web	9 days ago
img	Páginas Web	9 days ago
js	Páginas Web	9 days ago
README.md	Update README.md	5 days ago
actividades.php	Páginas Web	9 days ago
actividades5.php	Páginas Web	9 days ago
actualizar.php	Páginas Web	9 days ago
busqueda.php	Páginas Web	9 days ago
cambio.php	Páginas Web	9 days ago
cargar_datos_db.php	Páginas Web	9 days ago
comentario.php	Páginas Web	9 days ago

Figura 20. Proyecto alojado en GitHub (<https://github.com/Anabel-Pilicita/Trabajo>)

Como se visualiza en la Figura 20 la gran mayoría de ficheros son páginas web tipo PHP, también incluye carpetas que son utilizadas para el diseño de las páginas e incluye un fichero con el plan de pruebas con Jmeter lo que en el siguiente capítulo se explica pero también está incluido en este repositorio.

CAPÍTULO 4

4 Pruebas

En el presente capítulo se desarrolló un plan de pruebas de carga y concurrencia bajos diferentes escenarios utilizando tres páginas web con la herramienta Apache Jmeter. Las primeras pruebas fueron realizadas cuando las páginas web tenían una conexión contra la base de datos *db_datos* en MariaDB y luego de finalizar estas pruebas se cambió la conexión hacia la base de datos *db_datosp* en PostgreSQL. Los resultados obtenidos fueron puestos a consideración en una discusión para su análisis.

4.1 Arquitectura

El marco de trabajo utilizado para el desarrollo del plan de pruebas fue constituido bajo el concepto de “ceteris paribus”; como se visualiza en la Figura 21, donde todos los elementos de la arquitectura utilizada se mantienen constantes excepto MariaDB y PostgreSQL que presentan las variables de las pruebas. Por lo tanto el almacenamiento va ser el factor que influya en los resultados porque son es el único elemento que cambia ya que el resto de elementos se mantienen iguales.

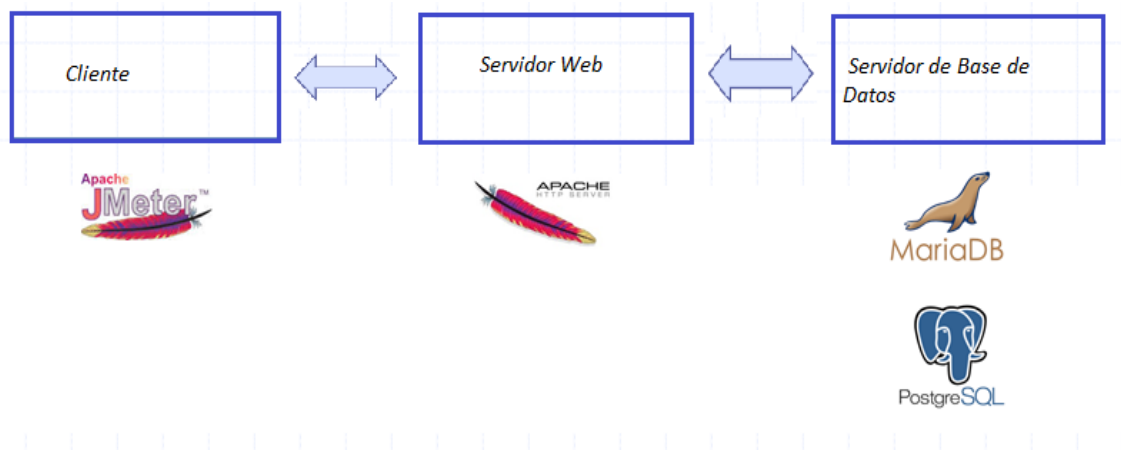


Figura 211. Arquitectura para Pruebas en un equipo

La aplicación web desarrollada cuenta con varias páginas web, algunas de ellas estáticas y otras dinámicas. Por lo tanto, se seleccionó tres páginas web que cuenten con código donde se realicen consultas hacia las bases de datos. Las páginas que fueron utilizadas para las pruebas son descritas a continuación.

- **actividades.php**: El código en esta página web permite realizar una consulta a la base de datos de tipo `"SELECT * FROM ACTIVIDADES"`. Esto, para mostrar toda la información almacenada en la tabla actividades.
- **actividades5.php** tiene el mismo código de la página web anterior; con la única diferencia que no muestra todas las actividades almacenadas en la base de datos sino solo las 5 primeras; esto con el fin de crear una página web con menor cantidad de elementos.
- **contacto.php** esta página web contiene un formulario para ser llenado y ejecutar una consulta tipo `"INSERT INTO CONTACTO (ID_CON, NOM_CON, EMA_CON, MEN_CON) VALUES (NULL, '$nombre', '$email', '$mensaje)'"`. Almacenando información de usuarios que llenan dicho formulario en la tabla CONTACTO.

Las pruebas de carga se las realiza mediante peticiones http hacia las páginas web ya mencionadas y este tipo de peticiones son modificadas dependiendo de método. Por lo que, las peticiones realizadas tuvieron estas variantes según la página web:

- Método GET → **actividades.php** y **actividades5.php**
- Método POST → **contacto.php**

Se estableció que las pruebas se realicen con 10, 50, 100 y 150 usuarios para probar la concurrencia a medida que aumenta la cantidad de peticiones. Por lo tanto las pruebas de carga fueron realizadas mediante peticiones http hacia las páginas web con el objetivo de visualizar el rendimiento a medida que aumenta el número de usuarios. Primero para un acceso simultáneo por parte de los usuarios visualizando información almacenada en las bases de datos y luego, para transacciones durante el tiempo que dura la carga de la página.

4.2 Almacenamiento de datos

Antes de poner en marcha pruebas, se decidió que luego poblar las bases de datos ya creadas. Por lo tanto, mediante código se implementó una página web para generar y almacenar datos y de esta manera llenar las tablas de *db_datos* en MariaDB y *db_datosp* en PostgreSQL.

Cargar Datos

Seccionar tabla ▼

Inicio

Fin

Almacenar

Figura 22. Página web cargar_datos.php

Como se muestra en la Figura 22, la interfaz permite seleccionar cualquiera de las tablas de la base de datos e ingresar un intervalo para generar una cantidad determinada de datos. Por lo tanto, cada tabla de MariaDB como PostgreSQL ya tiene una cantidad determinada de datos como veremos a continuación.

Tablas de las Bases De Datos		Cantidad de Registros
MariaDB	PostgreSQL	
DEPARTAMENTO		20
DOCENTE		1000
ACTIVIDADES		1000
COMENTARIOS		1000
INSCRIPCION		1000
CONTACTO		1000

Tabla 4. Datos Almacenados en MariaDB y PostgreSQL

4.2.1 Almacenamiento en MariaDB

Antes de visualizar las tablas de cada una de las bases de datos, cabe mencionar que MariaDB es compatible con MySQL; por tal motivo, se optó por utilizar la herramienta phpMyAdmin que permite la administración vía web de la base de datos. Como se puede visualizar en la Figura 23 en el lado derecho de la pantalla se encuentran creadas 2 bases de datos. La primera *db_asociacion* con la que se realizó las pruebas de implementación y funcionamiento de la aplicación web y la segunda *db_datos* la que se ha utilizado para almacenar la mayor parte de los datos en cada una de las tabla. Por lo tanto, esta última es la utilizada para realizar las pruebas con Apache Jmeter.

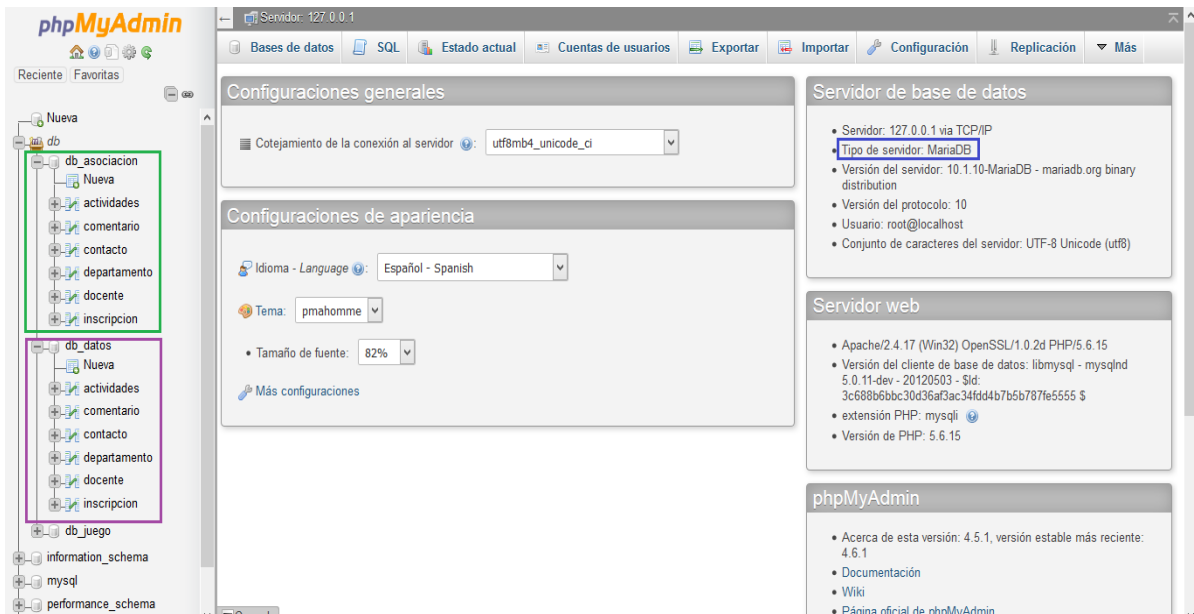


Figura 23. Panel de control de phpMyAdmin

Al ser una herramienta gráfica el realizar una consulta de la información de cada una de las tablas resulta más sencillo. Mediante la Figura 24 se visualiza que el formato utilizado para llenar la tabla DOCENTE; la misma que contiene un total de 1000 registros almacenados inicialmente.

The image shows the phpMyAdmin interface displaying a table view of the 'docente' table. The table has 11 columns: ID_DOC, NOM_DOC, APE_DOC, DNI_DOC, TEL_DOC, DIR_DOC, EMA_DOC, PAS_DOC, and ID_DEP. The table contains 1000 records, numbered 986 to 1000. Each row includes a checkbox for selection and icons for Edit, Copy, and Delete.

ID_DOC	NOM_DOC	APE_DOC	DNI_DOC	TEL_DOC	DIR_DOC	EMA_DOC	PAS_DOC	ID_DEP
986	nombre986	apellidos986	X498986	603669	calle986puerta986piso986	ejemplo986@upm.es	clave%986	6
987	nombre987	apellidos987	X498987	603670	calle987puerta987piso987	ejemplo987@upm.es	clave%987	7
988	nombre988	apellidos988	X498988	603671	calle988puerta988piso988	ejemplo988@upm.es	clave%988	8
989	nombre989	apellidos989	X498989	603672	calle989puerta989piso989	ejemplo989@upm.es	clave%989	9
990	nombre990	apellidos990	X498990	603673	calle990puerta990piso990	ejemplo990@upm.es	clave%990	10
991	nombre991	apellidos991	X498991	603674	calle991puerta991piso991	ejemplo991@upm.es	clave%991	1
992	nombre992	apellidos992	X498992	603675	calle992puerta992piso992	ejemplo992@upm.es	clave%992	2
993	nombre993	apellidos993	X498993	603676	calle993puerta993piso993	ejemplo993@upm.es	clave%993	3
994	nombre994	apellidos994	X498994	603677	calle994puerta994piso994	ejemplo994@upm.es	clave%994	4
995	nombre995	apellidos995	X498995	603678	calle995puerta995piso995	ejemplo995@upm.es	clave%995	5
996	nombre996	apellidos996	X498996	603679	calle996puerta996piso996	ejemplo996@upm.es	clave%996	6
997	nombre997	apellidos997	X498997	603680	calle997puerta997piso997	ejemplo997@upm.es	clave%997	7
998	nombre998	apellidos998	X498998	603681	calle998puerta998piso998	ejemplo998@upm.es	clave%998	8
999	nombre999	apellidos999	X498999	603682	calle999puerta999piso999	ejemplo999@upm.es	clave%999	9
1000	nombre1000	apellidos1000	X4981000	603683	calle1000puerta1000piso1000	ejemplo1000@upm.es	clave%1000	10

Figura 24. Tabla Docente en MariaDB

La tabla COMENTARIOS contiene todos los comentarios generados por los usuarios registrados en alguna de las actividades ofrecidas por la asociación de docentes, se tiene un total de 1000 registros.

	ID_COM	DES_COM	ID_DOC	ID_ACT	FEC_COM
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	985	Comentario985	985	5	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	986	Comentario986	986	6	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	987	Comentario987	987	7	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	988	Comentario988	988	8	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	989	Comentario989	989	9	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	990	Comentario990	990	10	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	991	Comentario991	991	1	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	992	Comentario992	992	2	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	993	Comentario993	993	3	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	994	Comentario994	994	4	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	995	Comentario995	995	5	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	996	Comentario996	996	6	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	997	Comentario997	997	7	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	998	Comentario998	998	8	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	999	Comentario999	999	9	2016-05-15
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1000	Comentario1000	1000	10	2016-05-15

Figura 25. Tabla Comentario en MariaDB.

Para los datos de contacto de las personas que requieren más información acerca de la asociación de docentes pueden dejar comentario o sugerencia se ha ingresado 1000 registros.

	ID_CON	NOM_CON	EMA_CON	MEN_CON
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	985	Nombre completo :985	Mensaje comentario o sugerencia:985	ejemplo985@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	986	Nombre completo :986	Mensaje comentario o sugerencia:986	ejemplo986@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	987	Nombre completo :987	Mensaje comentario o sugerencia:987	ejemplo987@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	988	Nombre completo :988	Mensaje comentario o sugerencia:988	ejemplo988@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	989	Nombre completo :989	Mensaje comentario o sugerencia:989	ejemplo989@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	990	Nombre completo :990	Mensaje comentario o sugerencia:990	ejemplo990@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	991	Nombre completo :991	Mensaje comentario o sugerencia:991	ejemplo991@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	992	Nombre completo :992	Mensaje comentario o sugerencia:992	ejemplo992@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	993	Nombre completo :993	Mensaje comentario o sugerencia:993	ejemplo993@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	994	Nombre completo :994	Mensaje comentario o sugerencia:994	ejemplo994@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	995	Nombre completo :995	Mensaje comentario o sugerencia:995	ejemplo995@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	996	Nombre completo :996	Mensaje comentario o sugerencia:996	ejemplo996@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	997	Nombre completo :997	Mensaje comentario o sugerencia:997	ejemplo997@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	998	Nombre completo :998	Mensaje comentario o sugerencia:998	ejemplo998@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	999	Nombre completo :999	Mensaje comentario o sugerencia:999	ejemplo999@hotmail.com
<input type="checkbox"/> Editar <input type="checkbox"/> Copiar <input type="checkbox"/> Borrar	1000	Nombre completo :1000	Mensaje comentario o sugerencia:1000	ejemplo1000@hotmail.com

Figura 26. Tabla Contacto en MariaDB

En el caso de la tabla DEPARTAMENTO se crearon 20 registros en representación de los departamentos existentes y a los cuales cada uno de los docentes pertenece.

