

**INSTITUTO TECNOLÓGICO SUPERIOR**

**“SAN GABRIEL”**



**PROYECTO DE TITULACIÓN**

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE:**

**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS DE  
SISTEMAS**

**TÍTULO:**

DISEÑO E IMPLEMENTACIÓN DE UN SITIO WEB PARA EL CONTROL DE MATERIALES Y EQUIPAMIENTOS DEL CUERPO DE BOMBEROS DEL CANTÓN MOCHA PROVINCIA DE TUNGURAHUA DESARROLLADO EN PHP ORIENTADO A OBJETOS Y GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2018

**AUTOR:**

ELENA MONSERRAT MARIÑO BARRENO

**RIOBAMBA-ECUADOR**

**2018**

## **CERTIFICACIÓN**

Certifico que el Sr. **ELENA MONSERRAT MARIÑO BARRENO**, con el N° de Cédula 1804751954 ha elaborado bajo mi Asesoría el Proyecto de Investigación titulado:

**DISEÑO E IMPLEMENTACIÓN DE UN SITIO WEB PARA EL CONTROL DE MATERIALES Y EQUIPAMIENTOS EN EL CUERPO DE BOMBEROS DEL CANTON MOCHA REALIZADO EN PHP ORIENTADO A OBJETOS Y EL GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2018**

Por tanto autorizo la presentación para la calificación respectiva.

---

**Ing. Andrea Moncayo**  
**DOCENTE DEL ITSGA**

“El presente Proyecto de Investigación constituye un requisito previo para la obtención del Título de **Tecnólogo en Informática Mención Análisis de Sistema**”



“Yo, **ELENA MONSERRAT MARIÑO BARRENO** con N° de Cédula 180475195-4, declaro que la investigación es absolutamente original, autentica, personal y los resultados y conclusiones a los que se han llegado es de mi absoluta responsabilidad.”

---

**ELENA MONSERRAT MARIÑO BARRENO**

**INSTITUTO TECNOLÓGICO SUPERIOR**  
**“SAN GABRIEL”**  
**ESPECIALIDAD INFORMÁTICA**  
**TESIS DE GRADO**  
**PREVIA A LA OBTENCION DEL TÍTULO DE:**  
**TECNÓLOGO EN INFORMÁTICA MENCIÓN ANÁLISIS DE**  
**SISTEMAS**

**TEMA:**

DISEÑO E IMPLEMENTACIÓN DE UN SITIO WEB PARA EL CONTROL DE MATERIALES Y EQUIPAMIENTOS EN EL CUERPO DE BOMBEROS DEL CANTON MOCHA REALIZADO EN PHP ORIENTADO A OBJETOS Y EL GESTOR DE BASE DE DATOS MYSQL EN EL PERIODO 2018

**APROBADO:**

**ASESOR DE TESIS DE GRADO** .....

**PRESIDENTE DEL TRIBUNAL** .....

**MIEMBRO DEL TRIBUNAL** .....

**MIEMBRO DEL TRIBUNAL** .....

**FIRMAS DE RESPONSABILIDAD**

**NOMBRES**

**FECHA**

**FIRMA**

**RECTORA**

.....

.....

**VICERRECTOR ACADÉMICO**

.....

.....

**DIRECTOR DE TESIS**

.....

.....

**MIEMBRO DEL TRIBUNAL**

.....

.....

**MIEMBRO DEL TRIBUNAL**

.....

.....

**MIEMBRO DEL TRIBUNAL**

.....

.....

**NOTA FINAL DE LA TESIS**

.....

## **DEDICATORIA**

La concepción de este trabajo de investigación está dedicada a mis padres, pilares fundamentales en mi vida. Sin ellos, jamás hubiese podido conseguir lo que hasta ahora. Su tenacidad y lucha insaciable han hecho, de ellos el gran ejemplo a seguir y destacar, no solo para mí, sino para mis hermanos y familia en general a mis abuelos quienes ya no están presentes pero con los consejos que me brindaron llegue hasta donde ahora estoy. También dedico este proyecto a una persona muy especial que forma parte de mi vida, compañero inseparable de cada jornada. A ellos este proyecto, que sin ellos, no hubiese podido ser.

Elena

## **AGRADECIMIENTO**

En primer lugar a Dios por haberme guiado por el camino de la felicidad hasta ahora; en segundo lugar a cada uno de los que son parte de mi familia a mi PADRE, mi MADRE mi ABUELA, a mis hermanos ; por siempre haberme dado su fuerza y apoyo incondicional que me han ayudado y llevado hasta donde estoy ahora. Por último a mis queridos profesores que me han impartido sus conocimientos a mi tutora de trabajo de investigación Ing. Andrea Moncayo quién me ayudó y tuvo paciencia en todo momento para poder culminar este trabajo.

Elena

## ABREVIATURAS

|                |   |
|----------------|---|
| <b>TCP/IP:</b> | Transmission Control Protocol/Internet Protocol |
| <b>CSS:</b>    | Cascading Style Sheets                          |
| <b>PDF:</b>    | Portable Document Format                        |
| <b>Swfs:</b>   | Small Web Format                                |
| <b>HTTP</b>    | Hypertext Transfer Protoco                      |
| <b>POO:</b>    | Programación Orientada a Objetos                |
| <b>PHP:</b>    | Hypertext Preprocessor                          |
| <b>HTML:</b>   | HyperText Markup Language                       |
| <b>SQL:</b>    | Structured Query Language                       |
| <b>IIS:</b>    | Internet Information Services                   |
| <b>WWW:</b>    | World Wide Web                                  |
| <b>URL:</b>    | Uniform Resource Locator                        |
| <b>GPL:</b>    | General Public License                          |



## GLOSARIO DE TÉRMINOS

**PHP:** siglas en ingles se hipertext preprocesor, lenguaje de programación que puede ser incrustado en páginas HTML.

**MySQL:** sistema de administración de base datos sus siglas en ingles Database Management system, sistema basado en SQL

**POO:** Programación Orientada a Objetos, paradigma en programación que revolucionó la forma de programar

**Webmaster:** proviene de 2 palabras compuestas que son web (red) y master (maestro), por tanto su significado estaría establecido por maestro de la red

**Custodio:** persona que se encarga de vigilar el buen desarrollo del sistema

**C y C++:** Lenguajes de programación en los cuales están basados la mayoría de programas de software libre.

**Apache:** servidor de software libre, compatible con una gran variedad de sistemas operativos

**Interfaz:** grupo de pantallas o dispositivos que son capaces de transformar las señales generadas ya sea por programación o un aparato en señales comprensibles por otro.

**Scripts:** código de programación que ejecuta diversas operaciones en un programa de computación

**Css:** sus siglas en ingles son cascading stylesheets lenguaje de diseño gráfico que permite la creacion de páginas HTML amigables para dl usuario.

## ÍNDICE GENERAL

|   |       |
|---|-------|
| CERTIFICACIÓN .....                                 | II    |
| FIRMAS DE RESPONSABILIDAD .....                     | V     |
| DEDICATORIA.....                                    | VI    |
| AGRADECIMIENTO .....                                | VII   |
| ABREVIATURAS .....                                  | VIII  |
| GLOSARIO DE TÉRMINOS .....                          | IX    |
| ÍNDICE GENERAL.....                                 | X     |
| ÍNDICE DE TABLAS.....                               | XIII  |
| ÍNDICE DE FIGURAS .....                             | XIV   |
| INTRODUCCIÓN.....                                   | XVI   |
| RESUMEN.....  | XVIII |
| SUMMARY.....  | XVIII |
| CAPÍTULO 1 .....                                    | 1     |
| 1    MARCO REFERENCIAL.....                         | 1     |
| 1.1    INTRODUCCIÓN .....                           | 2     |
| 1.2    ANTECEDENTES .....                           | 2     |
| 1.3    DEFINICIÓN DEL PROBLEMA .....                | 3     |
| 1.4    JUSTIFICACIÓN .....                          | 4     |
| 1.5    OBJETIVOS .....                              | 4     |
| 1.5.1    Objetivo General .....                     | 4     |
| 1.5.2    Objetivos Específicos .....                | 4     |
| CAPÍTULO II.....                                    | 6     |
| 2    MARCO TEÓRICO .....                            | 6     |
| 2.1    SITIOS WEB .....                             | 7     |
| 2.1.1    Que es un Sitio Web.....                   | 7     |
| 2.1.2    Tipos de Sitios Web .....                  | 7     |
| 2.1.3    Estructura de un Sitio Web.....            | 8     |
| 2.1.4    Características de Sitio Web.....          | 11    |
| 2.1.5    Etapas de la creación de un sitio web..... | 13    |
| 2.1.6    Beneficios.....                            | 17    |
| 2.2    Orientación a Objetos .....                  | 19    |
| 2.2.1    Concepto.....                              | 19    |

|                    |   |    |
|--------------------|---|----|
| 2.2.2              | Estructura de la Programación Orientada a Objetos ..... | 20 |
| 2.3                | PHP .....   | 23 |
| 2.3.1              | Historia .....  | 23 |
| 2.3.2              | Definición.....   | 23 |
| 2.3.3              | ¿Qué necesita para que funcione Php? .....              | 24 |
| 2.3.4              | Características .....                                   | 24 |
| 2.3.5              | Fortalezas y Debilidades .....                          | 25 |
| 2.3.6              | Programación Básica .....                               | 26 |
| 2.3.7              | Constructor .....                                       | 39 |
| 2.3.8              | Destructor .....  | 41 |
| 2.4                | MYSQL .....   | 42 |
| 2.4.1              | Historia .....  | 42 |
| 2.4.2              | Estructura interna de MySQL.....                        | 42 |
| 2.4.3              | Aplicaciones e Interfaces.....                          | 43 |
| 2.4.4              | Características o Ventajas .....                        | 44 |
| 2.4.5              | Debilidades.....  | 45 |
| 2.4.6              | Código Base .....                                       | 45 |
| 2.5                | JavaScript .....  | 52 |
| 2.5.1              | Historia .....  | 53 |
| 2.5.2              | Características de JavaScript.....                      | 53 |
| 2.5.3              | Usos de JavaScript.....                                 | 54 |
| 2.5.4              | ¿Cómo identificar código Javascript? .....              | 54 |
| 2.5.5              | Compatibilidad con navegadores .....                    | 54 |
| 2.6                | CSs3.....   | 55 |
| 2.6.1              | Concepto.....   | 55 |
| 2.6.2              | Características .....                                   | 55 |
| 2.6.3              | Usos de CSS3 .....                                      | 56 |
| CAPÍTULO III ..... |   | 59 |
| 3                  | ANÁLISIS Y DISEÑO.....                                  | 59 |
| 3.1                | RECOLECCIÓN DE LA INFORMACIÓN.....                      | 60 |
| 3.1.1              | Alcance.....  | 60 |
| 3.2                | ANÁLISIS .....  | 60 |
| 3.2.1              | Estudio de factibilidad.....                            | 61 |
| 3.2.2              | Análisis de los requerimientos .....                    | 62 |

|                    |   |    |
|--------------------|---|----|
| 3.2.3              | Definición de Casos de Uso .....                      | 64 |
| 3.3                | DISEÑO.....   | 68 |
| 3.3.1              | Diseño conceptual .....                               | 69 |
| 3.3.2              | Modelo relacional.....                                | 70 |
| 3.3.3              | Diccionario de datos .....                            | 71 |
| 3.3.4              | Diseño de interfaces .....                            | 75 |
| CAPÍTULO IV .....  |   | 79 |
| 4                  | IMPLEMENTACIÓN DEL SITIO WEB.....                     | 79 |
| 4.1                | CONFIGURACIÓN DE LAS HERRAMIENTAS DE DESARROLLO ..... | 80 |
| 4.2                | ARQUITECTURA DEL SISTEMA .....                        | 88 |
| 4.3                | IMPLEMENTACIÓN DEL SISTEMA.....                       | 89 |
| 4.3.1              | Definición de Módulos .....                           | 89 |
| 4.3.2              | Desarrollo de la aplicación implementación .....      | 90 |
| 4.3.3              | Implementación de la aplicación .....                 | 90 |
| 4.4                | PRUEBAS DE LA APLICACIÓN.....                         | 90 |
| 4.5                | CAPACITACIÓN AL PERSONAL.....                         | 92 |
| 4.6                | MANTENIMIENTO DE LA APLICACIÓN .....                  | 92 |
| CAPÍTULO V.....    |   | 93 |
| 5                  | CONCLUSIONES Y RECOMENDACIONES .....                  | 93 |
| 5.1                | Conclusiones.....                                     | 94 |
| 5.2                | Recomendaciones .....                                 | 94 |
| BIBLIOGRAFÍA ..... |   | 95 |
| ANEXOS .....       |   | 97 |