	ID_DEP	NOM_DEP
<input type="checkbox"/>	5	departamento5
<input type="checkbox"/>	6	departamento6
<input type="checkbox"/>	7	departamento7
<input type="checkbox"/>	8	departamento8
<input type="checkbox"/>	9	departamento9
<input type="checkbox"/>	10	departamento10
<input type="checkbox"/>	11	departamento11
<input type="checkbox"/>	12	departamento12
<input type="checkbox"/>	13	departamento13
<input type="checkbox"/>	14	departamento14
<input type="checkbox"/>	15	departamento15
<input type="checkbox"/>	16	departamento16
<input type="checkbox"/>	17	departamento17
<input type="checkbox"/>	18	departamento18
<input type="checkbox"/>	19	departamento19
<input type="checkbox"/>	20	departamento20

Figura 27. Tabla Departamento en MariaDB

En la tabla INSCRIPCIÓN se ingresaron 1000 registros correspondientes a las inscripciones realizadas en distintas actividades por los miembros de la asociación.

	ID_INS	ID_DOC	ID_ACT
<input type="checkbox"/>	985	985	5
<input type="checkbox"/>	986	986	6
<input type="checkbox"/>	987	987	7
<input type="checkbox"/>	988	988	8
<input type="checkbox"/>	989	989	9
<input type="checkbox"/>	990	990	10
<input type="checkbox"/>	991	991	1
<input type="checkbox"/>	992	992	2
<input type="checkbox"/>	993	993	3
<input type="checkbox"/>	994	994	4
<input type="checkbox"/>	995	995	5
<input type="checkbox"/>	996	996	6
<input type="checkbox"/>	997	997	7
<input type="checkbox"/>	998	998	8
<input type="checkbox"/>	999	999	9
<input type="checkbox"/>	1000	1000	10

Figura 28. Tabla Inscripción en MariaDB

Finalmente, para la tabla ACTIVIDADES se generó un total de 1000 registros con el formato que puede ser visualizado en la Figura 29.

	ID_ACT	NOM_ACT	FEC_ACT	DES_ACT
<input type="checkbox"/>	985	Seminario985Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion985:Durante el seminario se explicara e...
<input type="checkbox"/>	986	Seminario986Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion986:Durante el seminario se explicara e...
<input type="checkbox"/>	987	Seminario987Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion987:Durante el seminario se explicara e...
<input type="checkbox"/>	988	Seminario988Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion988:Durante el seminario se explicara e...
<input type="checkbox"/>	989	Seminario989Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion989:Durante el seminario se explicara e...
<input type="checkbox"/>	990	Seminario990Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion990:Durante el seminario se explicara e...
<input type="checkbox"/>	991	Seminario991Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion991:Durante el seminario se explicara e...
<input type="checkbox"/>	992	Seminario992Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion992:Durante el seminario se explicara e...
<input type="checkbox"/>	993	Seminario993Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion993:Durante el seminario se explicara e...
<input type="checkbox"/>	994	Seminario994Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion994:Durante el seminario se explicara e...
<input type="checkbox"/>	995	Seminario995Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion995:Durante el seminario se explicara e...
<input type="checkbox"/>	996	Seminario996Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion996:Durante el seminario se explicara e...
<input type="checkbox"/>	997	Seminario997Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion997:Durante el seminario se explicara e...
<input type="checkbox"/>	998	Seminario998Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion998:Durante el seminario se explicara e...
<input type="checkbox"/>	999	Seminario999Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion999:Durante el seminario se explicara e...
<input type="checkbox"/>	1000	Seminario1000Administracion de Memoria con JAVA	2016-08-03 13:00:00	Descripcion1000:Durante el seminario se explicara ...

Figura 29. Tabla Actividades en MariaDB

4.2.2 Almacenamiento en PostgreSQL

En el caso de PostgreSQL, se utilizó la herramienta pgAdmin III, que facilita el manejo y control de las bases de datos. Como se visualiza en la Figura 30, se creó la base de datos *db_datosp* con las mismas tablas y campos que las base de datos *db_datos* de MariaDB. Por lo tanto, para generar y poblar las tablas Se reutilizó el código usado para MariaDB simplemente se realizó cambios en la conexión hacia la nueva base de datos *db_datosp* en PostgreSQL pero el principio es el mismo.

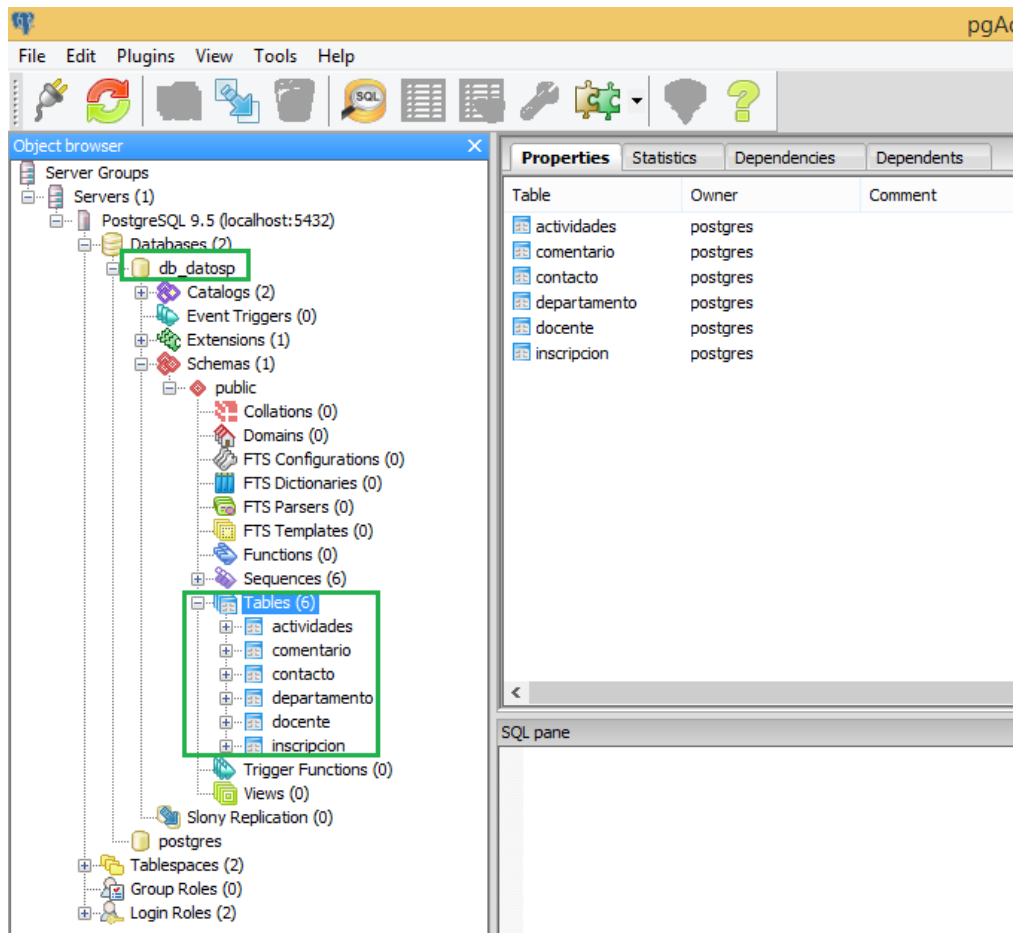


Figura 30. Panel de Control de pgAdmin III

A continuación se muestran las 6 tablas de la base de datos *db_datosp* de PostgreSQL que contienen la misma cantidad de registros que las tablas mostradas con anterioridad de MariaDB. La tabla DEPARTAMENTO cuenta con 20 registros.

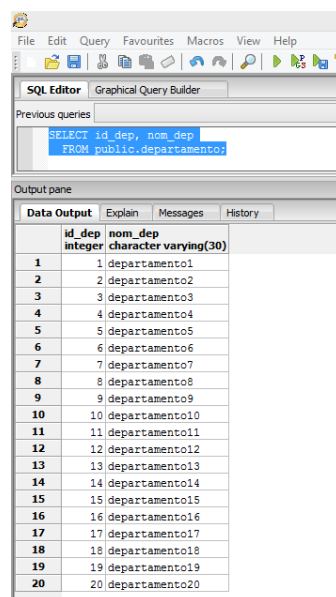


Figura 31. Tabla Departamento en PostgreSQL

En el caso de la tabla Docente tiene en mismo número de registros que MariaDB un total de 1000.

```
SELECT id_doc, nom_doc, ape_doc, dni_doc, tel_doc, dir_doc, ema_doc,
pas_doc, id_dep
FROM public.docente;
```

	id_doc integer	nom_doc character varying(30)	ape_doc character varying(60)	dni_doc character varying(9)	tel_doc integer	dir_doc text	ema_doc character varying(50)	pas_doc character varying(20)	id_dep integer
978	978	nombre978	apellidos978	X498978	603661	calle978puerta978piso978	ejemplo978@upm.es	clave\$978	8
979	979	nombre979	apellidos979	X498979	603662	calle979puerta979piso979	ejemplo979@upm.es	clave\$979	9
980	980	nombre980	apellidos980	X498980	603663	calle980puerta980piso980	ejemplo980@upm.es	clave\$980	10
981	981	nombre981	apellidos981	X498981	603664	calle981puerta981piso981	ejemplo981@upm.es	clave\$981	1
982	982	nombre982	apellidos982	X498982	603665	calle982puerta982piso982	ejemplo982@upm.es	clave\$982	2
983	983	nombre983	apellidos983	X498983	603666	calle983puerta983piso983	ejemplo983@upm.es	clave\$983	3
984	984	nombre984	apellidos984	X498984	603667	calle984puerta984piso984	ejemplo984@upm.es	clave\$984	4
985	985	nombre985	apellidos985	X498985	603668	calle985puerta985piso985	ejemplo985@upm.es	clave\$985	5
986	986	nombre986	apellidos986	X498986	603669	calle986puerta986piso986	ejemplo986@upm.es	clave\$986	6
987	987	nombre987	apellidos987	X498987	603670	calle987puerta987piso987	ejemplo987@upm.es	clave\$987	7
988	988	nombre988	apellidos988	X498988	603671	calle988puerta988piso988	ejemplo988@upm.es	clave\$988	8
989	989	nombre989	apellidos989	X498989	603672	calle989puerta989piso989	ejemplo989@upm.es	clave\$989	9
990	990	nombre990	apellidos990	X498990	603673	calle990puerta990piso990	ejemplo990@upm.es	clave\$990	10
991	991	nombre991	apellidos991	X498991	603674	calle991puerta991piso991	ejemplo991@upm.es	clave\$991	1
992	992	nombre992	apellidos992	X498992	603675	calle992puerta992piso992	ejemplo992@upm.es	clave\$992	2
993	993	nombre993	apellidos993	X498993	603676	calle993puerta993piso993	ejemplo993@upm.es	clave\$993	3
994	994	nombre994	apellidos994	X498994	603677	calle994puerta994piso994	ejemplo994@upm.es	clave\$994	4
995	995	nombre995	apellidos995	X498995	603678	calle995puerta995piso995	ejemplo995@upm.es	clave\$995	5
996	996	nombre996	apellidos996	X498996	603679	calle996puerta996piso996	ejemplo996@upm.es	clave\$996	6
997	997	nombre997	apellidos997	X498997	603680	calle997puerta997piso997	ejemplo997@upm.es	clave\$997	7
998	998	nombre998	apellidos998	X498998	603681	calle998puerta998piso998	ejemplo998@upm.es	clave\$998	8
999	999	nombre999	apellidos999	X498999	603682	calle999puerta999piso999	ejemplo999@upm.es	clave\$999	9
1000	1000	nombre1000	apellidos1000	X4981000	603683	calle1000puerta1000piso1	ejemplo1000@upm.es	clave\$1000	10

Figura 32. Tabla Docente en PostgreSQL

En el caso de la tabla Comentario tiene 1000 registros.

```
SELECT * from comentario;
```

	id_com integer	des_com text	id_doc integer	id_act integer	fec_com date
975	975	Comentario975	975	5	2016-05-15
976	976	Comentario976	976	6	2016-05-15
977	977	Comentario977	977	7	2016-05-15
978	978	Comentario978	978	8	2016-05-15
979	979	Comentario979	979	9	2016-05-15
980	980	Comentario980	980	10	2016-05-15
981	981	Comentario981	981	1	2016-05-15
982	982	Comentario982	982	2	2016-05-15
983	983	Comentario983	983	3	2016-05-15
984	984	Comentario984	984	4	2016-05-15
985	985	Comentario985	985	5	2016-05-15
986	986	Comentario986	986	6	2016-05-15
987	987	Comentario987	987	7	2016-05-15
988	988	Comentario988	988	8	2016-05-15
989	989	Comentario989	989	9	2016-05-15
990	990	Comentario990	990	10	2016-05-15
991	991	Comentario991	991	1	2016-05-15
992	992	Comentario992	992	2	2016-05-15
993	993	Comentario993	993	3	2016-05-15
994	994	Comentario994	994	4	2016-05-15
995	995	Comentario995	995	5	2016-05-15
996	996	Comentario996	996	6	2016-05-15
997	997	Comentario997	997	7	2016-05-15
998	998	Comentario998	998	8	2016-05-15
999	999	Comentario999	999	9	2016-05-15
1000	1000	Comentario1000	1000	10	2016-05-15

Figura 33. Tabla Comentario en PostgreSQL

En el caso de la tabla ACTIVIDADES está almacenados 1000 registros.

id_act integer	nom_act character varying(60)	fec_act date	des_act text
977	Seminario977Administracion de Memoria con JAVA	2016-08-03	Descripcion977:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
978	Seminario978Administracion de Memoria con JAVA	2016-08-03	Descripcion978:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
979	Seminario979Administracion de Memoria con JAVA	2016-08-03	Descripcion979:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
980	Seminario980Administracion de Memoria con JAVA	2016-08-03	Descripcion980:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
981	Seminario981Administracion de Memoria con JAVA	2016-08-03	Descripcion981:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
982	Seminario982Administracion de Memoria con JAVA	2016-08-03	Descripcion982:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
983	Seminario983Administracion de Memoria con JAVA	2016-08-03	Descripcion983:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
984	Seminario984Administracion de Memoria con JAVA	2016-08-03	Descripcion984:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
985	Seminario985Administracion de Memoria con JAVA	2016-08-03	Descripcion985:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
986	Seminario986Administracion de Memoria con JAVA	2016-08-03	Descripcion986:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
987	Seminario987Administracion de Memoria con JAVA	2016-08-03	Descripcion987:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
988	Seminario988Administracion de Memoria con JAVA	2016-08-03	Descripcion988:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
989	Seminario989Administracion de Memoria con JAVA	2016-08-03	Descripcion989:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
990	Seminario990Administracion de Memoria con JAVA	2016-08-03	Descripcion990:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
991	Seminario991Administracion de Memoria con JAVA	2016-08-03	Descripcion991:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
992	Seminario992Administracion de Memoria con JAVA	2016-08-03	Descripcion992:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
993	Seminario993Administracion de Memoria con JAVA	2016-08-03	Descripcion993:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
994	Seminario994Administracion de Memoria con JAVA	2016-08-03	Descripcion994:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
995	Seminario995Administracion de Memoria con JAVA	2016-08-03	Descripcion995:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
996	Seminario996Administracion de Memoria con JAVA	2016-08-03	Descripcion996:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
997	Seminario997Administracion de Memoria con JAVA	2016-08-03	Descripcion997:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
998	Seminario998Administracion de Memoria con JAVA	2016-08-03	Descripcion998:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
999	Seminario999Administracion de Memoria con JAVA	2016-08-03	Descripcion999:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri
1000	Seminario1000Administracion de Memoria con JAV	2016-08-03	Descripcion1000:Durante el seminario se explicara el manejo de memoria en JAVA junto con sus características. Se cubri

Figura 34. Tabla Actividades en PostgreSQL.