## ÍNDICE DE TABLAS

|   |    |
|---|----|
| Tabla 1. Factibilidad Técnica .....                             | 62 |
| Tabla 2. Diccionario de datos referenciales del sitio web ..... | 71 |
| Tabla 3. Diccionario de datos tabla base .....                  | 72 |
| Tabla 4. Diccionario de datos tabla tipoBien .....              | 72 |
| Tabla 5. Diccionario de datos tabla Gráfico .....               | 72 |
| Tabla 6. Diccionario de datos tabla Bien.....                   | 72 |
| Tabla 7. Diccionario de datos tabla Documentación.....          | 73 |
| Tabla 8. Diccionario de datos tabla CustodioBien .....          | 73 |
| Tabla 9. Diccionario de datos tabla Clase .....                 | 73 |
| Tabla 10. Diccionario de datos tabla Custodio.....              | 74 |
| Tabla 11. Diccionario de datos tablaCargo .....                 | 74 |
| Tabla 12. Diccionario de datos Usuario .....                    | 74 |
| Tabla 13. Pruebas de los requerimientos Funcionales.....        | 91 |
| Tabla 14. Pruebas de los requerimientos no funcionales.....     | 91 |

## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| Figura 1. Estructura de carpetas y archivos .....                                | 9  |
| Figura 2. Estructura de navegación jerárquica .....                              | 10 |
| Figura 3. Estructura de navegación lineal .....                                  | 10 |
| Figura 4. Estructura de navegación lineal .....                                  | 10 |
| Figura 5. Estructura de navegación lineal .....                                  | 11 |
| Figura 6. Esquema del Proceso de Presentación de Pagina WEB Elaborada en PHP ... | 24 |
| Figura 7. Esquema de Petición de una interfaz en PHP .....                       | 28 |
| Figura 8. Estructura Interna de MySQL .....                                      | 42 |
| Figura 9. Modelo de animaciones y transiciones en CSS.....                       | 56 |
| Figura 10. Ejemplo de layout en columnas en CSS .....                            | 56 |
| Figura 11. Ejemplo de gradientes en CSS .....                                    | 57 |
| Figura 12. Relaciones entre elementos en CSS .....                               | 57 |
| Figura 13. Ejemplo de fuentes en CSS .....                                       | 58 |
| Figura 14. Caso de uso página informativa .....                                  | 65 |
| Figura 15. Caso de uso ingresar al sistema.....                                  | 66 |
| Figura 16. Caso de uso ingresar o actualización datos de registro .....          | 67 |
| Figura 17. Caso de uso generar reportes.....                                     | 68 |
| Figura 18. Diseño conceptual del sitio web.....                                  | 69 |
| Figura 19. Modelo relacional del sitio web .....                                 | 70 |
| Figura 20. Pantalla de inicio .....  | 75 |
| Figura 21. Acceso al sistema .....   | 76 |
| Figura 22. Ingreso de datos.....   | 76 |
| Figura 23. Consulta de datos .....   | 77 |
| Figura 24. Actualización de datos .....  | 77 |
| Figura 25. Asignación de los bienes al custodio .....                            | 78 |
| Figura 26. Ejecución de Netbeans .....   | 80 |
| Figura 27. Instalador de Netbeans .....  | 80 |
| Figura 28. Contrato de uso de Netbeans.....                                      | 81 |
| Figura 29. Ubicación de Netbeans en una carpeta.....                             | 81 |
| Figura 30. Resumen de opciones Netbeans .....                                    | 82 |
| Figura 31. Instalación de Netbeans.....  | 82 |
| Figura 32. Netbeans 8.2.....   | 83 |

|  |    |
|--|----|
| Figura 33. Ejecución de XAMPP .....                                | 83 |
| Figura 34. Elección de componentes en XAMPP .....                  | 84 |
| Figura 35. Ubicación en una carpeta el programa XAMPP .....        | 84 |
| Figura 36. Pantalla de confirmación de XAMPP .....                 | 85 |
| Figura 37. Ejecución de XAMPP Control Panel .....                  | 85 |
| Figura 38. Elección de módulos .....                               | 86 |
| Figura 39. Verificación del servidor en modo activo .....          | 86 |
| Figura 40. PhpMyAdmin .....  | 87 |
| Figura 41. Creación de una nueva base de datos .....               | 87 |
| Figura 42. Ingreso de los campos para la base de datos creada..... | 88 |
| Figura 43. Arquitectura del sitio web .....                        | 88 |
| Figura 44. Definición de módulos del sistema web.....              | 90 |

## INTRODUCCIÓN

Los sitios web representan una herramienta tecnológica que ha venido facilitando a las personas, empresas, entidades o establecimientos, en el manejo y difusión de la información a gran escala, de esta manera se han convertido en un soporte de ayuda para el control de las actividades y por su puesto el ahorro de recursos que fomentan el desarrollo de cualquier tipo de organización. El Cuerpo de Bomberos Mocha, perteneciente a la municipalidad con el mismo nombre, de la provincia de Tungurahua, presta sus servicios de emergencias contra incendios, soporte en desastres naturales e incidencias que suscitan en la colectividad de la zona y provincia. Actualmente, cuenta con personal calificado, además de un amplio stock de materiales y equipos para brindar un servicio eficaz y oportuno, sin embargo, todos los ingresos de materiales y equipos que llegan al establecimiento se los realiza de forma manual, lo que ha ocasionado ciertas pérdidas de los bienes que se encuentran en bodega.

Esta falencia se verá cubierta gracias al desarrollo de un sitio web que tiene como principal función la de organizar y administrar correctamente los bienes existentes en el Cuerpo de Bomberos Mocha, permitiendo la generación de reportes, ingresos de materiales y equipos y baja de materiales que se encuentran con falencias.

En vista de la necesidad que posee el cuerpo de bomberos de encontrar un mecanismo que facilite el manejo de los bienes de forma acertada, y por aplicar mis conocimientos en programación web adquirida en la institución, motivó la realización de un sitio web enfocado a la solución de esta problemática.

El objetivo principal de la investigación es “Diseñar e implementar un sitio web para el control de materiales y equipamientos del cuerpo de bomberos del cantón Mocha provincia de Tungurahua desarrollado en PHP orientado a objetos y el gestor de base de datos MYSQL en el período 2018”.

Seguidamente se da a conocer los cuatro capítulos que compone este informe:

El Capítulo I, describe el marco referencial de la investigación que determina los antecedentes de la empresa beneficiaria, la problemática, la justificación de la investigación donde se explica los motivos del presente estudio aportes y beneficios. A



partir de ahí surge el objetivo general conjuntamente con los objetivos específicos para resolver dicho cuestionamiento.

En el Capítulo II, se define el Marco Teórico que representan las bases teóricas consultadas de fuentes bibliográficas que verifican todos los componentes y herramientas necesarias para la realización de un sitio web de control de materiales y equipos.

El Capítulo III, contiene el análisis de la información levantada y el diseño que se llevó a cabo para el desarrollo del sitio web para el cuerpo de Bomberos Mocha.

El Capítulo IV, se refiere al desarrollo práctico del sistema programado en lenguaje de php orientado a objetos y con el gestor de base de datos MySQL, con el que se elaboran las tablas con los materiales y equipos existentes, de ingresos y de los que se dan de baja con lo que se diseña el sitio web en el cuerpo de bomberos .

Por último se detallan las conclusiones a las que se llega con la investigación y las recomendaciones.

## RESUMEN

El Trabajo de Investigación titulado desarrollo e implementación de un sitio web para el control de materiales y equipamientos fue implementado en el Cuerpo de Bomberos del cantón Mocha provincia de Tungurahua en el período 2018. Para el desarrollo de la aplicación se utilizó el lenguaje de programación PHP para crear formularios para el ingreso de datos, interfaces para la interactividad con el usuario, acceso multimedia para mejorar los contenidos, enlaces para llevarle al usuario a otra parte del sitio a través de la arquitectura cliente servidor. El módulo de usuario creó los usuarios administradores del sitio; la sección de compañías crea las bases que son las sedes del Cuerpo de Bomberos; el entorno bienes permite ingresar los datos de todos los bienes que posee el cuerpo de bomberos, como base de datos el gestor de base de datos MySQL que es el almacenamiento físico de la información de todas las tablas que están relacionadas entre sí, como son bienes, tipo de bienes, documentación, clase de documentación, custodios, cargos que contienen la información necesaria para la correcta funcionalidad del sitio web implementado, el cual se programó de acuerdo a las necesidades y requerimientos del cuerpo de bomberos. La implementación de este sitio permitirá el ingreso de datos con respecto a materiales y equipamientos a ser manejado por dos usuarios, el administrador y el invitado quienes serán los encargados del manejo del sistema y del ingreso de la información, para así generar reportes organizados y de una manera sencilla, acción que permitirá al Cuerpo de Bomberos tener un control permanente sobre el control de materiales y equipamiento evitando así el desperdicio de tiempo y materiales.

## **SUMMARY**

The research work entitled development and implementation of a website for the control of materials and equipment was implemented in the fire department of the Mocha canton province of Tungurahua in the period 2018. For the development of the application the PHP programming language was used the management of user sites with the account manager for the user's privacy. of the site; the companies section creates the bases that are the headquarters of the Fire Department; the environment of the goods allows to enter the data of all the assets that the fire brigade owns, such as the MySQL database manager, which is the physical storage of the information of all the tables that are related to each other, such as they are goods, type of goods, documentation, kind of documentation, custodians, loads that contain the necessary information for the correct functionality of the implemented website, which is programmed according to the needs and requirements of the fire department. The implementation of this site allows the entry of data regarding the materials and equipment that are handled by two users, the administrator and the guests that are related to the management of the system and the entry of the information, in order to generate organized and in a simple way, the action that allows the Fire Department to have permanent control over the control of materials and equipment, thus avoiding wasting time and materials.

## **CAPÍTULO 1**

### **1 MARCO REFERENCIAL**

## **1.1 INTRODUCCIÓN**

El siguiente capítulo detalla la problemática actual que posee el Cuerpo de Bomberos del cantón Mocha de la provincia de Tungurahua, con respecto al control de materiales y equipos que posee. Actualmente, el establecimiento no cuenta con un sitio web que procese los datos antes mencionados sino se lo realiza de forma manual, lo que ha conllevado a que existan ciertas deficiencias en el inventario y que la promoción del servicio a la comunidad es limitada.

Finalmente se definirán los objetivos como una guía a seguir para la implementación del sitio web.

## **1.2 ANTECEDENTES**

En vista de la necesidad de instaurar en el Cantón Mocha un organismo de socorro que vele por la ciudadanía las veinte y cuatro horas del día, con un alto espíritu de ayuda social se pensó en la creación del Cuerpo de Bomberos, la iniciativa la toma el señor alcalde el Eco. Cipriano Ocaña Valle, quien realizó todas las gestiones encaminadas a dicho propósito, luego de reunir toda la información necesaria es ingresada la propuesta a la Secretaria Nacional de Gestión de Riesgos a través de la Coordinación Nacional de Cuerpos de Bomberos, organismo encargado de analizar bajo estudios técnicos de factibilidad la aceptación o negativa de la misma.

Después del reporte emitido, el 16 de Julio del 2010, bajo Resolución No. 107 SNGR, es aceptada la petición considerando:

Que, es deber del Estado Ecuatoriano y por ende de la Secretaria Nacional de Gestión de Riesgos, proteger a las personas, la colectividad, y la naturaleza frente a los efectos negativos de los desastres de origen natural o antrópico mediante la prevención ante el riesgo, la mitigación de desastres, la recuperación y mejoramiento de las condiciones sociales, económicas y ambientales, con el objeto de minimizar la condición de vulnerabilidad.

Que, mediante Decreto Ejecutivo 1670 del 14 de abril del 2009, se asigna a la Secretaria nacional de Gestión de Riesgos las competencias, atribuciones, funciones, representaciones, que la Ley de Defensa Contra Incendios establece, para el Ministerio de Bienestar Social, hoy Ministerio de Inclusión Económica y social (MIES).

Que mediante oficios No. 256 y 331 AGMM de 02 de junio y de 07 de julio del 2010, el señor Eco. Cipriano Ocaña Valle, Alcalde del cantón Mocha, Provincia del Tungurahua, solicita la creación de un Cuerpo de Bomberos en el antes mencionado Cantón.

El presente proyecto pretende dotar al cuerpo de bomberos de una herramienta tecnológica conocida como sitio web que facilite la publicación de actividades, datos históricos como misión, visión y generar un control adecuado de los materiales y equipamientos que se posee la Institución.

Un sitio web permite informar a las personas sobre actividades, compartir fotos, organizar los recursos de una Institución, etc. Desde el punto de vista de la utilidad una página web puede cubrir muchas necesidades.

Para la elaboración de esta herramienta web se utiliza el lenguaje de programación PHP Orientada a Objetos (POO), agregada a PHP 5 que hace de la construcción modular y reusable de aplicaciones web mucho más fácil. Con el lanzamiento de PHP 5, los programadores en PHP finalmente tienen el poder de programar en un alto grado de profesionalismo. Así como Java y C#, PHP finalmente tiene una completa infraestructura POO.

### **1.3 DEFINICIÓN DEL PROBLEMA**

El Cuerpo de Bomberos del cantón Mocha en la actualidad no cuenta con un sitio web para el control de sus materiales y equipos, lo que ha provocado ciertas deficiencias sobre sus bienes. Además de desaprovechar todas las ventajas que una herramienta tecnológica otorga sobre todo en la difusión de la información hacia la ciudadanía.

## **1.4 JUSTIFICACIÓN**

El Cuerpo de Bomberos de Mocha es una organización enfocada a brindar servicios de emergencia a la ciudad con el mismo nombre. En la actualidad dentro del establecimiento cuenta con varios equipos y materiales para cumplir a cabalidad con los servicios que oferta, esta tarea se lo realiza de manera manual, lo que ha ocasionado varias falencias en la conservación y buen uso de dichos bienes.

Debido a esta gran necesidad el trabajo respondió al desarrollo de una herramienta tecnológica o sitio web podrá dar a conocer información relevante a las personas sobre todo en las actividades que realizan a diario y la generación de una base de datos que vigile minuciosamente los materiales y equipos que se posee el Cuerpo de Bomberos.

Todo lo anteriormente expuesto permitirá al personal que labora en la Institución prestar un mejor servicio, dando respuesta en el registro de la información, proceso que será realizado en el menor tiempo posible, con la mayor veracidad y de forma eficiente.

El beneficiario directo será sin dudas el Cuerpo de Bomberos Mocha y la población en general, por lo que existe toda la predisposición por parte del Coronel Marcelo Ledesma, autoridad máxima de la institución, haciendo que el proyecto resulte factible en su aplicación.

## **1.5 OBJETIVOS**

### **1.5.1 Objetivo General**

Diseñar e Implementar un sitio web para el control de materiales y equipamientos del cuerpo de bomberos del cantón Mocha provincia de Tungurahua, desarrollado en PHP orientado a objetos y el gestor de base de datos MYSQL en el periodo 2018.

### **1.5.2 Objetivos Específicos**

- Investigar la relación que tienen el lenguaje de programación PHP orientado a objetos y el SGBD MySQL en el desarrollo de los sitios web

- Determinar las necesidades funcionales del Cuerpo de Bomberos Mocha indispensables para el diseño del sitio web de control de materiales y equipamientos del mismo
- Implementar el sitio web para el control de los materiales y equipamientos del Cuerpo de Bomberos del cantón Mocha



**CAPÍTULO II**  
**2 MARCO TEÓRICO**

## 2.1 SITIOS WEB

### 2.1.1 Que es un Sitio Web<sup>1</sup>

Sitio es un espacio o lugar que puede ser virtual o físico que es ocupado por algo. También, es conocido como WEB al grupo de redes que se interconectan mediante un conjunto de protocolos denominado TCP/IP.

Se define así como Sitio Web al espacio virtual en el cual reposan un conjunto de páginas, archivos o bases de datos.

### 2.1.2 Tipos de Sitios Web<sup>2</sup>

Los tipos de sitios web se clasifican de acuerdo a los siguientes parámetros:

#### 2.1.2.1 Visibilidad de contenidos

Se dividen en:

- **Intranets.**- Es un grupo de sitios web que se encuentran dentro de una empresa u organización que solo se puede visualizar dentro de la red privada de las mismas.
- **Extranets.**- Es una intranet que se puede visualizar en la red, pero sigue siendo una red privada con sus lineamientos y sus restricciones pero con la accesibilidad fuera de las instalaciones de la empresa.
- **Sitios web públicos** Como su nombre lo indica, este es un sitio para ser utilizado por usuarios fuera o dentro de una empresa.

#### 2.1.2.2 Actualización de contenidos

Se dividen en:

- **Sitios web estáticos.**- Son sitios que no pueden ser modificados, son aquellos que solo permiten consultar información. Suelen ser utilizados por instituciones para dar a conocer información básica.

---

<sup>1</sup> <https://definicion.de/sitio-web/>

<sup>2</sup> <http://www.estudioseijo.com/noticias/tipos-de-sitios-web.htm>

- **Sitios web dinámicos.-** Son aquellas que su contenido puede variar a modo que su información siempre está actualizada, ya sea por su programador o por usuarios externos.
- **Sitios web que combinan ambos tipos.-** Este tipo de sitios son aquellos q son utilizados por empresas o instituciones que combina los 2 tipos anteriores.

### 2.1.2.3 Actividad o fin principal

Según la actividad los sitios web pueden ser de tipo:

- **Comercial/publicitaria.-** Su objetivo, es ayudar a las empresas a difundir campañas, productos, servicios o noticias.
- **Venta.-**Estos sitios ayudan a las empresas a vender sus productos.
- **Sociales.-** Sitios con blogs, redes como facebook o myspace, etc. Que ayudan a la conexión de usuarios en todo el mundo.
- **Informativos.-** Son aquellos como las redes sociales o páginas de noticieros que sirven para dar a conocer información que varía cada día.
- **Buscadores.-** Sitios que ayudan a la investigación o búsqueda de información.

### 2.1.3 Estructura de un Sitio Web<sup>3</sup>

Para visualizar una buena estructura en la parte lógica de un Sitio Web se debe considerar las siguientes características:

#### 2.1.3.1 Estructura de carpetas y archivos

Para la creación y diseño de un Sitio Web se debe iniciar primero con la estructuración de sus carpetas y archivos. Tomando en cuenta que cuando se tiene un gran número de ficheros resulta más fácil ubicarlos en las carpetas ya anteriormente creadas esto ayuda a la localización y edición.

La variedad de archivo (audios, vídeos, hojas de estilo, imágenes, etc) y paginas HTML son guardadas en el directorio, carpetas o subcarpetas que pertenecen a el Sitio Web.

---

<sup>3</sup> <http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>

Las carpetas pueden estar divididas en:

- **Audios:** archivos de audio mp3.
- **CSS:** hojas de estilo css.
- **Descargas:** archivos zip, exe, etc que se ofrecen para descarga.
- **Images:** imágenes jpg, gif o png.
- **PDF:** documentos pdf.
- **Scripts:** archivos js con código javascript reutilizable.
- **Swfs:** archivos con animaciones flash (\*.swf).
- **Vídeos:** ficheros en distintos formatos de vídeo: \*.wmv, \*.flv

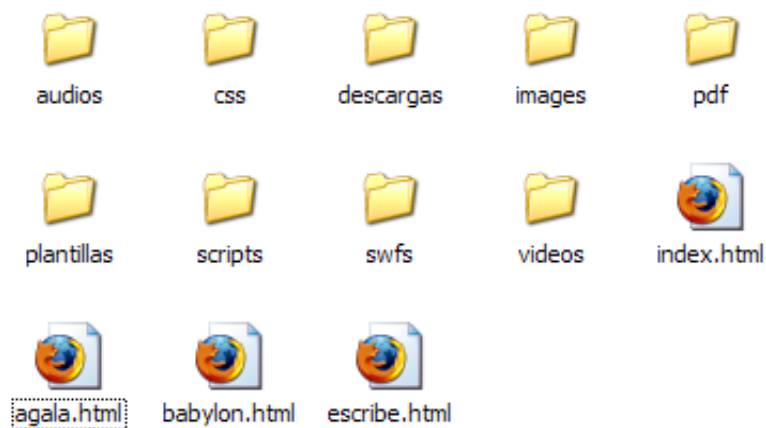


Figura 1. Estructura de carpetas y archivos

Fuente:<http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>

### 2.1.3.2 Estructura de navegación

Esta etapa se define por la experiencia del desarrollador, el visitante puede saltar de una página utilizando los hipervínculos creado para la navegación.

Esta estructura se define también en varios tipos descritos a continuación:

- **Jerárquica.-** Se hace relación a la estructura de un árbol, en este caso se puede empezar en una página(o ramificación) e ir saltando a los siguientes niveles y regresar a las anteriores, no se aconseja que tenga más de 4 niveles.

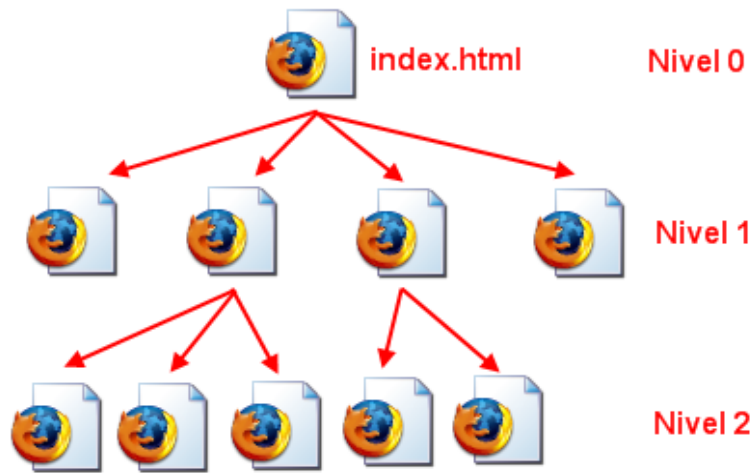


Figura 2. Estructura de navegación jerárquica

Fuente: <http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>

- **Lineal.-** Esta estructura es similar a un libro, requiere de hipervínculos lineales que significan que solo podrán ir de adelante hacia atrás y no existen niveles.



Figura 3. Estructura de navegación lineal

Fuente: <http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>

- **Lineal con jerarquía.-** Es la mezcla de las 2 estructuras anteriores, aprovechando las ventajas de estas combinándolas para mejorar la navegación para el usuario.



Figura 4. Estructura de navegación lineal

Fuente: <http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>

- **Red.-** A partir de la página índice o principal se puede navegar a otra u otras sin ningún orden aparente. Es una estructura más libre pero no es aconsejable cuando el número de páginas es elevado porque desorienta al usuario al no saber dónde está ni disponer de recursos para ir donde desea.