En el caso de la tabla CONTACTO tiene 1000 registros.

id_con integer	nom_con character varying(50)	ema_con character varying(50)	men_con text
976	Nombre completo :976	Mensaje comentario o sugerencia:976	ejemplo976@hotmail.com
977	Nombre completo :977	Mensaje comentario o sugerencia:977	ejemplo977@hotmail.com
978	Nombre completo :978	Mensaje comentario o sugerencia:978	ejemplo978@hotmail.com
979	Nombre completo :979	Mensaje comentario o sugerencia:979	ejemplo979@hotmail.com
980	Nombre completo :980	Mensaje comentario o sugerencia:980	ejemplo980@hotmail.com
981	Nombre completo :981	Mensaje comentario o sugerencia:981	ejemplo981@hotmail.com
982	Nombre completo :982	Mensaje comentario o sugerencia:982	ejemplo982@hotmail.com
983	Nombre completo :983	Mensaje comentario o sugerencia:983	ejemplo983@hotmail.com
984	Nombre completo :984	Mensaje comentario o sugerencia:984	ejemplo984@hotmail.com
985	Nombre completo :985	Mensaje comentario o sugerencia:985	ejemplo985@hotmail.com
986	Nombre completo :986	Mensaje comentario o sugerencia:986	ejemplo986@hotmail.com
987	Nombre completo :987	Mensaje comentario o sugerencia:987	ejemplo987@hotmail.com
988	Nombre completo :988	Mensaje comentario o sugerencia:988	ejemplo988@hotmail.com
989	Nombre completo :989	Mensaje comentario o sugerencia:989	ejemplo989@hotmail.com
990	Nombre completo :990	Mensaje comentario o sugerencia:990	ejemplo990@hotmail.com
991	Nombre completo :991	Mensaje comentario o sugerencia:991	ejemplo991@hotmail.com
992	Nombre completo :992	Mensaje comentario o sugerencia:992	ejemplo992@hotmail.com
993	Nombre completo :993	Mensaje comentario o sugerencia:993	ejemplo993@hotmail.com
994	Nombre completo :994	Mensaje comentario o sugerencia:994	ejemplo994@hotmail.com
995	Nombre completo :995	Mensaje comentario o sugerencia:995	ejemplo995@hotmail.com
996	Nombre completo :996	Mensaje comentario o sugerencia:996	ejemplo996@hotmail.com
997	Nombre completo :997	Mensaje comentario o sugerencia:997	ejemplo997@hotmail.com
998	Nombre completo :998	Mensaje comentario o sugerencia:998	ejemplo998@hotmail.com
999	Nombre completo :999	Mensaje comentario o sugerencia:999	ejemplo999@hotmail.com
1000	Nombre completo :1000	Mensaje comentario o sugerencia:1000	ejemplo1000@hotmail.com

Figura 35. Tabla Contacto en PostgreSQL

La tabla INSCRIPCION cuenta con 1000 registros.

Output pane			
Data Output	id_ins	id_doc	id_act
	integer	integer	integer
976	976	976	6
977	977	977	7
978	978	978	8
979	979	979	9
980	980	980	10
981	981	981	1
982	982	982	2
983	983	983	3
984	984	984	4
985	985	985	5
986	986	986	6
987	987	987	7
988	988	988	8
989	989	989	9
990	990	990	10
991	991	991	1
992	992	992	2
993	993	993	3
994	994	994	4
995	995	995	5
996	996	996	6
997	997	997	7
998	998	998	8
999	999	999	9
1000	1000	1000	10

Figura 36. Tabla Incripcion en PostgreSQL

4.3 Elementos de Pruebas

Para realizar las pruebas de rendimiento se utilizó un equipo personal con las siguientes características técnicas.


	Portátil HP Pavilion dm4 con
	Procesador Corei5.
	Memoria RAM de 6GB.
	Sistema Operativo Windows 7 de 64bits

Tabla 5. Características técnicas del equipo donde se realizaron las pruebas

Por otra parte las herramientas usadas para las pruebas tienen las siguientes versiones:






	Servidor Web Apache 2.4.17
	Servidor MariaDB 10.1.10
	Servidor PostgreSQL 9.5
	PHP versión 5.6.15.
	Apache Jmeter versión 2.13

Tabla 6. Versiones de las herramientas usadas para pruebas

Es importante mencionar las condiciones bajo las cuales se realizaron las pruebas ya que el objetivo es comparar los resultados obtenidos bajo las mismas circunstancias en las bases de datos MariaDB y PostgreSQL.

4.4 Plan de Pruebas en Jmeter

Luego de poblar las bases de datos tanto en MariaDB como en PostgreSQL y de establecer todos los elementos utilizados para las pruebas de rendimiento hacia páginas web. Se realizó un plan de pruebas de carga tomando a consideración las páginas web, el tipo de peticiones hacia las mismas y la concurrencia de los usuarios.

Finalmente, se muestran todos los escenarios para realizar las pruebas bajo las condiciones ya establecidas.

MariaDB y PostgreSQL	
Página Web: <i>actividades.php</i>	
Escenarios	Usuarios Conectados
1	10
2	50
3	100
4	150

Tabla 7. Plan de Pruebas para MariaDB y PostgreSQL en la página web actividades.php

MariaDB y PostgreSQL	
Página Web: <i>actividades5.php</i>	
Escenarios	Usuarios Conectados
5	10
6	50
7	100
8	150

Tabla 8. Plan de Pruebas para MariaDB y PostgreSQL en la página web actividades5.php

MariaDB y PostgreSQL	
Página Web: <i>contacto.php</i>	
Escenarios	Usuarios Conectados
9	10
10	50
11	100
12	150

Tabla 9. Plan de Pruebas para MariaDB y PostgreSQL en la página web contacto.php

Como observa se plantea 12 escenarios los mismos que fueron ejecutados primero con las páginas web conectadas a MariaDB y luego los mismos 12 escenarios fueron usados en PostgreSQL.

4.4.1 Configuración de parámetros en Jmeter

A continuación se muestra las configuraciones iniciales que se realizaron en Jmeter. Es necesario crear un plan de pruebas, dentro de las configuraciones iniciales se crearon variables definidas por el usuario donde se coloca el puerto y el nombre del servidor como se visualiza en la Figura 37.

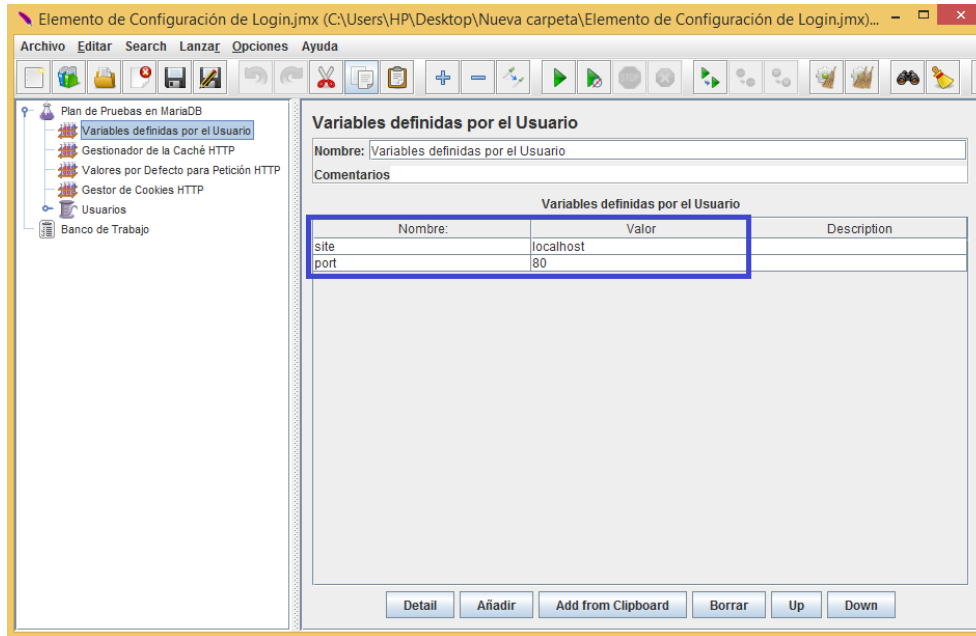


Figura 37. Variables Definidas por el Usuario en Jmeter

Las variables definidas nos permiten establecer una petición http por defecto con las mismas como se ve en la Figura 38.

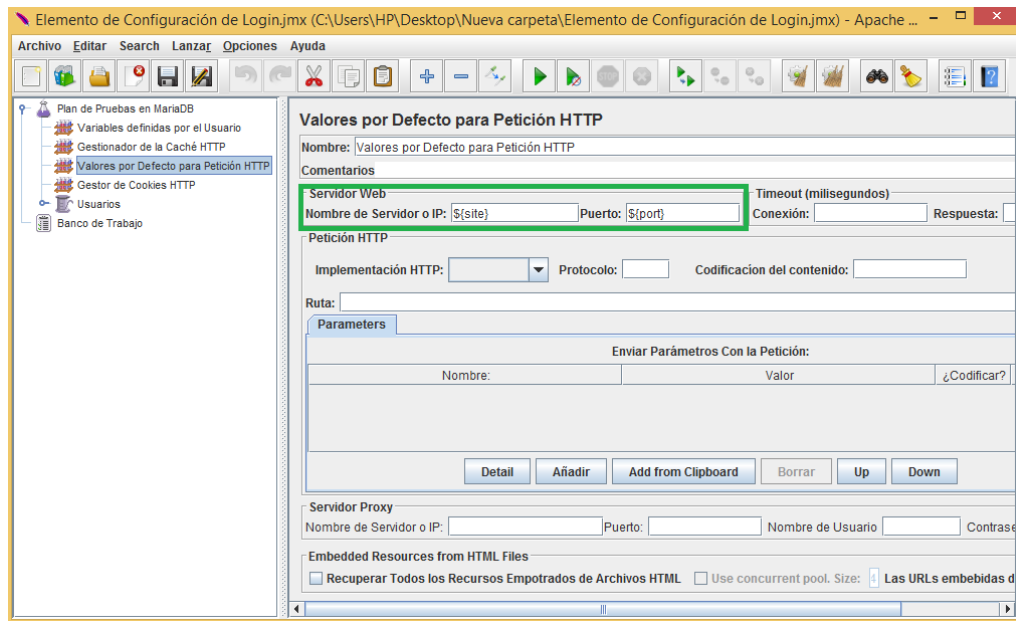


Figura 38. Valores por Defecto para Petición Web

Adicional a todas las consideraciones ya mencionadas, es necesario tomar en cuenta el grupo de hilos que se define como el número de usuarios que acceden. El Período de Subida indica cuantos segundos se tardará en alcanzar el número máximo de usuario; es importante mencionar que el periodo de subida debe ser suficientemente grande para evitar una carga excesiva en el inicio de la prueba, y lo suficientemente corto para que los últimos hilos arranquen antes de que los primeros terminen. Finalmente el contador del bucle representa el número de veces que cada usuario ejecutará la prueba.

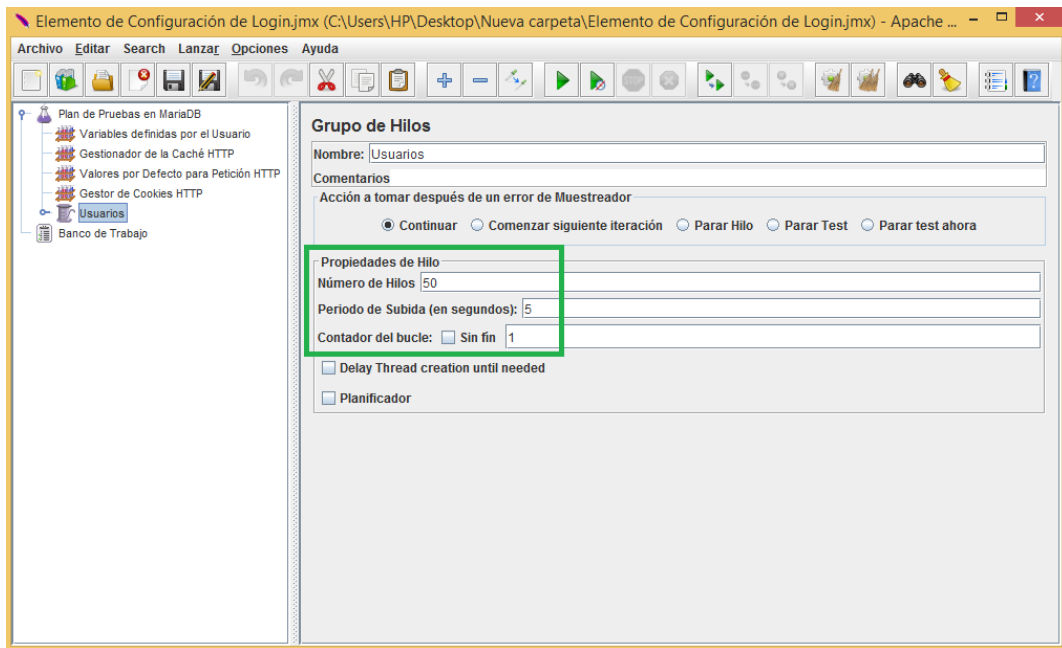


Figura 39. Valores para los Usuarios Conectados

En el caso de las peticiones realizadas a la página web **actividades.php** la configuración que se colocó fue la ruta de acceso a la misma.

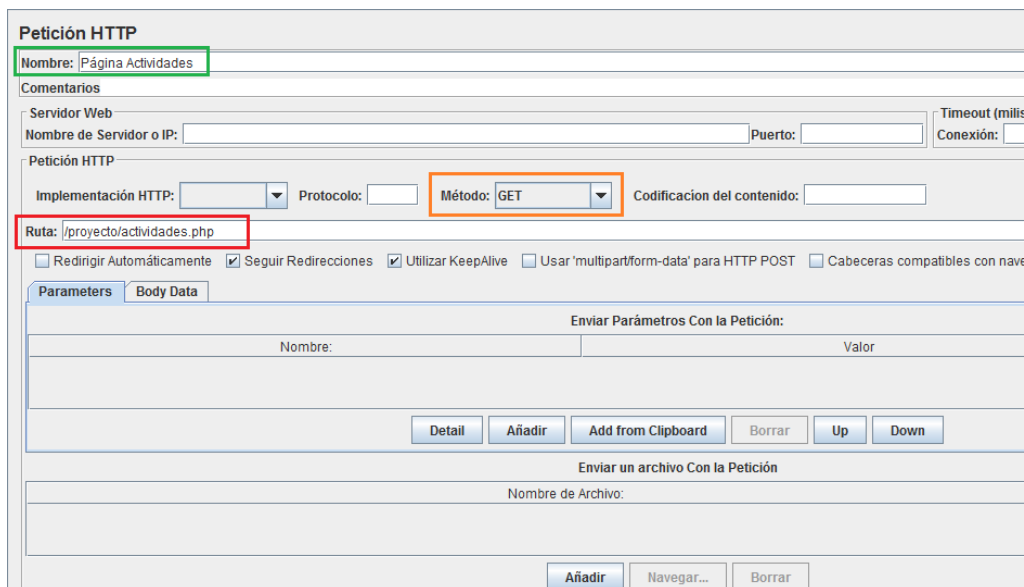


Figura 40. Petición HTTP hacia la página actividades.php

En el caso de la página web **actividades5.php** simplemente se cambia la ruta hacia la dirección donde se ubica la misma. El resto de parámetros se mantienen iguales.