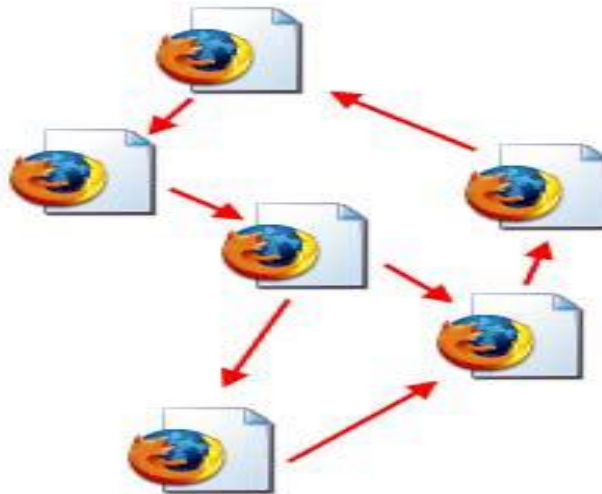


Figura 5. Estructura de navegación lineal

Fuente: <http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>

## 2.1.4 Características de Sitio Web

Gracias al Internet la vida cotidiana (pagar, comprar o relacionarnos), las comunicaciones, la distancia entre personas, se ha acortado, a esto se suma que el desarrollo móvil permite la conectividad en cualquier lugar.

Lo que conlleva que existen varios Sitios Web se conviertan en peligrosos, para lo cual los desarrolladores han dado varias pautas o características para crear sitios web seguros y confiables como son:

### 2.1.4.1 Buena ortografía

Al igual que en un libro, un sitio web de calidad permite la corrección de faltas ortográficas lo que la conllevaría a una conclusión de sitio de alto riesgo o de baja calidad.

Al igual que un sitio web que ha sido traducido se debe tomar muy en cuenta que no sea traducido por un traductor de máquina.

#### **2.1.4.2 No hay enlaces rotos**

En la navegabilidad no debe existir enlaces rotos, esto da la impresión de un sitio de baja calidad y de baja confiabilidad, para lo cual una de las soluciones de los desarrolladores fue redireccionar a 301 que no es más el lector a la página que tenga la información correcta y volver al inicio para seguir navegando.

#### **2.1.4.3 Diseño responsive**

En la actualidad gracias a la nueva tecnología, las personas, abren sus aplicaciones o sitios web en sus dispositivos móviles, esto conlleva que si una aplicación no se puede visualizar en el dispositivo móvil, esto lleva a pensar al usuario que este sitio no es confiable o que está mal diseñado.

#### **2.1.4.4 Forma de contacto**

Una característica primordial es poder tener contacto fácilmente, tanto con el desarrollador, como con la empresa que está dando los servicios, si un usuario no encuentra un contacto limitará la veracidad de este sitio, al contrario la facilidad el contacto llevaría a la conclusión de que el sitio sea seguro y confiable.

#### **2.1.4.5 Sitio web seguro<sup>4</sup>**

La confiabilidad en el desarrollo de un sitio web es lo principal porque han existido varios casos de robo de identidad o uso indebido de la información de los usuarios. ¿Pero en realidad existe elementos que permitan dar seguridad?, para esta pregunta existen certificados como:

---

<sup>4</sup> <https://colombiadigital.net/actualidad/articulos-informativos/item/9461-cuales-son-las-caracteristicas-de-un-sitio-web-confiable.html>

- **Certificados SSL.-** Ayuda para que los usuarios la certeza de ingresar sus datos personales, esto lo logra con los protocolos de seguridad SSL, mediante la cual los datos viajan en forma cifrada o encriptada, para comprobar que un Sitio Web tenga este protocolo solo se debe fijar si el http cuenta con una “s” es decir https
- **Certificado digital.-** Este certificado es electrónico el mismo que confirma al usuario la confiabilidad del sitio, esto se da al asociar un archivo generado por la entidad que entrega el servicio, confirmando la identidad del usuario lo que ayuda a crear una confiabilidad del sitio.

## 2.1.5 Etapas de la creación de un sitio web<sup>5</sup>

### 2.1.5.1 Primera Etapa: Planificar

- **El Equipo**

El Hardware para el desarrollo de un sitio web es una parte importante, tomando en cuenta que este es el repositorio para la concepción, construcción, evaluación y mantenimiento del sitio.

Otra parte es el grupo de los desarrolladores, que representan aquellos responsables de la estructura no visible del sitio. Este grupo o equipo debe estar compuesto por técnicos en varias disciplinas, con un líder conocido como Webmaster ya que la institución está poniendo toda su confianza en esta persona para que el proyecto se lleve a cabo con éxito.

Al igual que el equipo físico, el equipo humano no se puede tomar a la ligera tomando en cuenta que en dichos equipos está el éxito del proyecto, para esto requiere un proceso de consolidación, acuerdos y buscar la mejor propuesta, el grupo debe estar en constante comunicación, colaboración y conocimientos.

---

<sup>5</sup> <http://www.redalyc.org/html/161/16114408/>



- **Reunir información**

Una vez determinado el equipo físico y humano se debe definir qué tipo de proyecto se va a realizar, para ello se requiere de reuniones entre el equipo humano y el cliente, el cual, debe dar toda la información que requiere el desarrollador para la puesta en marcha del proyecto. Este segundo ayuda a determinar la parte teórica como técnica.

- **Determinar la misión y los objetivos del sitio**

Al tener la información bien definida de lo que el usuario desea, es importante establecer la misión del sitio, la misma que debe estar por escrito darla a conocer al resto del equipo para que fluyan ideas y perfeccionar la misión.

Igualmente se debe cumplir con los objetivos fijados por el usuario, considerando que este puede cambiar uno o todos los objetivos de acuerdo a sus necesidades.

- **Determinar la/s audiencia/s destinadas y posibles**

Para el desarrollador la etapa de adaptar la orientación de sitio Web a las necesidades no solo del usuario, sino también, a los usuarios reales, al igual que a los posibles usuarios potenciales. La utilidad en la información de un sitio web debe enfocarse a ser clara precisa y de utilidad.

Los usuarios reales o fijos en este caso buscan un sitio con productos específicos por lo tanto este tipo de estructuras son más sencillas.

- **Establecer un plan de trabajo**

Una vez definido todos los pasos anteriores, se establecerá al final el plan de trabajo que permita un camino tentativo que ayude al equipo a tomar acciones futuras y el camino tentativo para el desarrollo del sitio web.

### **2.1.5.2 Segunda Etapa: Concebir**

- **Establecer criterios de calidad**

El equipo de desarrollo establece contenidos y la usabilidad del sitio, para esto se debe contar con la aprobación previa de la empresa para la cual se entrega un listado de criterios, como políticas que sustenten la calidad a ser definido para el sitio.

- **Determinar los contenidos**

Es indispensable crear un plan, de contenidos, con información básica, lo principal en esta etapa es definir textos planos. Estos contenidos serán determinados para la aplicación, de acuerdo a las exigencias de la empresa, estos textos deben ser cortos, simples y no complejos, si son extensos estos deben ser opcionales y claros para dar al usuario un buen acceso para continuar navegando.

- **Determinar la estructura de los contenidos**

Al determinar los contenidos no basta solo que la información exista, esta debe ser secuencial y suministrar la adecuada navegación. Por ello, es primordial dividir en áreas, las mismas, deben tener una estructura jerárquica, permitiendo al usuario una mejor navegabilidad.

- **Determinar el formato gráfico**

Una vez definido la estructura de contenidos se crea en forma gráfica la estructura del sitio Web lo que permitirá al cliente una mejor visualización de sus requerimientos. Este esquema debe estar centrado en el usuario final/potencial y su accesibilidad.

Una vez presentado este esquema y al ser aceptado beneficiará al desarrollo como guía en la primera versión del sitio. El equipo de trabajo establecerá la forma más apropiada de presentar la estructura acordada. Se trata de materializar el ordinograma concebido en borrador a través de varias formas posibles de organización.

- **Creación de documentos de trabajo interno**

Las premisas útiles, pautas y políticas generales son definidas por el equipo, que establece, a lo largo del desarrollo su gestión con el marco institucional. También se puede crear documentos en el sitio web de manera restringida llamadas también “páginas silenciosas”.

### **2.1.5.3 Tercera Etapa: Construir, visualizar y testear**

- **Construyendo y evaluando**

En la primera y segunda etapa se definen los requerimientos, HW, y necesidades para la creación del sitio web. Posterior a ello, se describe la función principal referente al diseño de la información que consiste en minimizar la complejidad de los contenidos y permite un mejor acceso.

- **La página de acceso o principal**

La página principal es la puerta de entrada y la presentación del sitio la misma debe ser sencilla completa y fácil de navegar.

- **Decisiones de Navegación**

Al hablar de navegación, se debe seleccionar la estructuración y el diseño más adecuado, por esto un sitio web bien diseñado deber prever las necesidades de sus usuarios y también velar porque éstos encuentren la información que necesitan con el mínimo de esfuerzo.

- **Identidad Visual Corporativa**

Un sitio Web debe tener una identidad visual que permita identificarlo y asociarlo, no solo a partir de un logotipo o del uso de determinados colores, es deseable: la utilización de iconos, siempre en los mismos lugares.

## **2.1.6 Beneficios<sup>6</sup>**

### **2.1.6.1 La gente podrá buscarte en línea**

Para muchas personas, no existe, si no lo encuentran online. El no contar con un sitio web es como no aparecer en la guía telefónica. Ahora se vive en la “Era de la Información” y así como contar con un teléfono, el tener un sitio web es crucial para el éxito de cualquier negocio serio.

### **2.1.6.2 Abierto las 24 horas del día**

Cuando se cuenta con un sitio web, clientes nuevos y actuales pueden encontrar información acerca del negocio y las ventas. Incluso pueden realizar compras online. Esto da la capacidad de ganar dinero de día y de noche, 7 días a la semana.

### **2.1.6.3 Obtener credibilidad instantánea**

Tener un sitio web bien diseñado permite establecer una imagen creíble y profesional al instante para el negocio.

### **2.1.6.4 Contar con un canal de marketing económico**

Los Sitios Web, ofrecen una forma de promoción con bajo costo, mostrando productos y servicios a un público local, nacional o internacional. El costo es mínimo en comparación con el alto costo de los medios de comunicación y la publicidad tradicional.

### **2.1.6.5 Incorporación en nichos de mercado rentables**

Es relativamente bajo el costo de funcionamiento de un sitio web permitiendo a los emprendedores y propietarios de pequeñas empresas, centrar sus esfuerzos de marketing

---

<sup>6</sup> <https://www.infopresario.com/10-beneficios-tener-un-sitio-web/>

en pequeños grupos, altamente selectivos con clientes potenciales que quieren productos y servicios altamente especializados.

#### **2.1.6.6 Información clave del comportamiento del cliente**

Los análisis en la Internet permiten a los propietarios de sitios web rastrear el comportamiento online de todos los que visitan su sitio. Este tipo de información puede ayudar a una empresa a aumentar en gran medida el éxito de sus campañas de marketing online.

#### **2.1.6.7 Conocer las preferencias de los clientes**

Los sitios web y los blogs permiten a emprendedores interactuar con los clientes actuales y potenciales. Estas interacciones virtuales proporcionan información valiosa que pueden ayudar a satisfacer los gustos y preferencias de los clientes. Es decir, que la comprensión del cliente beneficiará en la generación y aumento de ventas.

#### **2.1.6.8 Oportunidades para generar ingresos pasivos**

Con un sitio web, se puede crear una tienda virtual o página de venta que permite a los clientes realizar pedidos y pagar a través de Internet. Todo esto puede suceder de forma automática y sin ninguna participación directa. Por ello, el proceso de intercambiar el tiempo y trabajo duro por dinero, será cosa del pasado. El sitio web se combina con sistemas de comercio electrónico. Es así, que se puede generar flujos potenciales de ingresos pasivos. Todo en piloto automático.

#### **2.1.6.9 Existen ventajas competitivas**

Si la competencia directa es con otra empresa que tiene un sitio web, tienen una clara ventaja; especialmente si son buenos en el marketing de su sitio web a través de la publicidad online, SEO y social media.

#### **2.1.6.10 Negociación mundial**

Muchos propietarios de negocios se limitan a una pequeña área geográfica. No tiene por qué ser así. Con un sitio web y una presencia en línea, el acceso será inmediato a un nuevo mundo de lucrativas oportunidades comerciales. En otras palabras, se podrán hacer negocios con cualquier cliente, en cualquier país del mundo.

## **2.2 ORIENTACIÓN A OBJETOS<sup>7</sup>**

### **2.2.1 Concepto**

Se define como objeto a una serie de sentencias y datos abstractos que son creados como métodos para trabajar sobre el mismo. Los métodos y datos creados están estrechamente relacionados, dado que los datos no pueden ser utilizados por métodos no adecuados, al igual que los métodos no pueden tener datos no apropiados.

Para entender mejor de que trata la orientación a objetos, se describen varios conceptos a continuación:

#### **2.2.1.1 Encapsulación**

Al crear los objetos este concepto se asegura que el control de acceso sea selectivo tanto para los miembros como para los métodos.

#### **2.2.1.2 Herencia**

Se define como herencia, al método que se denomina padre, el cual, permite la reutilización del código. Además, permite a sus métodos hijos heredar todas sus clases.

**Ejemplo:** Pilas y Colas.

#### **2.2.1.3 Polimorfismo**

---

<sup>7</sup> <https://compu2poo.wordpress.com/2013/11/29/programacion-orientada-a-objetos-poo-definicion-origen-y-caracteristicas/>

El polimorfismo permite tratar a objetos de distintas clases de igual manera, el único requisito es que dichos objetos respondan al objeto, esto permite ahorrar código y realizar los métodos de forma más simple.

#### **2.2.1.4 Constructores y destructores**

Los métodos pueden o no tener definido un constructor o destructor, los mismos que se ejecutan automáticamente, definiendo estos dos conceptos. El constructor es el que se ejecuta cuando el método es creado, y el destructor se establece cuando el objeto queda fuera del ámbito de creación.

### **2.2.2 Estructura de la Programación Orientada a Objetos<sup>8</sup>**

La estructura de una POO se basa en 3 componentes que trabajan independientemente cada uno:

#### **2.2.2.1 Relaciones**

Son los enlaces de los objetos que permite la relación entre objetos formando parte de una misma organización, esta relación admite que el objeto se inserte esencialmente con punteros a otros objetos. Existen 2 tipos de relación:

- **Relaciones jerárquicas.**- Se define como Jerarquía a todo aquello que tiene niveles en donde el objeto padre es aquel que se encuentra encima del segundo y por lo tanto el siguiente es hijo del primero (padre). Dentro de este tipo de relación existe también la organización jerárquica simple que se define como que un hijo que tenga un solo padre y la relación jerárquica compleja se trata de que un hijo que cuenta con varios padres.
- **Relaciones semánticas.**- Referente a las propiedades y consecuencias de dichos objetos, solo obedecen a su significado y no a su posición como es la relación jerárquica.

---

<sup>8</sup> [http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Prog\\_Obj01.html](http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Prog_Obj01.html)

### 2.2.2.2 Propiedades

Las propiedades de un objeto en POO corresponden a las “variables” de la programación estructurada, las mismas que tendrán a su vez uno o varios valores. Este componente distingue un objeto determinado de los restantes que forman parte de la misma organización y tienen valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Una de las diferencias entre "variables", y las propiedades que se heredan pueden ser:

- **Propiedades propias.**- Son aquellas que son propias del objeto definido.
- **Propiedades heredadas.**- Se define así a los objetos que heredan propiedades, esto quiere decir un hijo puede heredar las características de un padre o abuelo.

### 2.2.2.3 Métodos

Los métodos son programas, que tienen parámetros o argumentos, los mismos que son heredados por varios objetos, estos pueden tener dos maneras de disponer los métodos como son:

- **Métodos propios.**- Están incluidos dentro de la cápsula del objeto.
- **Métodos heredados.**- Son aquellos que están definidos en un objeto diferente, ya sea padre o abuelo.

### 2.2.2.4 Beneficios que se obtienen del desarrollo con POO<sup>9</sup>

Existe una variedad de beneficios de la POO, pero las más importantes suelen ser:

- **Reusabilidad.** Al diseñar las clases adecuadamente, estas pueden ser usadas por uno o más proyectos.
- **Mantenibilidad.** La POO tiene como característica principal la implementación de la ocultación de detalles de implementación, y dejando visibles detalles relevantes.

---

<sup>9</sup> [http://www.ciberaula.com/articulo/ventajas\\_poo](http://www.ciberaula.com/articulo/ventajas_poo)



- **Modificabilidad.** Es un beneficio muy especial e importante porque ayuda al desarrollador a añadir, suprimir o modificar objetos de una forma más sencilla.
- **Fiabilidad.** Cuando se trata de un objeto en partes pequeñas, ayuda a aislar posibles errores, y probar el método u objeto independientemente. Además, de manera óptima reduce el tiempo de desarrollo.

#### 2.2.2.5 Problemas derivados de la utilización de POO

A pesar que un sistema POO puede ser un paraíso virtual, existen también problemas tales como:

- **Curvas de aprendizaje largas.** Ya que la comunicación entre los objetos debe realizarse mediante mensajes, la POO no está escrito de esta manera, es por eso que los desarrolladores deben capacitarse para poder usarlo.
- **Dependencia del lenguaje.** A pesar de que el sistema orientado a objetos, en la práctica tiene muchos beneficios, en la actualidad existen varios lenguajes de orientación a objetos que compiten por dominar el mercado, pero la desventaja principal es el cambio de lenguaje para el desarrollo esto conlleva a una tarea para nada sencilla.
- **Determinación de las clases.** Una clase es más que un arte, pero a pesar de esto definirla es importante, si bien hay muchas clases predefinidas también se debe crear clases específicas. Las mismas, luego de 6 meses ó 1 año, el desarrollador debe crear o volver a construir otras clases afectando esto en la planificación original de proyecto.
- **Performance.** En la actualidad un diseño que aplica orientación a objetos, que no cuenta con la performance, no es comerciable, siendo que en este sistema todo objeto y toda interacción es a través de mensajes, y esto afecta a él o la performance.

## 2.3 PHP

### 2.3.1 Historia<sup>10</sup>

PHP tiene sus inicios en el año de 1994 su creador es Rasmus Lerdorf, fue desarrollado en C como un CGI, en el principio lo denominaron como Personal Home Page Tools con el paso del tiempo varias personas solicitaron se les permitiera utilizar programas en sus páginas.

Al ver la aceptación de la primera versión de PHP, Rasmus creó un sistema para procesar formularios con el nombre de FI (Form Interpreter), como consecuencia a estas dos creaciones se formó la primera versión PHP/FI. Al incluir nuevas funcionalidades a mediados del '97 se brindaron soportes a nuevos protocolos de Internet así se creó la versión de PHP 3.

En la versión PHP4 se añadió el motor Zend, que fue desarrollado para cubrir necesidades y solucionar algunos inconvenientes de las versiones anteriores.

### 2.3.2 Definición<sup>11</sup>

PHP es un lenguaje de código abierto, del lado del servidor e incorporado en HTML que es compatible con todos los principales servidores web (especialmente Apache). PHP le permite insertar fragmentos de código en páginas HTML normales, el cual se interpreta como páginas útiles hasta usuarios. PHP también sirve como un lenguaje "pegamento", por lo que es fácil conectar sus páginas web a bases de datos del lado del servidor.

PHP es un lenguaje de creación de páginas WEB dinámicas, el código de PHP es invisible para el usuario, definiendo así la interacción, desarrollo, aplicación de páginas creadas y transformadas para que el usuario solo pueda visualizar imágenes, multimedia y formatos que permitan interactuar añadiendo o descargando información.

---

<sup>10</sup>

<http://educagratis.cl/moodle/mod/resource/view.php?id=4105&usg=AOvVaw2V9ryQz7Xw2ijt2jvVZOKD>

<sup>11</sup> [https://www.amazon.com/PHP5-MySQL-Bible-Tim-Converse/dp/0764557467&usg=AOvVaw3jPzqnw\\_eipJF\\_zU5\\_wTIy](https://www.amazon.com/PHP5-MySQL-Bible-Tim-Converse/dp/0764557467&usg=AOvVaw3jPzqnw_eipJF_zU5_wTIy)

### 2.3.3 ¿Qué necesita para que funcione Php?

- Versión compilada de PHP (<http://www.php.net>).
- Un servidor web (Apache, PWS, IIS, Etc.).
- Para el manejo de base de datos se recomienda MySQL Server (<http://www.mysql.com>).



Figura 6. Esquema del Proceso de Presentación de Pagina WEB Elaborada en PHP  
Fuente:<http://educagratis.cl/moodle/mod/resource/view.php?id=4105&usg=AOvVaw2V9ryQz7Xw2ijt2jvVZOKD>

### 2.3.4 Características<sup>12</sup>

El lenguaje PHP presenta cuatro grandes características:

- **Velocidad:** PHP no requiere de grandes recursos tanto Hardware como Software esto permite que sea rápido para su ejecución, por lo tanto, esto no genera retrasos. PHP se integra muy bien junto a otras aplicaciones, especialmente bajo ambientes Unix.
- **Estabilidad:** Al utilizar su propio sistema de administración de recursos PHP se convierte en un sistema robusto y estable que posee un sofisticado método de manejo de variables.

---

<sup>12</sup> <http://redgrafica.com/El-lenguaje-de-programacion-PHP>

- **Seguridad:** PHP maneja distintos niveles de seguridad, estos pueden ser configurados desde el archivo **.ini**
- **Simplicidad:** Para poder programar en PHP no es necesario ser expertos o tener experiencia, los usuarios que conocen o tienen experiencia en C y C++ podrán utilizar rápidamente. También dispone de una amplia gama de librerías, y gracias a su constante actualización existe una gran variedad de documentación en la WEB.

### 2.3.5 Fortalezas y Debilidades

Tabla 1. Fortalezas y debilidades de PHP

| Descripción             | Fortaleza  | Debilidad   |
|-------------------------|--|---|
| Código Spaghetti        | PHP es muy flexible en general, no hay menos de 30 maneras de realizar la misma tarea.   | Cuando las personas les gusta tener su propia forma de programar es difícil, para otros continúen su labor. Esto podría dar como resultado montón de código espagueti y basura.                         |
| Seguridad               | PHP es un sistema que puede ser usado para cualquier cosa, este actualmente funciona con cualquier sistema operativo, esto permite que los desarrolladores aporten y alimenten aún más el lenguaje | PHP al ser un lenguaje de fácil aprendizaje implica que hay aplicaciones terriblemente inseguras. Esto también implica que las personas no desarrolladoras no conozcan de la seguridad de este sistema. |
| Baja barrera de entrada | Cualquier persona con nociones de programación puede programar en PHP  | Los desarrolladores inexpertos, pueden mal utilizar los recursos del sistema, provocando una aplicación de carga lenta y pesada.  |

|                              |  |  |
|------------------------------|--|--|
| Flexible, pero “atolondrado” | PHP tiene la capacidad de adaptarse ya que es sencillo obtener y conseguir lo que el desarrollador requiere. Sin la necesidad de configurar varios recursos previamente. | Desafortunadamente, esta también es una de sus mayores limitaciones. PHP, mientras es un gran lenguaje de desarrollo web, todavía tiene un desarrollo poco confiable.                    |
| Copy-Paste Hacking           | Existe una gran cantidad de código free para cualquier tipo de problemas que el desarrollador requiera.  | Por existir una gran cantidad de código existente que está disponible para crear una variedad de programas. Esto llena a nuestro servidor a contener una gran cantidad de código basura. |

Fuente:<http://www.baluart.net/articulo/ocho-expertos-comentan-los-pros-y-contras-de-programar-con-php>

### 2.3.6 Programación Básica<sup>13</sup>

#### 2.3.6.1 Configuración del entorno de trabajo

- **Configuración del servidor**

Los pasos para configurar un servidor web con soporte para PHP depende del sistema Operativo que se desea utilizar.