Petición HTTP

Nombre: Página Actividades5

Comentarios

Servidor Web

Nombre de Servidor o IP: Puerto: Timeout (r Conexión:

Petición HTTP

Implementación HTTP: Protocolo: Método: **GET** Codificación del contenido:

Ruta: proyecto/actividades5.php

Redirigir Automáticamente Seguir Redirecciones Utilizar KeepAlive Usar 'multipart/form-data' para HTTP POST Cabeceras compatibles con navegadores

Parameters Body Data

Enviar Parámetros Con la Petición:

Nombre:	Valor

Detail Añadir Add from Clipboard Borrar Up Down

Enviar un archivo Con la Petición

Nombre de Archivo:

Figura 41. Petición HTTP hacia la página actividades5.php

Finalmente se creó una petición HTTP para la página web **contacto.php**, usando el método POST para ingresar valores en los campos pertenecientes a la tabla contacto.

Petición HTTP

Nombre: Página de Contacto

Comentarios

Servidor Web

Nombre de Servidor o IP: Puerto: Timeout (milisegundos) Conexión: Respuesta:

Petición HTTP

Implementación HTTP: Protocolo: Método: **POST** Codificación del contenido:

Ruta: proyecto/contacto.php

Redirigir Automáticamente Seguir Redirecciones Utilizar KeepAlive Usar 'multipart/form-data' para HTTP POST Cabeceras compatibles con navegadores

Parameters Body Data

Enviar Parámetros Con la Petición:

Nombre:	Valor	¿Codificar?	¿Incluir Equals?
NOM_CON	Anabel Pilicita	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
EMA_CON	ejemplo@upm.es	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
MEN_CON	Mensaje de ejemplo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Detail Añadir Add from Clipboard Borrar Up Down

Enviar un archivo Con la Petición

Nombre de Archivo: Nombre de Parámetro: Tipo MIME:

Figura 42. Petición http hacia la página contacto.php

4.5 Resultados

Una vez realizadas las configuraciones pertinente, se lanzaron las pruebas en los diferentes escenarios y como elemento de visualización se utilizó un reporte de resumen que entregó resultados con los siguientes parámetros que son explicados a continuación.

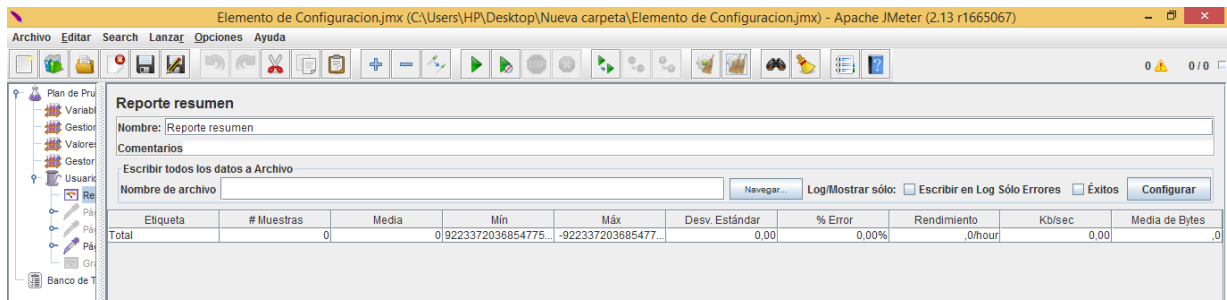


Figura 43. Reporte de Resumen

- *Número de Muestras*: Número de usuarios accediendo a la página web.
- *Media [s]*: Tiempo promedio invertido en una petición.
- *Mínimo [s]*: Tiempo mínimo invertido en una petición.
- *Máximo [s]*: Tiempo máximo que demora un usuario en acceder a una página web.
- *Error [%]*: Porcentaje de error en los accesos de los usuarios a la página web.
- *Rendimiento*: Número de peticiones procesadas en una cantidad de tiempo, que puede ser por segundo, minuto u hora.
- *Kb/s*: Al igual que el punto anterior representa el rendimiento pero relacionado con la cantidad de datos en lugar de peticiones.
- *Media en bytes*: Tamaño medio de datos como respuesta del servidor.

4.5.1 Resultados con MariaDB

Con el fin de consolidar los resultados obtenidos luego ejecutar las pruebas se elaboraron tablas y gráficas en el programa Excel, para una mejor visualización de los resultados entregados por Jmeter.

Se presentan los resultados en orden, partiendo de los 12 escenarios ejecutados mediante la conexión a MariaDB.

MARIADB: Página Web actividades.php								
Escenario	Usuarios	Tiempo Medio [ms]	Tiempo Mínimo de Acceso [ms]	Tiempo Máximo de Acceso [ms]	Desviación Estándar	Porcentaje de Error [%]	Rendimiento	Velocidad de acceso [Kb/s]
1	10	4219	3203	4863	544,85	0	1,3/s	892,31
2	50	15109	5554	26390	6175,32	0	1,7/s	1165,34
3	100	31132	5737	64504	18559,97	0	1,5/s	898,93
4	150	51021	6918	72414	20391,97	0	2,0/s	572,54

Tabla 10. Resultados escenarios 1-4 MariaDB

Los resultados muestran que los tiempos de acceso se aumentan a medida que incrementa el número de usuarios que acceden a la página web. Donde el tiempo promedio de acceso cuando acceden 10 usuarios es de 2,4 segundos, en el caso de 50 el tiempo es de 15 segundos, cuando hay 100 usuarios se tienen un promedio de 31 segundos y finalmente con 150 usuarios 51 segundos.

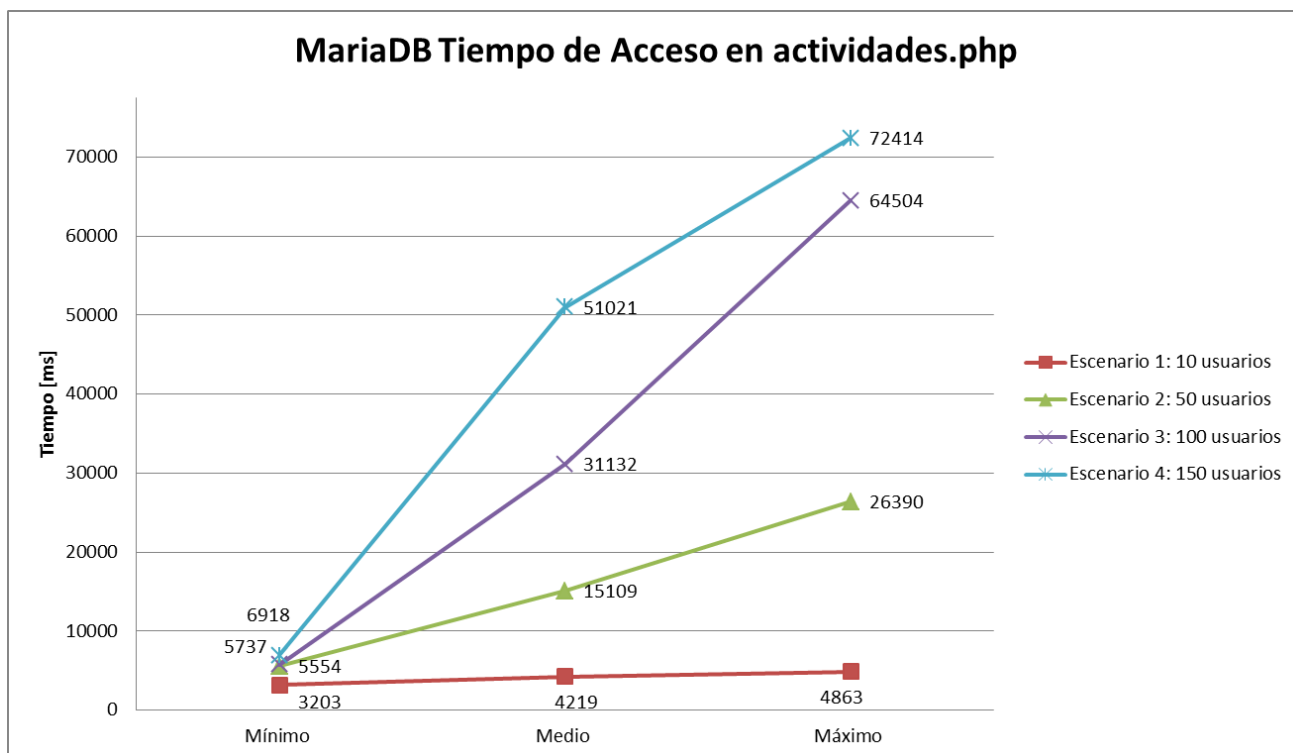


Figura 44. Gráfico actividades.php en MariaDB

Los valores de los tiempos son graficados indicándonos el incremento de los mismos a medida que aumenta la cantidad de usuarios. Al tener valores en función del tiempo a continuación se realiza la misma prueba con la diferencia que la cantidad de datos mostrados se limita a los 5 primeros registros alojados en la base de datos, usando la página web actividades5.php.

MariaDB: Página Web actividades5.php								
Escenario	Usuarios	Tiempo Medio [ms]	Tiempo Mínimo de Acceso [ms]	Tiempo Máximo de Acceso [ms]	Desviación Estándar	Porcentaje de Error [%]	Rendimiento	Velocidad de acceso [Kb/s]
5	10	57	56	61	1,36	0	2,2/s	17,21
6	50	60	56	74	4,61	0	10/s	78,73
7	100	69	56	98	9,39	0	19,8/s	155,09
8	150	107	59	193	92,36	0	29,4/s	230,50

Tabla 11. Resultados escenarios 5-8 MariaDB

Los valores en función del tiempo a comparación de los resultados anteriores han disminuido en gran medida obteniendo valores de acceso promedio en 10 usuarios de 57ms, en el caso de 50 el tiempo es de 60ms, cuando hay 100 usuarios se tienen un promedio de 69ms y finalmente con 150 usuarios 107ms. Estos valores son correctos y están dentro de los rangos que los usuarios admiten a la espera de acceder a un sitio web.

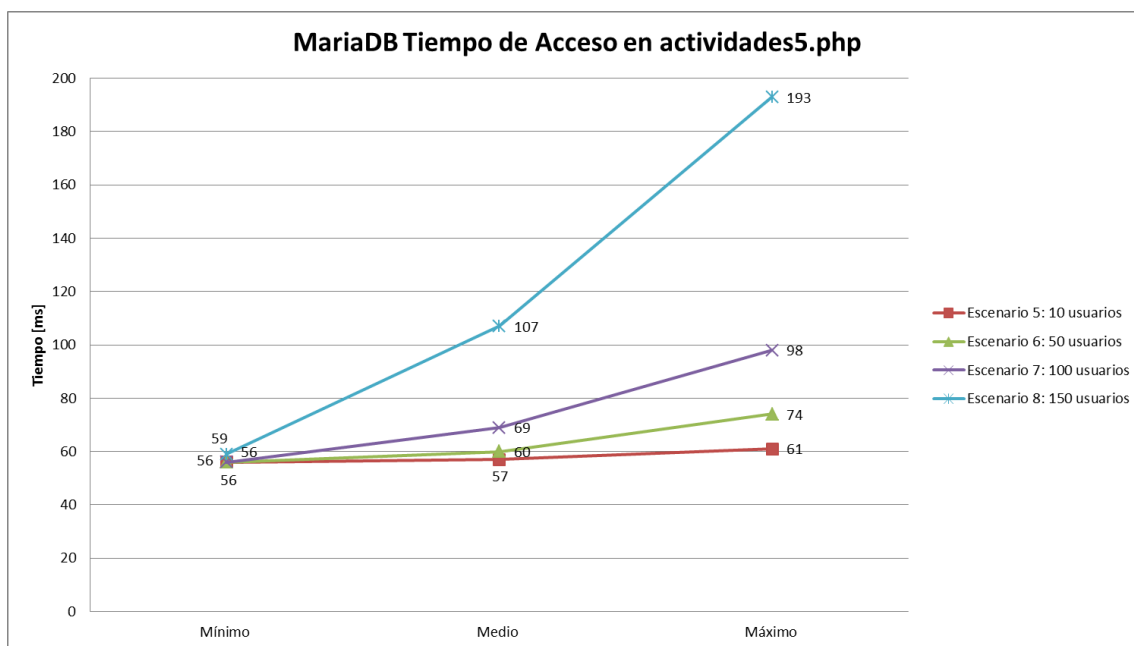


Figura 45. Gráfico actividades5.php en MariaDB

La visualización ayuda a comprender el aumento del tiempo en función de los usuarios que acceden al sitio tomando como referencia valores mínimo, promedio y máximo. Los resultados no superan los 2 o 3 segundos.

A continuación se presentan los resultados obtenidos utilizando la página web contacto.php, a diferencia de las anteriores en esta se llena un formulario para ingresar datos usando una petición http mediante un POST.

MariaDB: Página Web contacto.php								
Escenario	Usuarios	Tiempo Medio [ms]	Tiempo Mínimo de Acceso [ms]	Tiempo Máximo de Acceso [ms]	Desviación Estándar	Porcentaje de Error [%]	Rendimiento	Velocidad de acceso [Kb/s]
9	10	70	55	90	32,33	0	2,2/s	13,22
10	50	73	57	168	27,69	0	10,0/s	60,32
11	100	104	57	400	6,60	0	19,8/s	119,18
12	150	220	63	760	7,32	0	29,5/s	117,57

Tabla 12. Resultados escenarios 9-12 MariaDB

Los resultados muestran valores en la misma escala de la página web actividades5.php. Donde, el tiempo promedio con 10 usuarios es de 70ms, con 50 usuarios es de 73ms, frente a 100 usuarios es de 104ms y finalmente con 150 usuarios es de 220ms. Por lo tanto nos valores acorde con lo esperado, ya que los tiempos no son excesivamente altos como el caso de actividades.php.