- **Trabajo remoto**

De ser necesario y si se desea se puede trabajar con la máquina. Para lo cual, se debe tener y solicitar la URL correspondiente a la maquina destino.

- **Entorno de trabajo**

---

<sup>13</sup> [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)

Como entorno de trabajo se toma en consideración, el entorno con el acceso a funciones como el coloreado de sintaxis PHP, entorno gráfico, gestión de proyectos, pre visualización de resultado, sincronización con de directorios remotos, etc.

### 2.3.6.2 Inclusión de Código PHP en una Página HTML

PHP a pesar de ser una herramienta de libre código, siempre requiere la identificación de una estructura donde el código pueda visualizarse, al incluir código PHP se deberá colocarlo dentro de la etiqueta:

#### Sintaxis:

|                             |  |
|-----------------------------|--|
| <b>Etiqueta de apertura</b> | <code>&lt;?php,</code><br><br>Código de programación |
| <b>Etiqueta de cierre</b>   | <code>?&gt;.</code>                                  |

#### Ejemplo:

```
<?php
include 'include/objElemMenu.php';
class objPagina{
protected $estilo = " ;
protected $direstilo = " ;
protected $script = " ;
protected $dirsript = " ;
protected $titulopagina = " ;
public function __construct() {
    $this->estilo = 'estilos.css' ;
    $this->direstilo = 'css' ;
    $this->script='scripts.js';
    $this->dirsript='scripts';
    $this->titulopagina = 'Sitio Web del Cuerpo de Bomberos Mocha' ;
}
?>
```

Cuando las etiquetas de apertura y cierre estén colocadas, se debe proceder con el código de programación, la misma que si el servidor web se ha configurado correctamente, este detectará dicho código y se presentará al cliente la página WEB (el diseño, y no en código fuente).

En estos casos el esquema de una petición sería como el siguiente ejemplo:

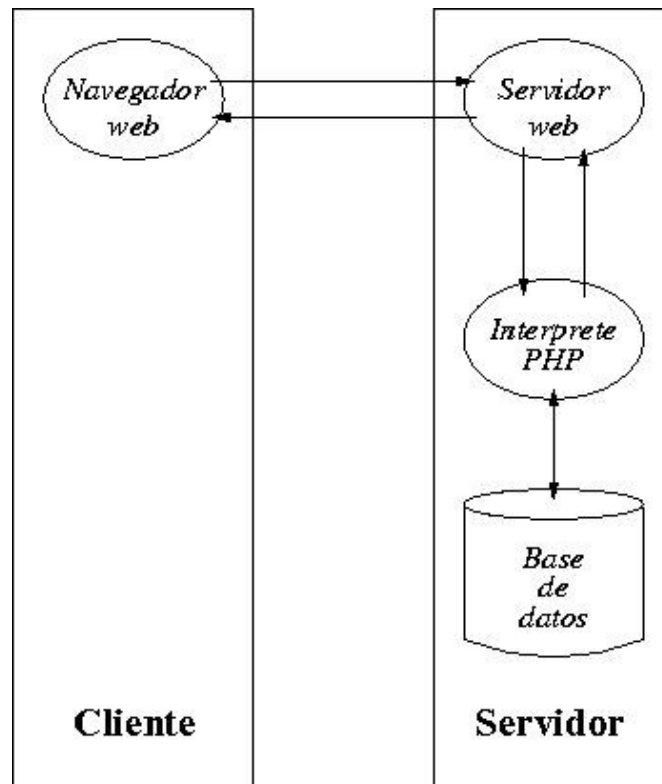


Figura 7. Esquema de Petición de una interfaz en PHP

Fuente: [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)

### 2.3.6.3 Separación de instrucciones

Las instrucciones se separan igual en cualquier lenguaje de programación, terminando cada sentencia con un punto y coma.

**Sintaxis:**

```
<?php
    echo "Esto es una prueba";

?>
```

**Ejemplo:**

```
<?php
session_start();
$inicioSesion=0;
if(!empty($_SESSION['nick'])) {
    $inicioSesion = 1;
```

```

    }
    /* Si no hay una sesión creada, redireccionar al login. */
    if(empty($_SESSION['nick'])) {
        header('Location: login.php');
    }
    include 'include/objUsuario.php';
    include 'include/objConexion.php';
    include 'include/objPagina.php';
    include 'include/funciones.php';
    //$idBien = $_GET['idBien'];
    $pagina= new objPagina();
    echo $pagina->gethead();
    echo $pagina->getbody_ini();
    echo $pagina->getmenu($inicioSesion);
    if (isset($_GET['resultado']))
    {
        $resultado = $_GET['resultado'];
        if ($resultado==1)
        {
            echo '<script type = "text/javascript"> alert("El registro ha sido actualizado
correctamente");</script>';
        }
        else
        {
            echo '<script type = "text/javascript">alert("No se pudo actualizar el
registro");</script>';
        }
    }
    $pdo = objConexion::connect();
    $nombreTipo="";
    $graficoID="";
    $sql = "SELECT * FROM Bien where idBien = '".$_GET['idBien']."'";
    foreach ($pdo->query($sql) as $row) {

```



```

        $bien = new
objBien($row['idbien'],$row['codigoBien'],$row['nombreBien'],$row['cantidadBien'],$r
ow['fechaIngreso'],$row['estadoBien'],

        $row['serieBien'],$row['marcaBien'],$row['modeloBien'],$row['colorBien'],$row
[idBase'],$row['idTipoBien'],$row['idTipoBien']);
        $graficoID = $row['idGrafico'];
    }
objConexion::disconnect();
?>

```

#### 2.3.6.4 Comentarios en PHP

PHP soporta comentarios tipo 'C', 'C++' y Shell de Unix. Por ejemplo:

##### Sintaxis

**<?php**

```
    echo "Prueba"; // Comentario tipo C++ para una línea
```

**?>**

**<?php**

```
    /*
```

```
    Esto es un comentario multilínea otra línea más de comentario
```

```
    */
```

```
    echo "Esto es aún otra prueba";
```

**?>**

##### Ejemplo

**<?php**

```
//$colores = array('Amarillo', 'Azul', 'Rojo', 'Verde', 'Negro', 'Blanco');
```

```
/**
```

```
function daListaColores($parBien)
```

```
{
```

```
    $listaColores =
```

```
    array('Blanco','Amarillo','Azul','Rojo','Verde','Gris','Dorado','Anaranjado','Rosado','Caf
&eacute;','Morado');
```

```
    $cantColores=11;
```

```
    for ($i = 0; $i < $cantColores-1; $i++) {
```

```

    echo "<option value='".$listaColores[$i]."' ";
    if ($parBien->getColor()==$listaColores[$i])
    echo "selected";
    echo ">".$listaColores[$i]."</option>>";
  }
}
*/
?>

```

### 2.3.6.5 Variables

Se define como variable a un elemento que atribuye un valor o una determinada información la cual puede variar según trascorra el programa. Las variables pueden ser consideradas como la base de la programación.

En PHP las variables se representan con un signo de pesos seguido por el nombre de la variable. El nombre de la variable es sensible a minúsculas y mayúsculas.

#### Sintaxis

**<?php**

```

    $var = "var1";
    $Var = "var2";
    echo "$var, $Var";

```

**?>**

#### Ejemplo

```

function reemplazaCaracteres($cadena,$caracter,$porCaracter)
{
    $cadenaTemporal = str_replace($caracter,$porCaracter,$cadena);
    return $cadenaTemporal;
}

```

### 2.3.6.6 Tipos de datos

Tabla 2. Sintaxis de tipos de datos

| Tipo de datos  |  | Sintaxis  |
|----------------|--|---|
| <b>Entero</b>  | <p>Se definen o se especifican con la sintaxis:</p> <p>Dentro de este tipo se encuentra también:</p> <ul style="list-style-type: none"> <li>– Enteros flotantes o doblé</li> </ul> | <pre>\$a = 1234; # número decimal \$a = -123; # un número negativo \$a = 0123; # número octal               (equivalente al 83 decimal) \$a = 0x12; # número hexadecimal               (equivalente al 18 decimal)  \$a = 1.234; \$a = 1.2e3;</pre> |
| <b>Cadenas</b> | <p>Existe en PHP una variedad de funciones para trabajar con cadenas.</p> <p>Para las cadenas PHP interpreta diferente las comillas simples y dobles.</p>                          | <pre>\$a = "Hola"; \$b = \$a . "Mundo"; // Ahora \$b contiene "Hola Mundo"  \$a = "Mundo"; echo = 'Hola \$a'; //Esto escribirá "Hola \$a" echo = "Hola \$a"; //Esto escribirá "Hola Mundo"</pre>  |

Fuente: [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)

### 2.3.6.7 Caracteres protegidos

`\n` Nueva línea

`\r` Retorno de carro

`\t` Tabulación horizontal

`\\` Barra invertida

`\$` Signo del dólar

`\"` Comillas dobles

`\[0-7]{1,3}` La secuencia de caracteres que coincida con la expresión regular es un carácter en notación octal

\x[0-9A-Fa La secuencia de caracteres que coincida con la expresión regular es f]{1,2} un carácter en notación hexadecimal

### 2.3.6.8 Operadores de comparación

$\$a < \$b$   $\$a$  menor que  $\$b$

$\$a > \$b$   $\$a$  mayor que  $\$b$

$\$a <= \$b$   $\$a$  menor o igual que  $\$b$

$\$a >= \$b$   $\$a$  mayor o igual que  $\$b$

$\$a == \$b$   $\$a$  igual que  $\$b$

$\$a != \$b$   $\$a$  distinto que  $\$b$

### 2.3.6.9 Operadores lógicos

$\$a \text{ AND } \$b$  Verdadero si ambos son verdadero

$\$a \&\& \$b$  Verdadero si ambos son verdadero

$\$a \text{ OR } \$b$  Verdadero si alguno de los dos es verdadero

$\$a \text{ !! } \$b$  Verdadero si alguno de los dos es verdadero

$\$a \text{ XOR } \$b$  Verdadero si sólo uno de los dos es verdadero

$!\$a$  Verdadero si  $\$a$  es falso

### 2.3.6.10 Operadores de asignación

$\$a = \$b$  Asigna a  $\$a$  el contenido de  $\$b$

$\$a += \$b$  Le suma a  $\$b$  a  $\$a$

$\$a -= \$b$  Le resta a  $\$b$  a  $\$a$

$\$a *= \$b$  Multiplica  $\$a$  por  $\$b$  y lo asigna a  $\$a$

$\$a /= \$b$  Divide  $\$a$  por  $\$b$  y lo asigna a  $\$a$

$\$a .= \$b$  Añade la cadena  $\$b$  a la cadena  $\$a$

### 2.3.6.11 Constantes

PHP define como constantes aquellas que proporcionan un mecanismo para definir un valor que no cambiara con las acciones de la programación. Las constantes son como las variables, salvo por las dos circunstancias de que las constantes deben ser definidas usando la función.

Las constantes predefinidas (siempre disponibles) son:

Tabla 3. Constantes predefinidas

| Constante                | Descripción   |
|--------------------------|---|
| <code>__FILE__</code>    | Si se usa dentro de un archivo que ha sido incluido o requerido, entonces se da el nombre del archivo incluido, y no el nombre del archivo padre.       |
| <code>__LINE__</code>    | Si se usa dentro de un archivo incluido o requerido, entonces se da la posición dentro del archivo incluido.  |
| <code>PHP_VERSION</code> | La cadena que representa la versión del analizador de PHP en uso en la actualidad.  |
| <code>PHP_OS</code>      | El nombre del sistema operativo en el cuál se ejecuta el analizador PHP.<br>TRUE<br>FALSE   |
| <code>E_ERROR</code>     | Denota un error distinto de un error de interpretación del cual no es posible recuperarse.  |
| <code>E_WARNING</code>   | Denota una condición donde PHP reconoce que hay algo erróneo, pero continuará de todas formas; pueden ser capturados por el propio archivo de comandos. |
| <code>E_PARSE</code>     | El intérprete encontró sintaxis inválida en el archivo de comandos. La recuperación no es posible   |
| <code>E_NOTICE</code>    | Ocurrió algo que pudo ser o no un error. La ejecución continúa.   |
| <code>E_*</code>         | se usan típicamente con la función <b>error_reporting()</b> para configurar el nivel de informes de error.  |

Se pueden definir constantes adicionales usando la función **define()**.  
Nótese que son constantes, con una constante sólo se pueden representar datos escalares válidos

Ejemplo:

```
<?php
    function report_error($archivo, $linea, $mensaje)
    {
        echo "Un error ocurrió en $archivo en la línea $linea:
            $mensaje.";
    }
    report_error(__FILE__, __LINE__, "Algo esta mal!");
?>
```

Ahora se muestra la definición de las constantes propias:

```
<?php
    define("CONSTANTE", "Hola mundo.");
    echo CONSTANTE; // muestra "Hola mundo."
?>
```

Fuente: [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)

### 2.3.6.12 Sentencias de control<sup>14</sup>

Son aquellas que permiten realizar una programación más controlada. Estas sentencias finalizan siempre con un punto y coma, también están agrupadas como sentencias encapsuladas entre corchetes.

- **IF.-** Permite la ejecución condicional de fragmentos de código, esta expresión es evaluada a su valor booleano, si la expresión se evalúa como TRUE, se ejecutará la sentencia y si se evalúa como FALSE la ignorará.

#### Sintaxis

```
if ()
{
    sentencia
}
```

#### Ejemplo

```
if(!empty($_SESSION['nick'])) {
    $inicioSesion = 1;
}
```

- **ELSE.-** Se utiliza cuando se desea extender una sentencia if , en caso de que esta se evalúe como FALSE.

#### Sintaxis

```
if (expr)
{
    sentencia
}
else
{
    sentencia
}
```

---

<sup>14</sup>

<http://educagratis.cl/moodle/mod/resource/view.php?id=4105&usg=AOvVaw2V9ryQz7Xw2ijt2jvVZOKD>

## Ejemplo

```
if (isset($_GET['resultado']))
{
    $resultado = $_GET['resultado'];
    if ($resultado==1)
    {
        echo '<script type = "text/javascript"> alert("El registro ha sido
        actualizado");</script>';
    }
    else
    {
        echo '<script type = "text/javascript">alert("No se pudo actualizar el
        registro");</script>';
    }
}
```

- **ELSEIF/ELSE IF.**- A diferencia de IF ELSE esta sentencia es una expresión alternativa que solo se ejecutara si ELSE IF se evalúe como TRUE.

## Sintaxis

```
if (expr)
{
    sentencia
}
elseif
{
    sentencia
}
else
{
    sentencia
}
```

## Ejemplo

```
if (isset($_GET['resultado']))
{
    $resultado = $_GET['resultado'];
    if ($resultado==1)
    {
        echo '<script type = "text/javascript"> alert("Registro Ingresado");</script>';
    }
    else
    {
        if ($resultado==2)
        {
            echo '<script type = "text/javascript"> alert("No existe la cantidad
suficiente");</script>';
        }
        else
        if ($resultado==3){
            echo '<script type = "text/javascript">alert("Archivo cargado al
registro");</script>';
            $cargadoArchivo="1";
        }
    }
}
```

- **FOR.-** El bucle FOR evalúa la expresión al inicio del mismo si se evalúa como TRUE esta continúa su ejecución, si evalúa como FALSE finaliza la ejecución.

## Sintaxis

```
for (expr1; expr2; expr3)
{
    Sentencia
}
```

Cada una de las expresiones puede estar vacía o contener múltiples expresiones separadas por comas.



## Ejemplo

```
for ($i = 0; $i < $cantColores-1; $i++) {  
    echo "<option value=\"".$listaColores[$i].\" \"";  
    if ($parBien->getColor()==$listaColores[$i])  
        echo "selected";  
    echo ">".$listaColores[$i]."</option>>";  
}  
}
```

- **FOR EACH .-** Este constructor proporciona un modo sencillo de interactuar sobre arrays, foreach funciona sólo sobre arrays y objetos, y emitirá un error al intentar usarlo con una variable de un tipo diferente de datos o una variable no inicializada.

- 

## Sintaxis:

```
foreach (expresión_array as $valor)
```

```
{  
    sentencias  
}
```

```
foreach (expresión_array as $clave => $valor)
```

```
{  
    sentencias  
}
```

## Ejemplo

```
foreach ($pdo1->query($consulta_submenu) as $row_sub) {  
    $inicioSesion = false;  
    $salioSesion = false;  
    if(!empty($_SESSION['nick']) and ($row_sub['nombreMenu']=='Acceder al  
Sistema')) {  
        $inicioSesion=true;  
    }  
}
```

```

        if(empty($_SESSION['nick']) and ($row_sub['nombreMenu']=='Salir del
Sistema')) {
            $inicioSesion=true;
        }
        if (!$inicioSesion){
            $menu=$menu."<li><a
href="."$row_sub['direccionMenu'].>"."$row_sub['nombreMenu'].</a>";
        }
        //$subMenu.=$this->DaSubMenu($row_sub,$acceso);
        $consulta_submenu="SELECT * FROM menu WHERE idPadreMenu =
"."$row_sub['idMenu'].>" and accesoMenu <= "."$acceso.">";
        $menu = $menu."<ul>";

        foreach ($pdo2->query($consulta_submenu) as $row_sub_sub) {
            $menu=$menu."<li><a
href="."$row_sub_sub['direccionMenu'].>"."$row_sub_sub['nombreMenu'].</a>
</li>";
        }

        $menu=$menu."</ul></li>";
    }

```

### 2.3.7 Constructor<sup>15</sup>

Son métodos que a comparación de otros métodos que son invocados, los constructores se disparan cuando se crea un objeto, esto permite que el objeto creado tenga todas las características de la clase.

Esto es cuando como ejemplo:

```
$objeto = new Clase();
```

Una vez creada una clase y el objeto Persona

---

<sup>15</sup> <https://fernando-gaitan.com.ar/php-orientado-a-objetos-parte-2-metodo-constructor-y-destructor/>

### Ejemplo:

```
<?php
class Persona {
    public $nombre;
    public $apellido;
    public $edad;
    public function __construct($nombre, $apellido, $edad) {
        $this->nombre = $nombre;
        $this->apellido = $apellido;
        $this->edad = $edad;
    }
    public function saludar(){
        return 'Hola, soy ' . $this->nombre . ' ' . $this->apellido . ' y tengo ' . $this->edad . '
años ';
    }
}
?>
```

### objeto Persona:

```
$persona = new Persona();
```

Al ejecutar el archivo **test.php** se comprobara el funcionamiento. A esta clase se llama con el método **\_\_construct()**. En caso de que el método no tenga nombre se mostrara solo como **\_\_construct()**.

En tanto al crear el objeto y se define el valor de las propiedades y se lo realizara de la siguiente manera:

```
<?php
//Importamos la clase Persona.
require_once 'clases/Persona.php';
//Creamos el objeto con los valores que se definen en el constructor.
$persona = new Persona('Fernando', 'Gaitan', 26);
?>
```

Al finalizar solo tendremos que invocar un método como ejemplo **saludar()** y mostrar en pantalla:

```

<?php
//Importamos la clase Persona.
require_once 'clases/Persona.php';
//Creamos el objeto con los valores que se definen en el constructor.
$persona = new Persona('Fernando', 'Gaitan', 26);
//Mostramos por pantalla los valores.
echo $persona->saludar();
?>

```

### 2.3.8 Destructor

Cuando un objeto es eliminado no se borra del todo para lo cual utilizaremos el método **destruct** este se dispara cuando el objeto es eliminado e memoria.

#### Ejemplo:

```

<?php
class Persona {
    public $nombre;
    public $apellido;
    public $edad;
    public function __construct($nombre, $apellido, $edad) {
        $this->nombre = $nombre;
        $this->apellido = $apellido;
        $this->edad = $edad;
    }
    public function __destruct() {
        echo 'Objeto destruido';
    }
    public function saludar(){
        return 'Hola, soy ' . $this->nombre . ' ' . $this->apellido . ' y tengo ' . $this->edad . '
años ';
    }
}
?>

```

## 2.4 MYSQL

### 2.4.1 Historia<sup>16</sup>

MySQL sistemas de base de datos relacional, este sistema utiliza código abierto con licencia GPL de GNU, tuvo sus inicios con Michael Widenius en la época de los 90, utilizaba mSQL para crear sus propias rutinas de bajo nivel (ISAM), pero tras probarlas con unas pequeñas pruebas este se dio cuenta, que no se adaptaba a sus necesidades de programación, esto lo llevo a crear una interfaz SQL que tiene la misma aplicación API la misma que fue diseñada para soportar código de terceras partes que es totalmente compatible con mSQL.

El logo de MySQL es “Sakila” se obtuvo de una lista que los usuarios sugirieron en un concurso denominado “Name the Dolphin”. Ambrose Twebaze un programador de código abierto fue el ganador.

### 2.4.2 Estructura interna de MySQL

Se define en 4 partes que son:

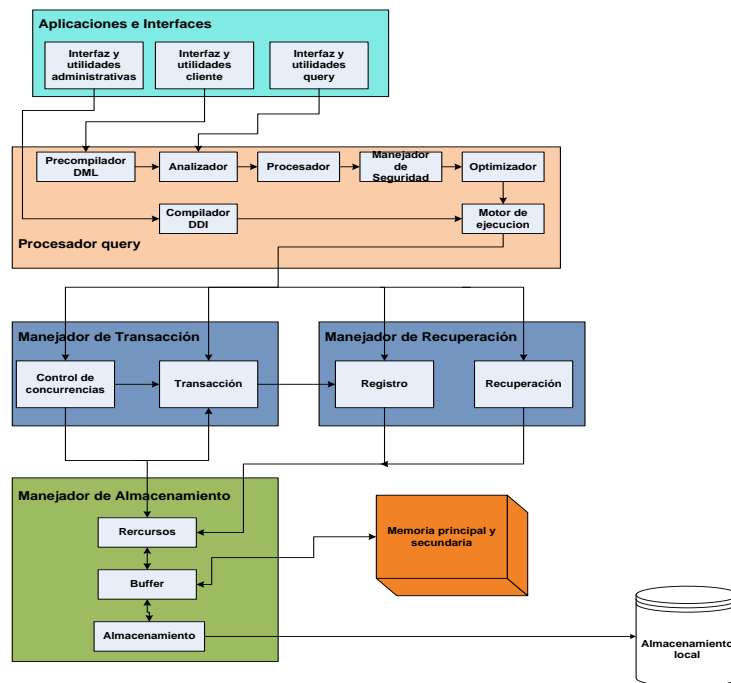


Figura 8. Estructura Interna de MySQL

Fuente: <https://tallerbd.wikispaces.com/-+Estructura+de+MySQL>

<sup>16</sup> <https://downloads.mysql.com/docs/refman-5.0-es.pdf>

### 2.4.3 Aplicaciones e Interfaces<sup>17</sup>

Aquí los clientes y usuarios interactúan con la Base de Datos, la misma que consta de tres elementos.

- Interfaz y utilidades administrativa
- Interfaz y utilidades cliente
- Interfaz y utilidades query

#### 2.4.3.1 Procesador Query

Cuando un usuario desea visualizar o manipular los datos de un sistema, es donde se produce la mayor carga de interacción sistema en el almacenaje.