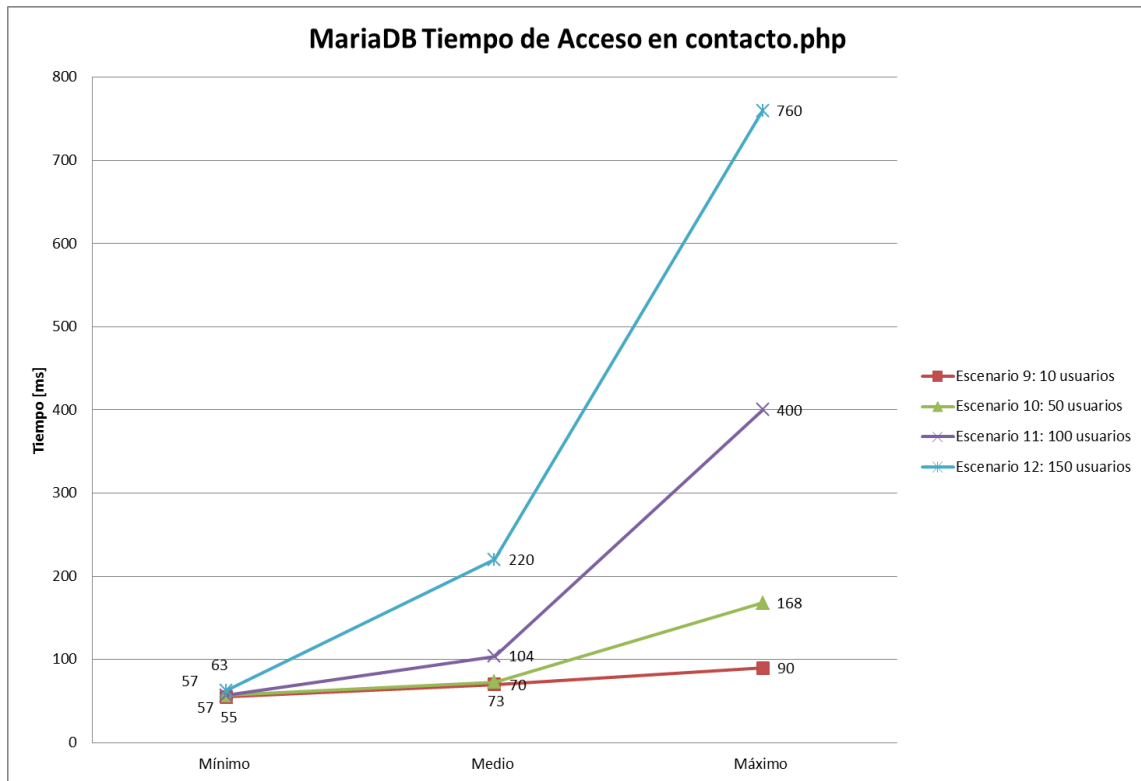


Figura 46. Gráfico contacto.php en MariaDB

Es importante recalcar que todos los accesos a las tres páginas web tuvieron éxitos y se presentó un porcentaje de error de 0%. A continuación se presentan los resultados obtenidos con PostgreSQL.

4.5.2 Resultados con PostgreSQL

Bajo los mismos parámetros de las pruebas en MariaDB, se realizó las pruebas en PostgreSQL para los 12 escenarios relacionados con las páginas web *actividades.php*, *actividades5.php* y *contacto.php*. Por lo tanto, se recopiló todos los resultados generados por Jmeter para mostrarlos en las siguientes tablas.

PostgreSQL: Página Web actividades.php								
Escenario	Usuarios	Tiempo Medio [ms]	Tiempo Mínimo de Acceso [ms]	Tiempo Máximo de Acceso [ms]	Desviación Estándar	Porcentaje de Error [%]	Rendimiento	Velocidad de acceso [Kb/s]
1	10	4150	3192	4755	492,83	0	1,3/s	892,78
2	50	14812	5813	24075	5790,10	0	1,8/s	1220,22
3	100	38978	8067	87573	25729,14	0	1,0/s	715,05
4	150	62722	6079	96354	31605,51	0	1,9/s	603

Tabla 13. Resultados escenarios 1-4 PostgreSQL

Al visualizar los resultados uno de los indicadores nos dice que todas las peticiones han sido atendidas ya que el porcentaje de error se mantiene en todos los escenarios en 0%. Sin embargo los tiempos de acceso a la página web nos permiten realizar la siguiente gráfica.

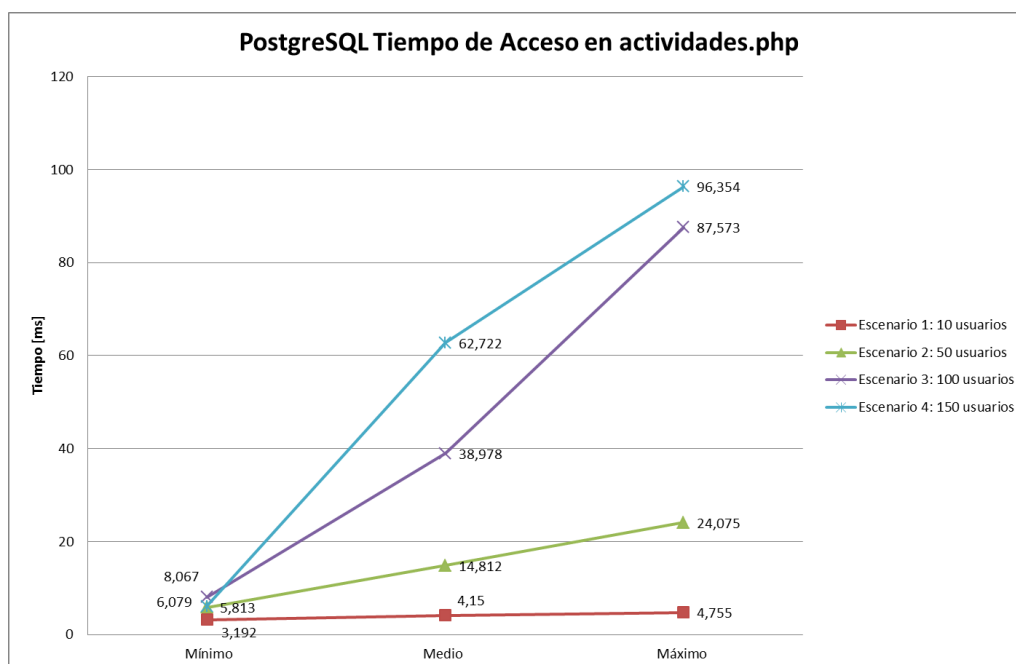


Figura 47. Gráfico actividades.php en PostgreSQL

El tiempo que tardan en ser atendidas las peticiones de acceso dependen del número de usuarios que soliciten acceso, ya que como se muestra en la Figura 44 el tiempo aumenta mientras mayor sea el número de usuarios. En el caso de esta página web se obtuvo valores de tiempo que exceden muy por encima los 2 a 3 segundos que es el tiempo que se estipula para que un usuario espere.

Al obtener estos valores se realizaron pruebas con la página web actividades donde se realiza la misma consulta a la base de datos pero al usuario muestra solo los 5 primeros valores de la tabla actividades de la base de datos.

PostgreSQL: <i>Página Web actividades5.php</i>								
Escenario	Usuarios	Tiempo Medio [ms]	Tiempo Mínimo de Acceso [ms]	Tiempo Máximo de Acceso [ms]	Desviación Estándar	Porcentaje de Error [%]	Rendimiento	Velocidad de acceso [Kb/s]
5	10	100	95	120	6,93	0	2,2/s	17,08
6	50	103	96	129	6,97	0	10,0/s	78,15
7	100	247	83	557	124,01	0	19,7/s	154,70
8	150	1323	213	2508	420,07	0	23,9/s	187,55

Tabla 14. Resultados escenarios 5-8 PostgreSQL

De igual forma como se visualiza en la Tabla 14 todos los accesos tuvieron éxito y los tiempos de acceso disminuyeron considerablemente a valores más acorde con lo esperado.

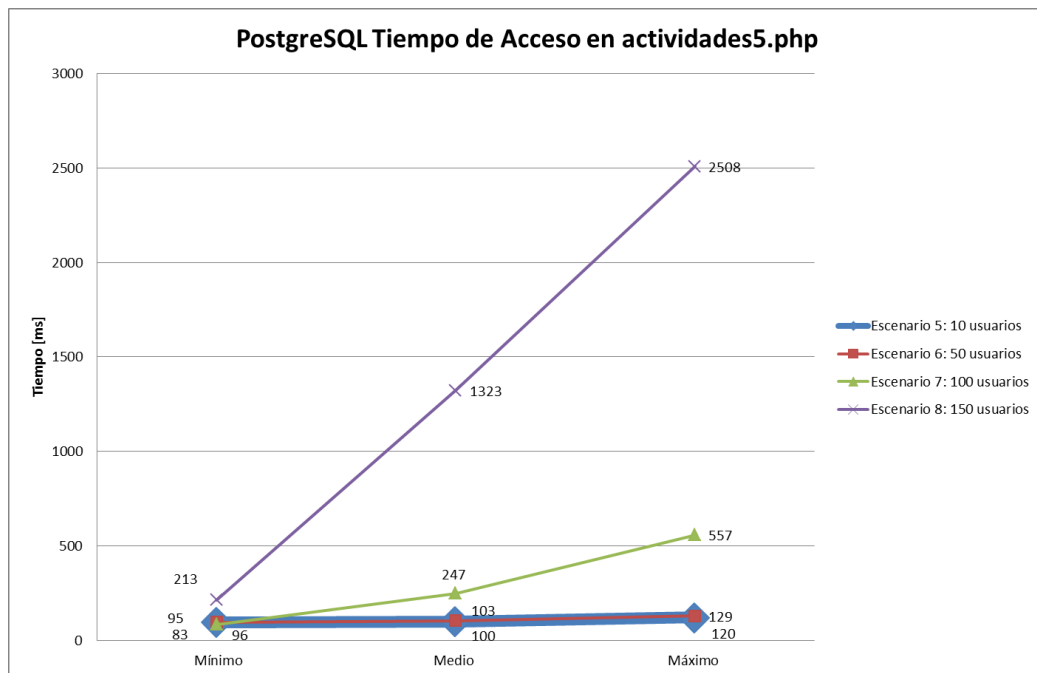


Figura 48. Gráfico actividades5.php en PostgreSQL

En el caso de la página web actividades5.php, los tiempos entre 10 a 50 accesos se mantiene semejantes con valores de media que rondan lo 100ms, aumentando a 247ms en 100 usuarios y a 1323ms en 150 usuarios. Por lo tanto los valores se encuentran en rangos medios aceptables como se visualiza en la Figura 45.

Finalmente se obtuvo resultados hacia la página web contacto.php donde se tiene valores de tiempo bajo y muy parecido a los de actividades5.php.

<i>PostgreSQL: Página Web contacto.php</i>								
Escenario	Usuarios	Tiempo Medio [ms]	Tiempo Mínimo de Acceso [ms]	Tiempo Máximo de Acceso [ms]	Desviación Estándar	Porcentaje de Error [%]	Rendimiento	Velocidad de acceso [Kb/s]
9	10	106	96	153	16,10	0	2,2/s	13,12
10	50	101	84	137	13,10	0	10,0/s	60,10
11	100	287	72	950	234,03	0	19,8/s	119,04
12	150	720	155	1582	313,52	0	28,01/s	169,00

Tabla 15. Resultados escenarios 9-12 PostgreSQL

Los tiempos más bajos se mantienen con 10 y 50 usuarios, aumentando al tener 100 usuarios con 287ms y se eleva a 720ms con 150 usuarios. La Figura 46 muestra el aumento del tiempo a medida que aumenta los usuarios conectados.

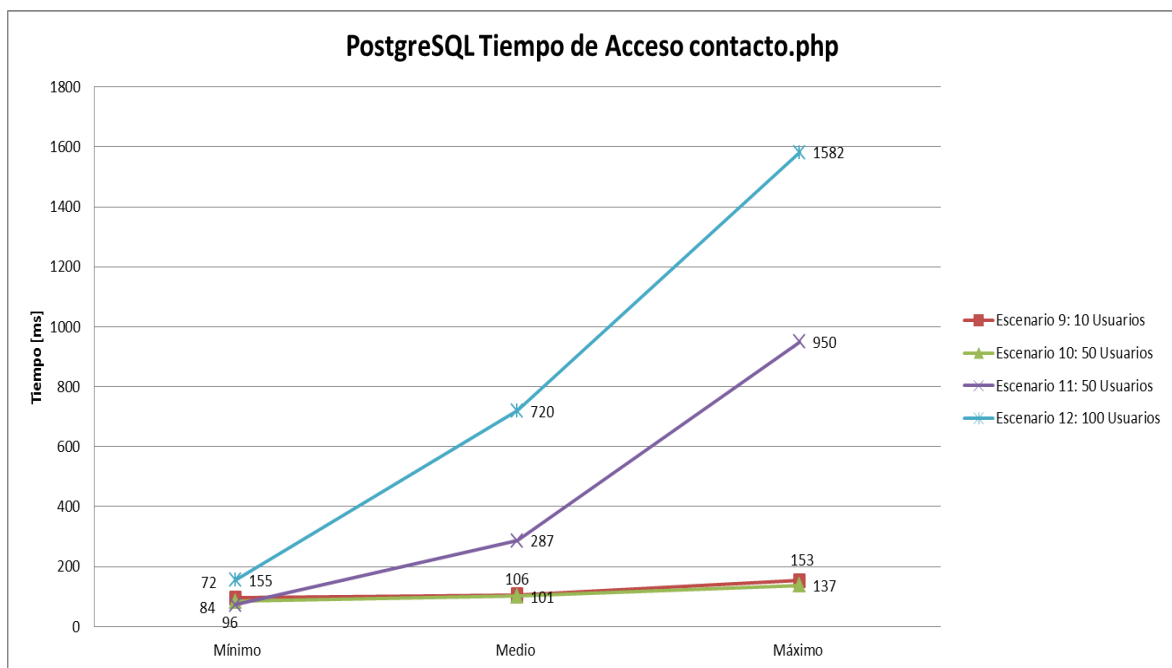


Figura 49. Gráfico contacto.php en PostgreSQL

4.6 Resultados en conjunto MariaDB y PostgreSQL

Luego de mostrar resultados por parte de MariaDB y PostgreSQL por separado. A continuación se presentan los resultados obtenidos en cada escenario comparando los parámetros de respuesta en MariaDB y PostgreSQL.

Cabe recalcar que los gráficos muestran todos los datos entregados por el resumen de resultados. Cada parámetro maneja diferentes unidades, por lo que no se especifican una escala o unidades en los ejes.

Si bien es cierto que graficamos todos los parámetros de resultados. Nos centramos a continuación en los tiempos promedio de acceso, en los que se refleja la mayor diferencia con respecto al tiempo que tarda en cargar las páginas web.

4.6.1 MariaDB y PostgreSQL actividades.php

Escenario 1: Específicamente en actividades.php con 10 usuarios accediendo simultáneamente tanto con MariaDB como con PostgreSQL se obtuvieron resultados similares con un rendimiento de 1,3[*peticiones/s*] en ambas base de datos. Los parámetros de resultados están graficados lo que permite ver que ambas bases de datos están a la par en función de los valores entregados. Al graficar los resultados vemos que la gráfica tiende a ser igual.

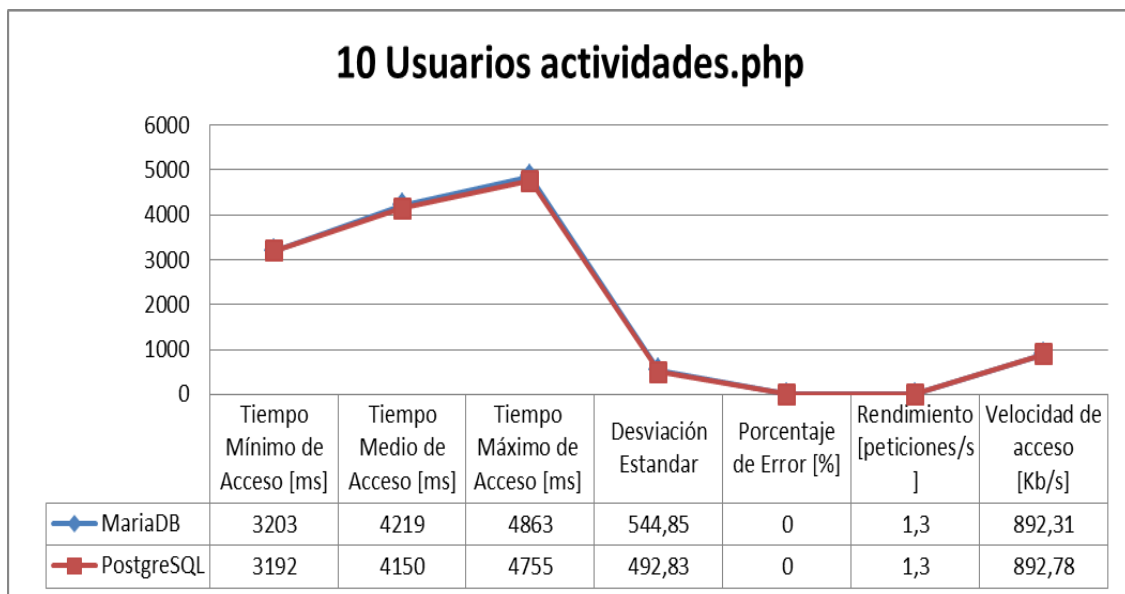


Figura 50. Gráfico 10 usuarios actividades.php PostgreSQL y MariaDB

Escenario 2: En actividades.php con 50 usuarios se visualiza, ligeramente un poco más bajos los tiempos de PostgreSQL que los de MariaDB específicamente en los tiempos máximos; así como también un ligero aumento del rendimiento en PostgreSQL. Pero los valores de resultado son muy similares.

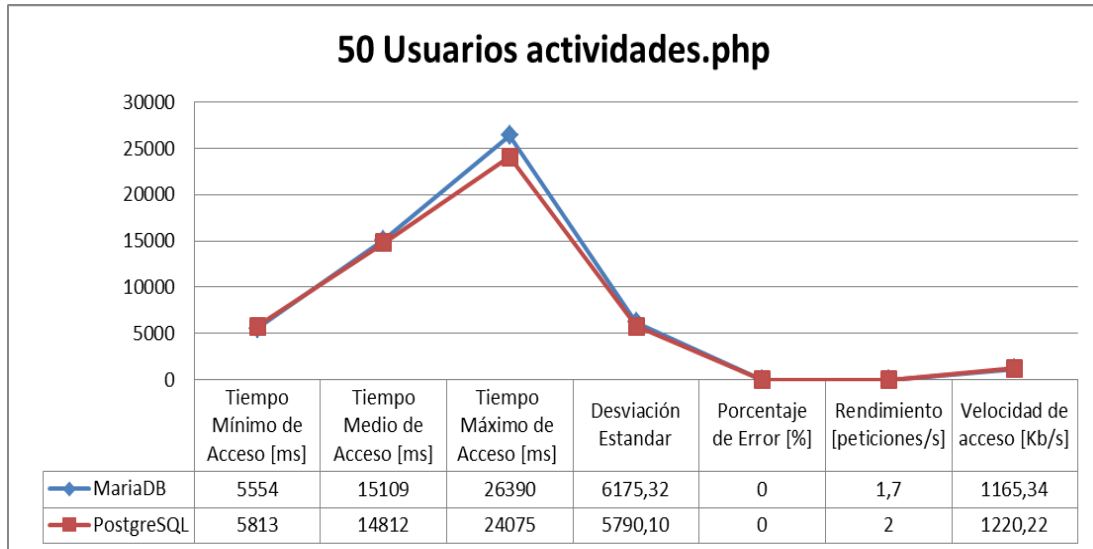


Figura 51. Gráfico 50 usuarios actividades.php PostgreSQL y MariaDB

Escenario 3: En actividades.php con 100 usuarios conectados simultáneamente los valores promedios de tiempo en MariaDB son de 31 segundos, y en el caso de PostgreSQL con 39 segundos. Se visualiza un ligero aumento en el rendimiento en MariaDB.

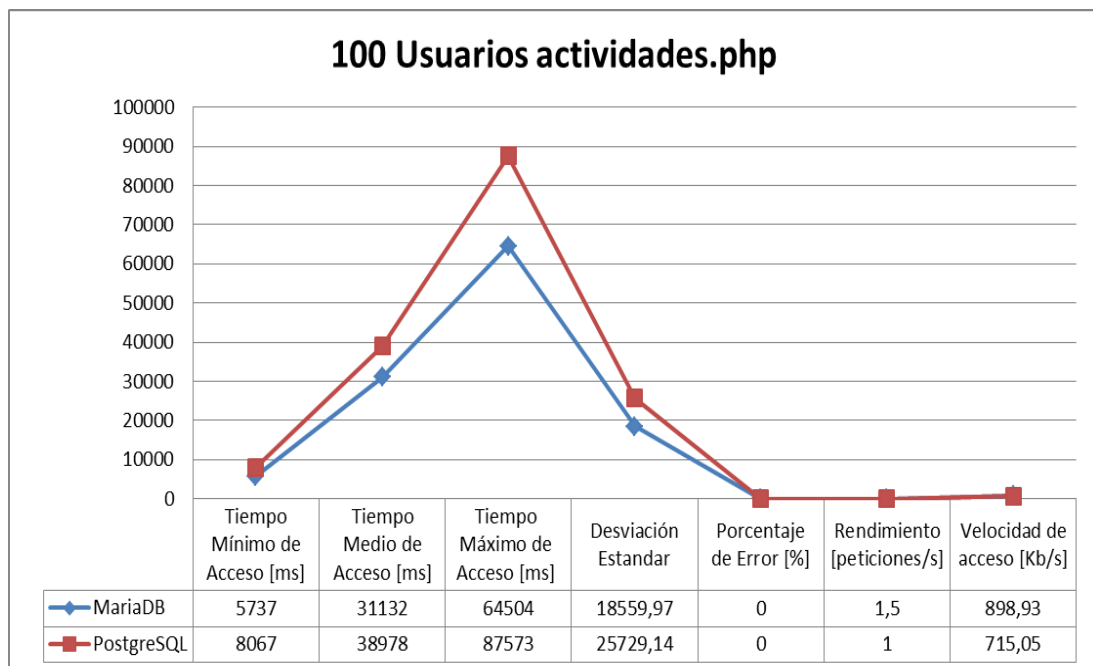


Figura 52. Gráfico 100 usuarios actividades.php PostgreSQL y MariaDB

Escenario 4: Con 150 usuarios en la página web actividades.php. El tiempo promedio de acceso es de 51 segundos en MariaDB y en PostgreSQL es de 52 segundos. El porcentaje de error es del 0% al igual que en todos los casos.

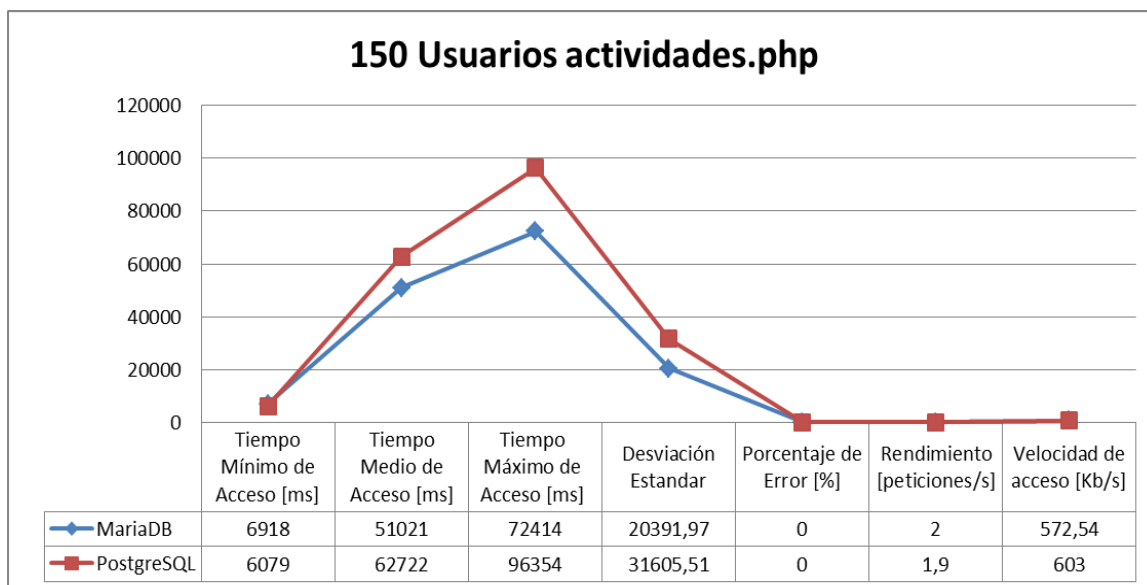


Figura 53. Gráfico 150 usuarios actividades.php PostgreSQL y MariaDB

4.6.2 MariaDB y PostgreSQL actividades5.php

Escenario 5: En actividades5.php con 10 usuarios conectados simultáneamente, se evidencia que los tiempos en MariaDB son menores a los de PostgreSQL. Donde el tiempo promedio es de 57 ms en MariaDB y en PostgreSQL es de 100ms. Una diferencia entre la una y la otra de 43ms.

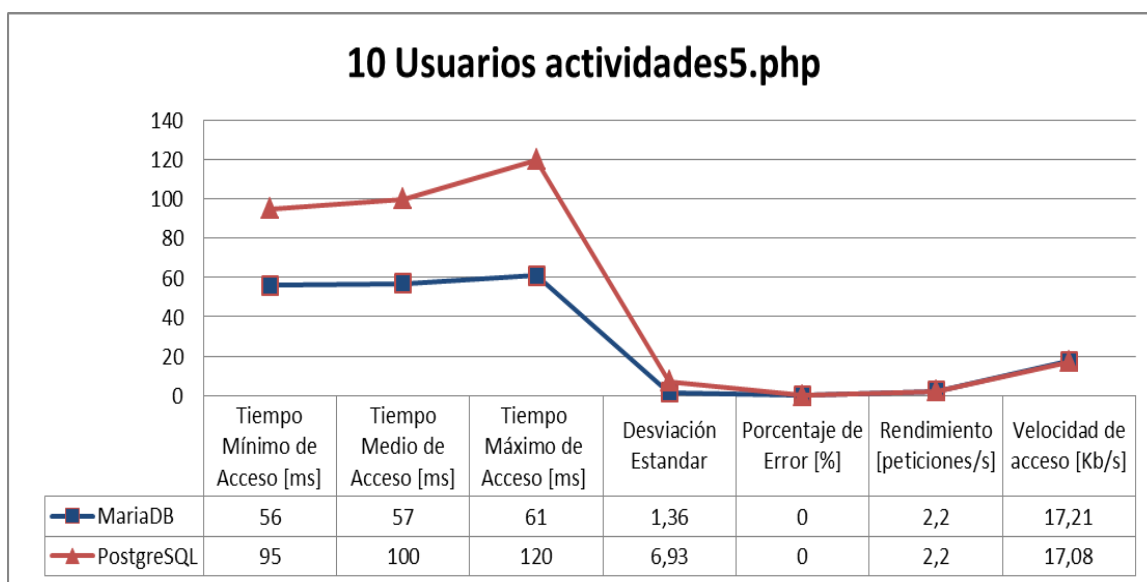


Figura 54. Gráfico 10 usuarios actividades5.php PostgreSQL y MariaDB

Escenario 6: En actividades5.php con 50 usuarios conectados simultáneamente los tiempos son menores en MariaDB con un tiempo promedio de 60 ms y en el caso de PostgreSQL de 103 ms. Se mantiene una diferencia de 43ms igual que el escenario 5.

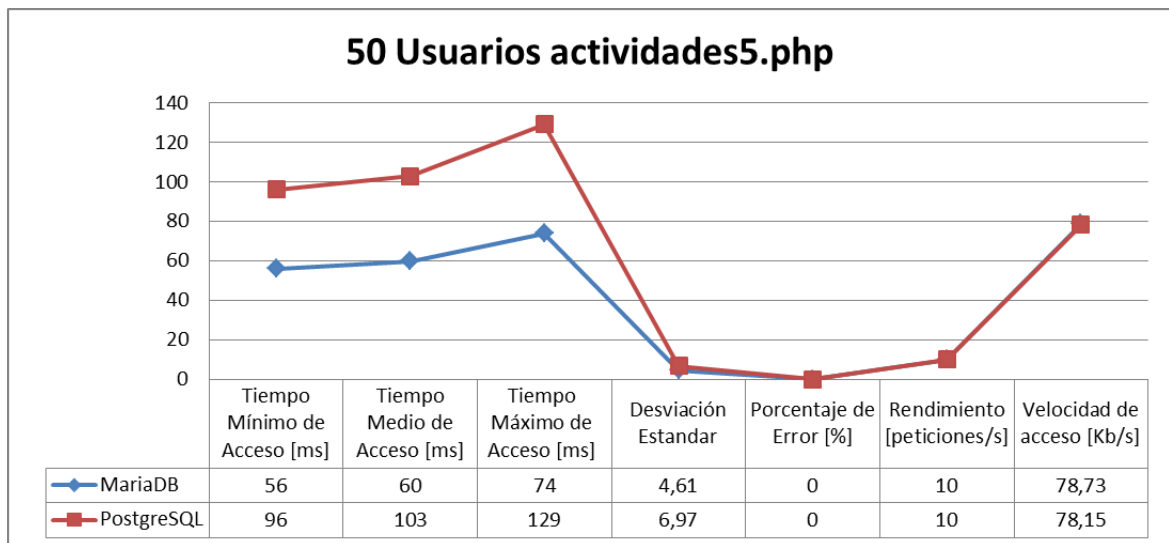


Figura 55. Gráfico 50 usuarios actividades5.php PostgreSQL y MariaDB

Escenario 7: En actividades5.php con 100 usuarios conectados simultáneamente se evidencia de forma muy clara que MariaDB tiene tiempos mucho más bajos que PostgreSQL como se visualiza en la Figura 56. Con un tiempo promedio de 69ms en MariaDB y en PostgreSQL de 247ms. La diferencia de tiempo aumenta en 178ms.

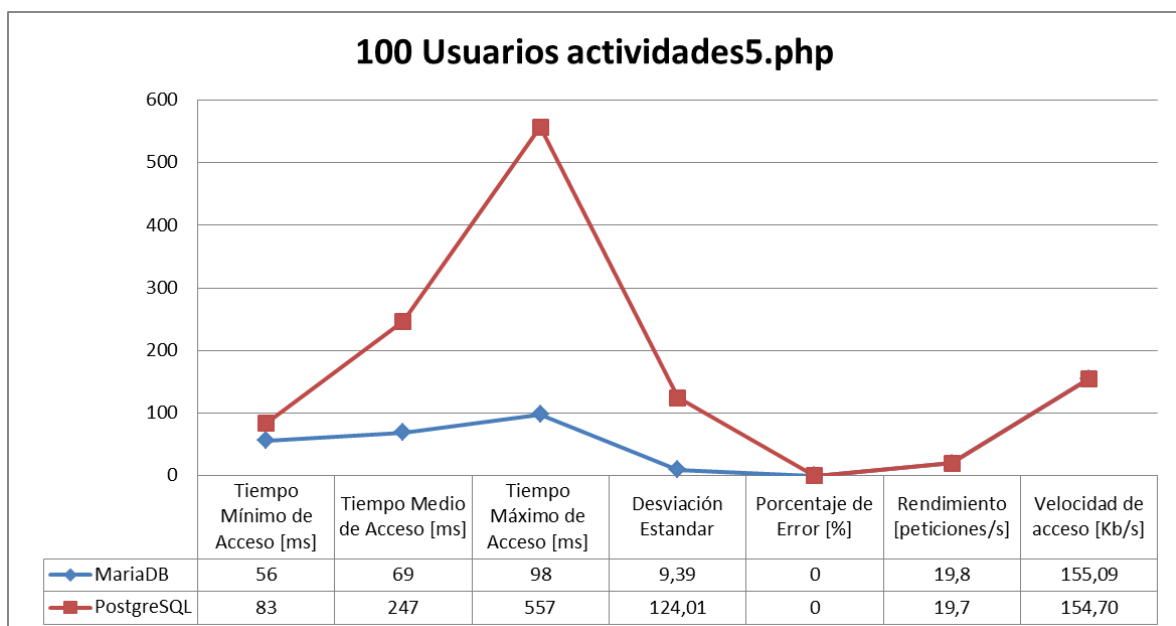


Figura 56. Gráfico 100 usuarios actividades5.php PostgreSQL y MariaDB

Escenario 8: En actividades5.php con 150 usuarios conectados simultáneamente el tiempo promedio de acceso en MariaDB es de 107ms y en el caso de PostgreSQL es de 1,3 segundos. Con 1 segundo de diferencia, es muy notable este cambio.

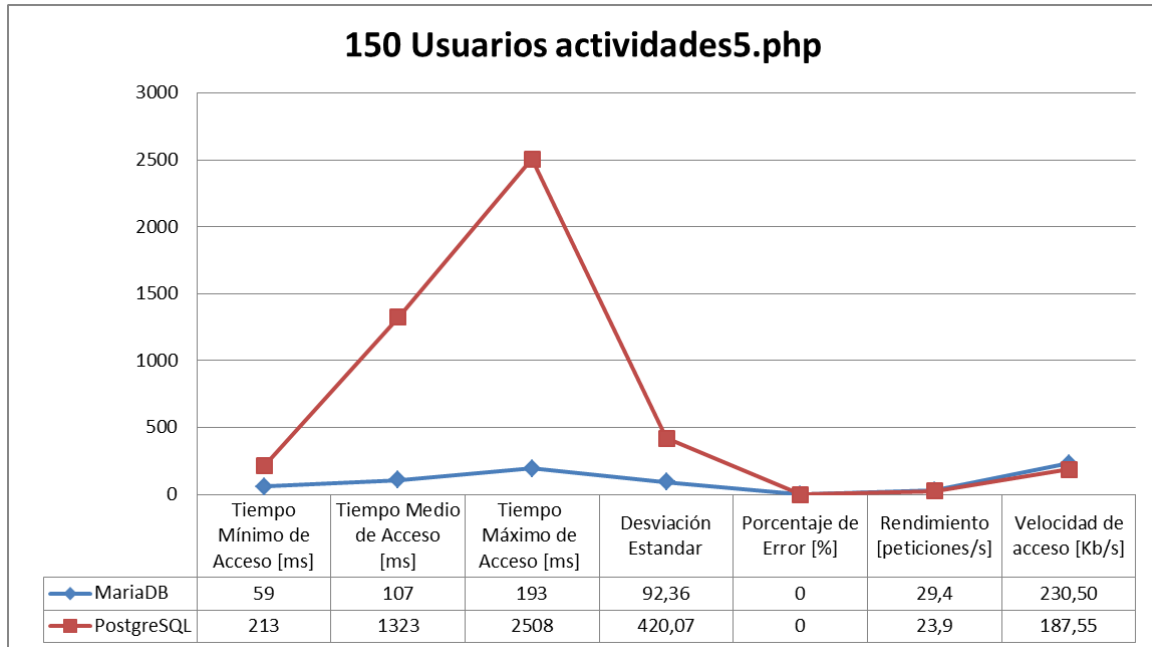


Figura 57. Gráfico 150 usuarios actividades5.php PostgreSQL y MariaDB

4.6.3 MariaDB y PostgreSQL contacto.php

Escenario 9: En contacto.php con 10 usuarios accediendo tenemos tiempo promedio de acceso en caso de MariaDB de 70ms y en el caso de PostgreSQL de 106ms. Tiempo de diferencia entre ambas de 36ms.

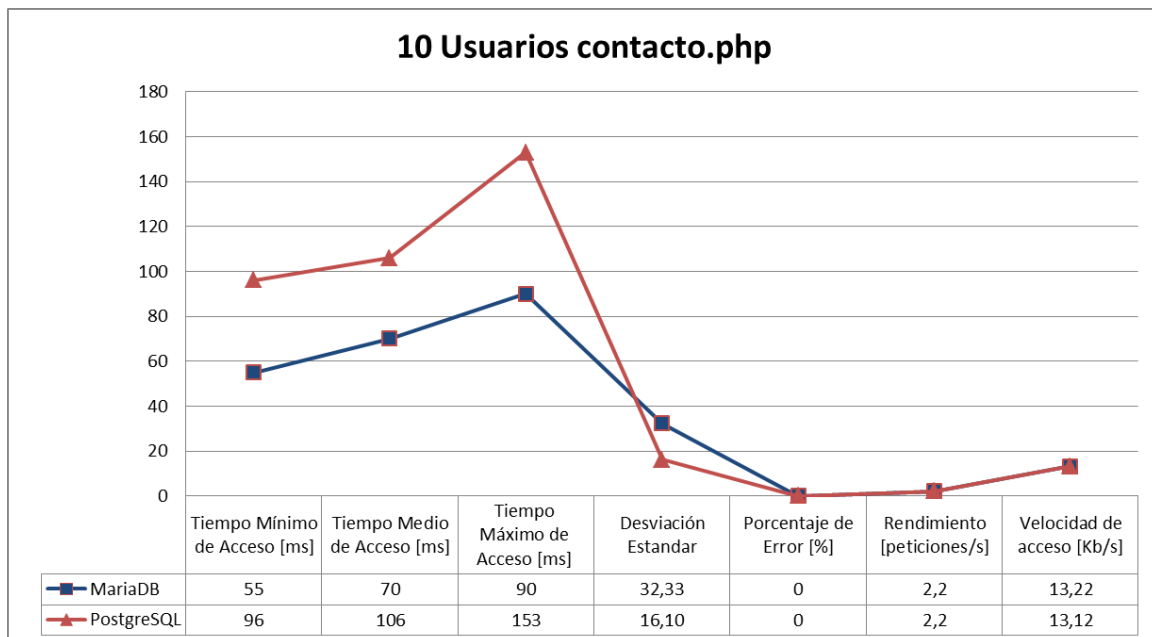


Figura 58. Gráfico 10 usuarios contacto.php PostgreSQL y MariaDB

Escenario 10: En contacto.php con 50 usuarios conectados simultáneamente el tiempo promedio es de 73ms en MariaDB y en PostgreSQL de 130ms. Diferencia de 57ms.

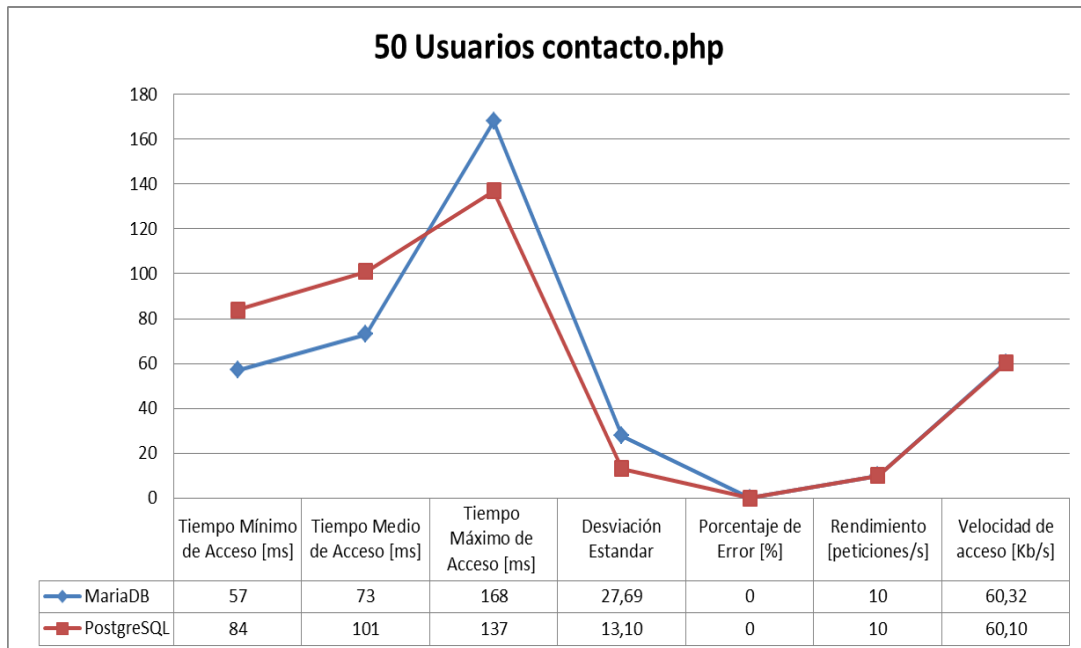


Figura 59. Gráfico 150 usuarios contacto.php PostgreSQL y MariaDB

Escenario 11: En contacto.php con 100 usuarios conectados simultáneamente el tiempo promedio es mejor en MariaDB con 104ms, en cambio con PostgreSQL es de 287ms. A medida que aumenta el número de usuarios la diferencia es mayor en este caso se tiene 183ms.

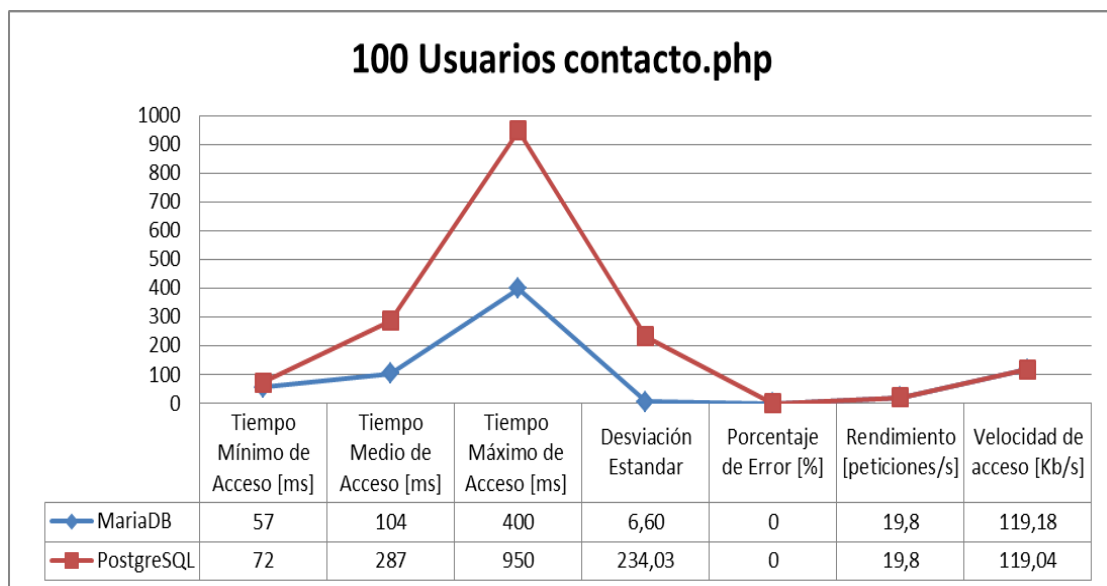


Figura 60. Gráfico 100 usuarios contacto.php PostgreSQL y MariaDB

Escenario 12: En contacto.php con 150 usuarios conectados en MariaDB, el tiempo promedio de acceso es de 220ms y en PostgreSQL es de 720ms. Es evidente que MariaDB se mantiene más estable que PostgreSQL y en este último escenario la diferencia entre sus tiempos es de 500 ms.

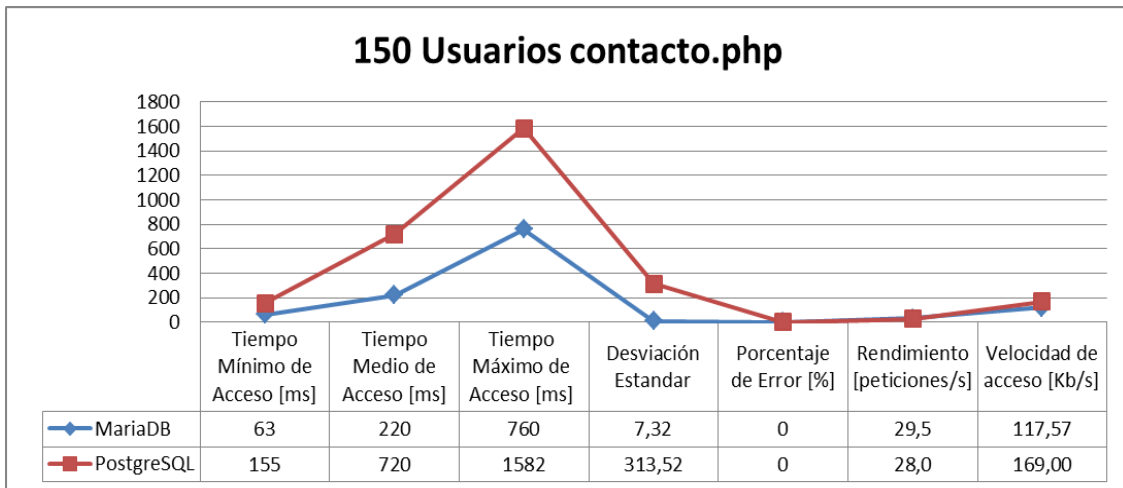


Figura 61. Gráfico 150 usuarios contacto.php PostgreSQL y MariaDB

4.6.4 Consolidación de Resultados MariaDB y PostgreSQL

A continuación se muestran los resultados en conjunto de la comparación entre los resultados obtenidos con MariaDB y PostgreSQL a medida que aumenta en número de usuarios en cada página web en función del tiempo promedio.

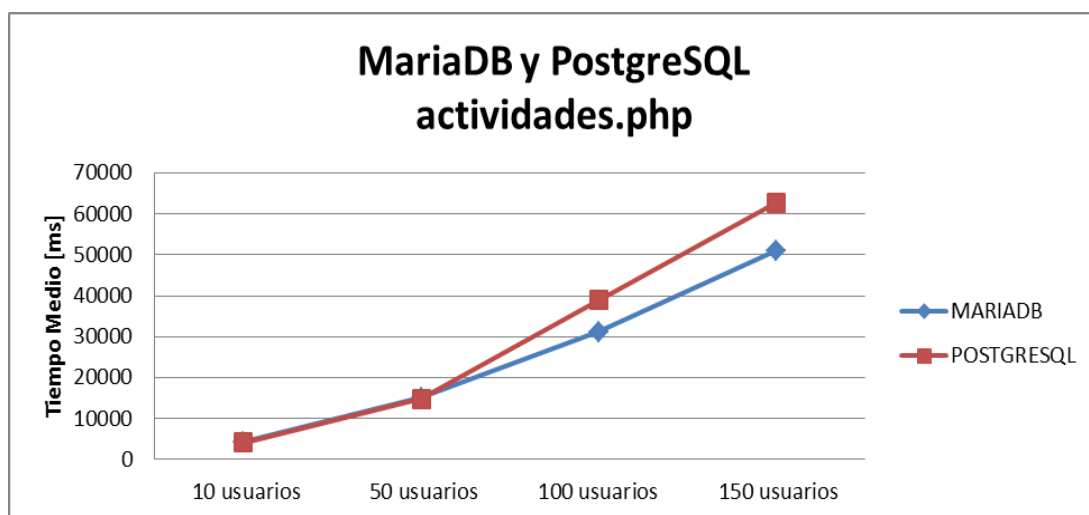


Figura 62. MariaDB vs PostgreSQL actividades.php

En el caso particular de la página web actividades.php los tiempos de espera son altos por ser una página web con gran cantidad de contenido. Por lo tanto los no se visualiza claramente los cambios que se generan al cambiar las bases de datos.

En el caso de actividades5.php el tiempo medio de acceso en todos los casos es menor en MariaDB; y a medida que aumenta el número de usuarios la diferencia entre las dos aumenta. Inclusive al tener 150 usuarios conectados simultáneamente el tiempo llega a ser superado en 1 segundo.

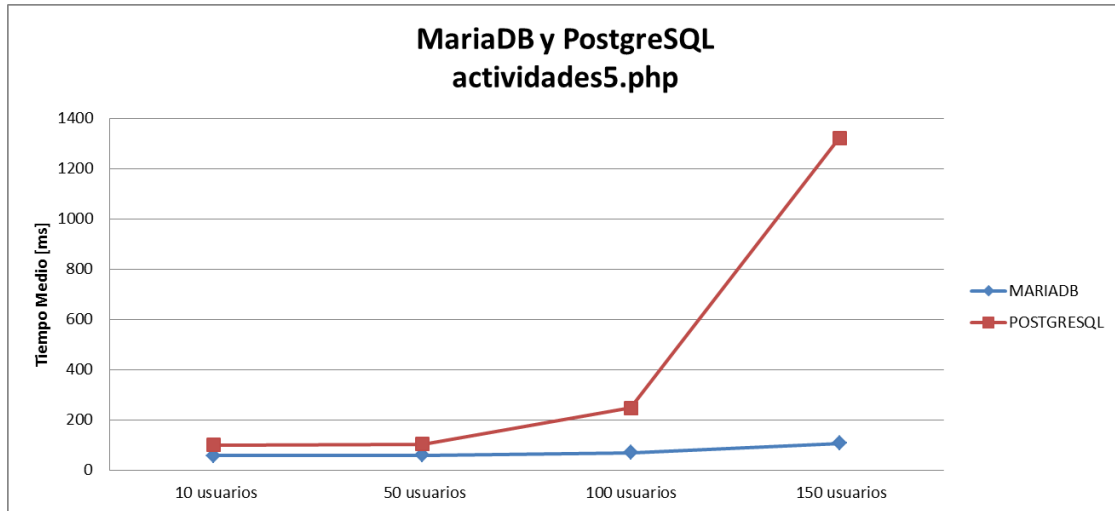


Figura 63. MariaDB vs PostgreSQL actividades5.php

En el caso de la página web contacto.php. La gráfica es similar al caso anterior. El tiempo de acceso promedio independientemente del número de usuarios es más bajo al utilizar MariaDB. Por lo tanto MariaDB luego de realizar las respectivas pruebas bajo los escenarios planteados ha presentado mejores resultados que PostgreSQL.

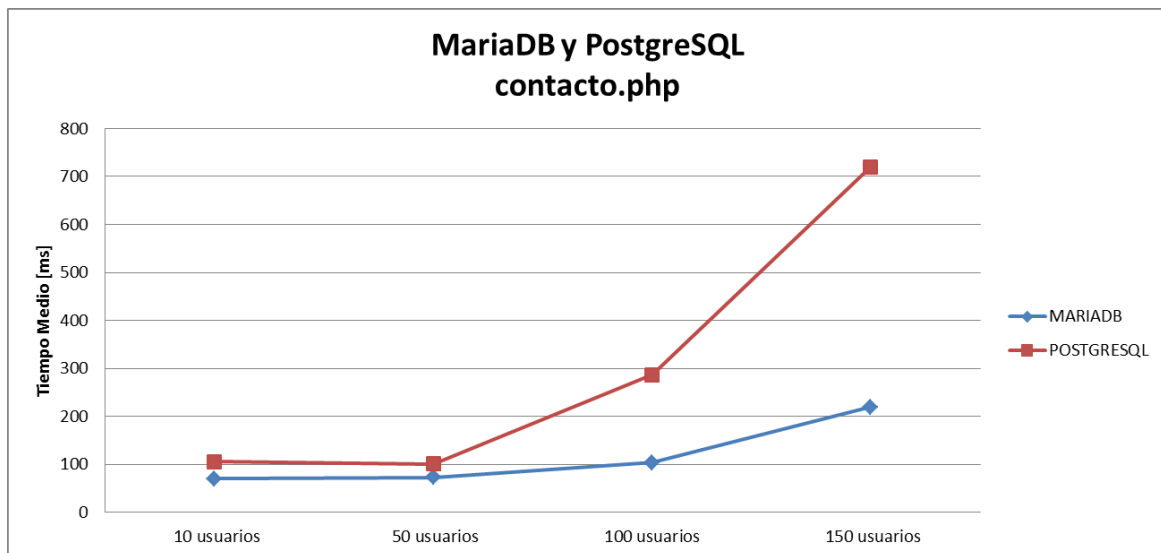


Figura 64. MariaDB vs PostgreSQL contacto.php

4.7 Discusión

Dentro de los objetivos del presente trabajo está el analizar los resultados obtenidos luego del diseño e implementación de una aplicación web de la cual se tomaron tres páginas web para su conexión con una base de datos en MariaDB y luego con otra base de datos alojada en PostgreSQL, con el fin de determinar el rendimiento en cada caso. A continuación, se pone a conocimiento los principales hallazgos luego de realizar las pruebas con Apache Jmeter.

De los resultados obtenidos tras realizar las respectivas pruebas, se puede afirmar que el tiempo en atender las peticiones hacia las tres páginas web indistintamente de la base de datos conectada se incrementa a medida que el número de usuarios aumenta. Esto se debe a que si el número de usuarios conectados simultáneamente no es muy grande como es el caso de 10 usuarios el servidor atiende en menor tiempo las peticiones, a diferencia de tener 150 usuarios conectados. Aquí se evidencia la cantidad de peticiones procesadas por segundo las mismas que aumentan debido al procesamiento que debe realizar el servidor.

Al hablar específicamente de los tiempos de carga de una página web los rangos que se manejan son de 2 a 3 segundos proporcionando una mejor experiencia al usuario. En el caso particular de la página web actividades.php los tiempos sobrepasaban mucho el rango ya mencionado esto se debe principalmente a que esta página web mostraba los 1000 registros almacenados en la tabla ACTIVIDADES de cada una de la base de datos. Por lo tanto, el hecho de visualizar una página web compuesta por muchos elementos influye en el tiempo de acceso a la misma. Por otra parte es importante tomar a consideración las características del equipo donde se realizaron las pruebas lo cual también es una de las razones para obtener tiempos sean altos en el procesamiento de peticiones.

Los tiempos son para cargar la página web actividades.php, por lo que no se llega a observar claramente las diferencias. Se decidió realizar pruebas en páginas compuestas con una menor cantidad de elementos.

En el caso de actividades5.php fue notable la reducción en los tiempos de acceso ya que en todos los escenarios con 10, 50,100 y 150 usuarios los tiempos de acceso promedio no sobrepasaron los 3 segundos, esto como ya se mencionó debido a ser una página que tiene menor cantidad de elementos. Similares resultados fueron evidenciados en la página web contacto.php la misma que cuenta con un formulario sin muchos componentes. Tiempos aceptables dentro de un rango que genera a los usuarios una mejor experiencia al cargar una página web.

Al hablar del número de usuarios conectados a las páginas web actividades.php, actividades5.php y contacto.php, se evidencia específicamente con 10 y 50 que los tiempos son menores por parte de MariaDB que de PostgreSQL la diferencia es alrededor de 40ms. Pero donde realmente se evidencian los mayores cambios es a medida que aumenta la cantidad de usuarios específicamente con 100 y 150 usuarios donde es evidente que MariaDB ofrece mejores tiempos que PostgreSQL con diferencias de más de 500ms. Lo que significa que el tiempo promedio invertido en una petición en MariaDB es mejor que el realizado en PostgreSQL bajos los parámetros definidos en este trabajo.

Luego de realizar pruebas y comparar resultados a partir de los escenarios propuestos y bajo todas las herramientas utilizadas se observó que el comportamiento de MariaDB es mejor que PostgreSQL en función de lo tiempos obtenidos. El tiempo de carga es un factor fundamental y de gran interés para captar día a día más usuarios, por lo tanto es una referencia muy importante a la hora de elegir un Sistema de Gestión de Base de Datos.

CAPÍTULO 5

5 Conclusiones y líneas Futuras

En el presente trabajo se ha diseñado, desarrollado e implementado una aplicación web para una asociación de docentes. Partiendo de esta aplicación se ha manipulado bases de datos en MariaDB y PostgreSQL con el fin de realizar pruebas con la herramienta Jmeter y poder comparar los resultados. El presente trabajo propuso objetivos cumplidos a lo largo de su desarrollo por lo cual se puede concluir y plantear líneas futuras.

5.1 Conclusiones

- Para el desarrollo del presente trabajo se aprendió a manejar y a utilizar herramientas como Bootstrap, PHP, phpMyAdmin, pgAdmin III, Apache Jmeter cada una de ellas permitió la consolidación del trabajo. Por lo considero que en la actualidad herramientas de tipo código abierto son de gran ayuda y de vital importancia para el desarrollo de proyectos.
- Con respecto al diseño de las páginas web el uso de Bootstrap y sus componentes facilitó el maquetado de las mismas; este tipo de herramientas contribuyen a disminuir el tiempo de diseño que emplea un programador para la elaboración de las páginas web. Todo esto en base a la reutilización de código obteniendo como trabajo final páginas web con un aspecto más profesional que resultan atractivas para el usuario final.
- El modelo Entidad-Relación es fundamental en el diseño de una base de datos, ya que permite entender como es la estructura de la base de datos diseñada y entender como están constituidas las entidades y las relaciones que se generan entre ellas.
- Para la lógica de la aplicación se utilizó el lenguaje PHP que permitió enlazar las peticiones del usuario con las bases de datos. Es un lenguaje fácil de entender y programar con gran cantidad de tutoriales e información. Esto facilitó en gran medida el desarrollo de la aplicación web.
- Apache Jmeter es una herramienta que permitió el analizar el comportamiento del entorno montado ante diferentes escenarios, y comprobar los resultados frente a cambios de diferentes parámetros.
- Las pruebas de carga realizadas permitieron comprobar el rendimiento cuando un número de usuarios acceden de manera concurrente, donde se verificó aspectos como la escalabilidad y tiempo de respuesta. Esto ayuda a garantizar que un sistema funcione en las peores circunstancias sin ningún problema

crítico detectando problemas de concurrencia que no son percibidos durante el desarrollo.

- Uno de los aspectos primordiales fue establecer un marco para comparar a MariaDB y PostgreSQL, donde se concluye que en función del equipo de prueba utilizado y los escenarios planteados en el presente trabajo se obtuvo mejores resultados utilizando a MariaDB conectada con la aplicación web. Tiempos promedio de acceso que llegaron a superar los 500ms.

5.2 Líneas Futuras

- El trabajo fue realizado bajo un equipo con características específicas de tipo personal. Por lo tanto resultaría interesante realizar las mismas pruebas en un equipo con mejores características técnicas; así como también el realizar pruebas utilizando otro tipo de base de datos.
- Los tiempos esperados por los usuarios al cargar una página web cada día son más exigentes por lo que brindar tiempos de carga óptimos es de vital importancia. Es ahí donde el programador web puede optimizando el código y hacer un buen uso de las plataformas para mejorar los tiempos.
- En el trabajo presentado se realizaron pruebas de carga y concurrencia. Por lo que es posible ampliar pruebas el plan de pruebas como pruebas de estrés.

Bibliografía

- [1] D. Riehle, *Framework Design: A Role Modeling Approach*. Swiss Federal Institute of Technology', 2000.
- [2] D. Cochran, *Twitter Bootstrap Web Development*. Packt Publishing, 2012.
- [3] *History of PHP and related projects*. The PHP Group.
- [4] "XAMPP - Wikipedia, la enciclopedia libre." [Online]. Available: <https://es.wikipedia.org/wiki/XAMPP>. [Accessed: 21-Apr-2016].
- [5] "phpMyAdmin." [Online]. Available: <https://www.phpmyadmin.net/>. [Accessed: 21-Apr-2016].
- [6] "Arquitectura del SGBD e independencia con respecto a los datos." [Online]. Available: <http://www.ub.edu.ar/catedras/ingenieria/Datos/capitulo1/cap14.htm>. [Accessed: 22-Apr-2016].
- [7] "MariaDB - Wikipedia, la enciclopedia libre." [Online]. Available: <https://es.wikipedia.org/wiki/MariaDB>. [Accessed: 22-Apr-2016].
- [8] Óscar Ray, "Una docena de herramientas para probar y mejorar el rendimiento de nuestra web," 2012.
- [9] I. Molyneaux, *The art of application performance testing*. O'Reilly Media, Inc, 2009.
- [10] "Capítulo 5. Frameworks (CSS avanzado)." [Online]. Available: http://librosweb.es/libro/css_avanzado/capitulo_5.html. [Accessed: 20-Apr-2016].
- [11] *GitHub: Popular Watched Repositories*. .
- [12] Mark Otto and M. Otto, *Bootstrap from Twitter*. 2011.
- [13] "Responsive CSS Framework Comparison: Bootstrap, Foundation, Skeleton." [Online]. Available: <http://responsive.vermilion.com/compare.php>. [Accessed: 21-Apr-2016].
- [14] *Historia de PHP y Proyectos Relacionados*. .
- [15] *PHP: Extensiones de bases de datos - Manual*. PHP Group.
- [16] *PHP: Using PHP from the command line - Manual*: The PHP Group.
- [17] *dompdf - HTML to PDF converter (PHP5)*. .
- [18] *What has changed in PHP 5.5.x*. .
- [19] *Built-in web server*. .
- [20] *Historical trends in the usage of server-side programming languages for websites*. W3Techs.

- [21] *Last 5.3 release ever available: PHP 5.3.29 - 5.3 now EOL.* PHP.
- [22] *RFC: PHP 7.0 Timeline.* 2015.
- [23] "XAMPP Installers and Downloads for Apache Friends." [Online]. Available: <https://www.apachefriends.org/es/index.html>. [Accessed: 21-Apr-2016].
- [24] "Desarrollo Web: Tabla comparativa de los sistemas gestores de base de datos." [Online]. Available: <http://desarrollowebydesarrolloweb.blogspot.com.es/2015/02/tabla-comparativa-de-los-sistemas.html>. [Accessed: 22-Apr-2016].
- [25] *PHP: Sintaxis básica.* .
- [26] "PHP: ¿Qué necesito? - Manual." [Online]. Available: <http://php.net/manual/es/tutorial.requirements.php>. [Accessed: 21-Apr-2016].
- [27] luismi, "Pruebas de Rendimiento con Apache Jmeter," *Blog*, 2015.
- [28] "53_mysql_guide_ES.pdf." .
- [29] D. P. Daswani, "Medición de Rendimientos de Servicios WMS con JMeter," 2008.