- **Precompiler Encajado de DML.-** Cuando una petición se realiza en la etapa anterior es aquí en donde se precompila la petición en el DML (lengua de manipulación de datos), la misma que extrae las peticiones relevantes.
- **Recopilador de DDL.-** Aquí se tiene acceso a las bases de datos de MySQL que son recibidas por el administrador y procesadas por el DDL
- **Programa de análisis de Query.-** En esta etapa, el objetivo del programa es el análisis de dicha pregunta y transformarla en una pregunta, que pueda ser comprendida por los otros componentes.
- **Preprocesador Query.-** Con la pregunta anterior se analiza, y según lo obtenido se utiliza el procesador, el cual se comprueba la sintaxis y la semántica del programa para determinar si dicha pregunta es válida.
- **Encargado De Security/Integration.-** Cuando el procesador anterior demuestra que la pregunta es TRUE se comprueba el Access Control List para saber si hay el cliente.
- **Optimizador De Query.-** Al realizar los pasos anteriores MySQL utiliza el optimizador para ejecutar preguntas tan rápidamente como le sean posibles y obtener la respuesta solicitadas.
- **Motor De la Ejecución.-** al entregar la pregunta a l motor de ejecución este realiza una optimización para que se pueda ejecutar la búsqueda solicitada en MySQL, para después proceder a ejecutar las declaraciones que tengan acceso a la capa física de la base de datos.

---

<sup>17</sup> <https://tallerbd.wikispaces.com/-+Estructura+de+MySQL>

### 2.4.3.2 Manejador de Transacción y Recuperación

Se explica cada funcionamiento en esta etapa

- **Gestor de transacciones.-** El gestor de transacciones es responsable de cerciorarse de que la transacción está registrada y ejecutada.
- **Control Concurrencia.-** Una vez que se adquiriera la cerradura, solo se permite una consulta a la vez. Si varias peticiones intentan manipular los mismos datos, el encargado de la concurrencia-control rechaza la petición hasta que la primera transacción es completa.
- **Encargado Del Registro.-** Responsable de registrar la operación de una misma petición en la base de datos.
- **Encargado De la Recuperación.-** Aquí permite que una vez que se encuentre

### 2.4.3.3 Manejador de Almacenado

Etapa donde se realiza un cierto tipo de almacenaje secundario no obstante el acceso dinámico de este medio no es práctico. Mediante unos almacenadores intermedios los cuales residen en memoria principal o virtual. A continuación se muestra la estructura:

- **Encargado Del Almacenaje.-** Aquí se trata de las peticiones del almacenaje almacenador intermediario y el almacenaje secundario.
- **Encargado Del Buffer.-** Es aquel que se encarga de tomar las peticiones ajustarlas al formato para la siguiente etapa.
- **Encargado De Recurso.-** El encargado de recurso recibe referencias a los datos dentro de la memoria del encargado del almacenador intermediario y vuelve estos datos a las capas superiores.

## 2.4.4 Características o Ventajas<sup>18</sup>

MySQL describe las características que a su vez se convierten en ventajas más importantes:

---

<sup>18</sup> <http://ftp.tsrc.edu.tw/MySQL/doc/refman/5.0/es/features.htm>

- Fácil utilización ya que es basado en C y en C++
- Se puede compilar en varios sistemas.
- Es fácil de instalar en diferentes plataformas.
- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Utilidad al añadir interfaz SQL a otros sistemas de almacenamiento
- Tiene un sistema de reserva de memoria muy rápido basado en threads.
- Las tablas hash en memoria, que son usadas como tablas temporales.
- No existe reserva de memoria tras compilar las consultas.
- Librerías optimizadas y rápidas.

Tiene un entorno de red cliente/servidor esto se da gracias a que el servidor se encuentra como un programa separado.

- También está disponible como biblioteca y puede ser incrustado en aplicaciones autónomas.
- Cuando no existen red disponible las aplicaciones pueden usarse.
- Permite la migración a otros sistemas de bases de datos.
- El código se puede ejecutar sentencias desde un fichero de texto.

#### **2.4.5 Debilidades**

MySQL al ser un sistema que se encuentra creada como open source sus debilidades se han ido corrigiendo con el tiempo.

- Dificultad en la normalización de tablas.
- A pesar de existir una comunidad oficial de MySQL, todavía existe documentación no oficial en las redes.
- Su interfaz es poco intuitiva.

#### **2.4.6 Código Base <sup>19</sup>**

##### **2.4.6.1 Tipos de datos**

---

<sup>19</sup> <https://disenowebakus.net/tipos-de-datos-mysql.php>



Una vez terminada la fase de diseño, se procede a la implementación de la base de datos. Para lo cual, se debe crear tablas con sus respectivos tipos de datos y características. Pudiendo dividirse en 3 grupos principales como son:

Los tipos de datos que puede haber en un campo, se pueden agrupar en tres grandes grupos:

- **Datos numéricos**

El rango de valores es la única diferencia entre algunos tipos de datos. Entre estos tipos se encuentran:

- **Numéricos enteros**

Tabla 4. Detalle de tipos de datos numéricos enteros

| <b>Tipos de datos</b> | <b>Bytes</b> | <b>Valor mínimo</b>  | <b>Valor máximo</b> |
|-----------------------|--------------|----------------------|---------------------|
| TINYINT               | 1            | -128                 | 127                 |
| SMALLINT              | 2            | -32768               | 32767               |
| MEDIUMINT             | 3            | -8388608             | 8388607             |
| INT o<br>INTEGER      | 4            | -2147483648          | 2147483647          |
| BIGINT                | 8            | -9223372036854775808 | 9223372036854775807 |

Fuente: <https://disenowebakus.net/tipos-de-datos-mysql.php>

- **También se observan los tipos de datos sin signo:**

Tabla 5. Datos sin signos

| <b>Tipo de dato</b> | <b>Bytes</b> | <b>Valor mínimo</b> | <b>Valor máximo</b> |
|---------------------|--------------|---------------------|---------------------|
| TINYINT             | 1            | 0                   | 255                 |
| SMALLINT            | 2            | 0                   | 65535               |
| MEDIUMINT           | 3            | 0                   | 16777215            |
| INT                 | 4            | 0                   | 4294967295          |

| Tipo de dato | Bytes | Valor mínimo | Valor máximo         |
|--------------|-------|--------------|----------------------|
| BIGINT       | 8     | 0            | 18446744073709551615 |

Fuente: <https://disenowebakus.net/tipos-de-datos-mysql.php>

Los mismos que se definen mediante el modificador **UNSIGNED**.

○ **Números con decimales**

Tabla 6. Tipos de datos con números decimales

| Tipos de datos | Decimales | Valor mínimo | Valor máximo |
|----------------|-----------|--------------|--------------|
| FLOAT          | 0-24      | -999.99      | 999.99       |
| DOUBLE         | 25-53     | -999.99      | 999.99       |
| DECIMAL        | 64        | -999.99      | 999.99       |

Fuente: <https://disenowebakus.net/tipos-de-datos-mysql.php>

En los datos DOUBLE se puede usar **FLOAT**, que es de simple precisión, pero esto también puede llevar a que existan problemas de redondeo y pérdida de los decimales restantes.

• **Datos alfanuméricos**

Para almacenar datos alfanuméricos (cadenas de caracteres) en MySQL se aprecian los siguientes tipos de datos:

Tabla 7. Tipos de datos alfanuméricos

| Tipos de datos           | Longitud de caracteres   |              |
|--------------------------|--|--------------|
|                          | Valor mínimo   | Valor máximo |
| CHAR                     | 1  | 255          |
| VARCHAR                  | 1  | 65.535       |
| BINARY<br>y<br>VARBINARY | Son idénticos que CHAR y VARCHAR con la diferencia que aquí se almacenan bytes |              |
| TEXT                     | 1  | 65.535       |

| Tipos de datos | Longitud de caracteres  |        |
|----------------|---|--------|
| BLOB           | Es utilizado cuando se almacena archivos binarios, pero aquí solo se almacena la imagen y no la ruta. También se tiene:<br><b>TINYBLOB</b> es de <b>255 bytes</b><br><b>MEDIUMBLOB</b> es de <b>16.777.215 bytes</b> , y<br><b>LONGBLOB</b> es de <b>4 Gb</b> (o lo máximo que permita manipular el sistema operativo). |        |
| ENUM           | 1   | 65.535 |
| SET(conjunto)  | Al igual que ENUM, se debe especificar una lista, pero de hasta 64 opciones solamente.  |        |

Fuente: <https://disenowebakus.net/tipos-de-datos-mysql.php>

- La principal desventaja de TEXT es que no puede indexarse fácilmente (a diferencia de VARCHAR).
- En el tipo de dato ENUM si se desea separar los datos por comas estas deben estar envueltas con comillas.

- **Datos de fecha y hora**

Dentro de las diferencias de uno y otro, y sus usos principales, se podrán elegir el tipo de dato apropiado en cada caso.

- **DATE.-** Este tipo de dato nos permite almacenar fechas con formato:  
**AAAA-MM-DD** en donde los cuatro primeros dígitos para el **año**, los dos siguientes para el **mes**, y los últimos dos para el **día**. Tomando muy en cuenta que MySQL es susceptible a la fecha no se puede ingresar la información DD-MM-AAAA.
- **DATETIME.-** Este tipo de dato permite no solo el ingreso de la fecha sino también de la hora en formato:  
**AAAA-MM-DD HH:MM:SS** en donde los primeros rangos es igual a **DATE** y el segundo rango es para la hora

- **TIME.-** Aquí nos permite almacenar datos de tipo tiempo en horas, minutos y segundos, en el formato **HH:MM:SS**, y su rango permitido va desde **-839:59:59** hasta **839:59:59**.

- **Atributos de los campos**

Al conocer los diferentes tipos de datos, estos deben estar definidos por atributos que se especifican al crear los campos (esto como consejo para facilitar la programación, los mismos que durante la programación puede variar).

Tabla 8. Atributos de los campos

| Nombre                         | Descripción  |
|--------------------------------|--|
| ¿Null o Not Null?              | En programación existen varias opciones en donde el ingreso de la información puede estar vacía, por lo tanto se utiliza este tipo de atributo para no crear conflictos. |
| Valor predeterminado (default) | Muchas veces se necesita agilizar la carga de datos mediante un valor por defecto (default).   |
| Ordenamiento binario           | Se utiliza junto a los datos CHAR y VARCHAR, este permite ordenar los datos mediante mayúsculas y minúsculas.  |

Fuente: <https://disenowebakus.net/tipos-de-datos-mysql.php>

- **Índices**

Con la creación de este campo, ayuda a mantener el orden en los registros, esto no es más que una tabla paralela (virtual), la misma que ordena por campos y no por orden de inserción.

Tabla 9. Creación del campo índices

|          |   |
|----------|---|
| UNIQUE   | Aquí se obliga al campo a no repetirse en el registro pero considerando que no es clave primaria.   |
| FULLTEXT | En el campo tipo text este ayuda a almacenar datos palabra por palabra, almacenándolas en una celda de una matriz ayudando de esta manera a una mejor búsqueda. |

Fuente: <https://disenowebakus.net/tipos-de-datos-mysql.php>

## 2.4.6.2 Herramientas y estructura de MYSQL<sup>20</sup>

- **Vistas**

Es una estructura estática que facilita y restringe el acceso a los datos a usuarios limitados.

Por ende, si una vista tiene un `SELECT * FROM tabla` y después de definirla se le añade un campo a la tabla original, no lo podrá ser vista.

Si después ocurren cambios en la definición de algún campo de la tabla original entonces se generará un error/warning, al utilizarla pues seguramente tendrá que truncar valores: los campos eliminados los pasará a NULL.

**Ejemplo:** Creación de una vista para ver las ventas por vendedor:

- `CREATE VIEW Ventasporvendedor AS`
- `SELECT idvendedor`
- `SUM (cantidad * precio) total`
- `FROM pedidos`
- `GROUP by idvendedor`
- `ORDER BY total DESC`

- **Trigger**

Son procedimientos que permiten ejecutar acciones automáticas. No se pueden invocar explícitamente, porque su ejecución está asociada a una tabla/transacción/tiempo.

**Ejemplo** antes de insertar una venta se considera necesario verificar si el producto está disponible en stock, el evento insert dispara el trigger verstock automáticamente.

---

<sup>20</sup> <https://www.solvetic.com/tutoriales/article/559-herramientas-y-estructura-de-mysql-5/>

- CREATE TRIGGER verstock BEFORE INSERT ON Ventas
- FOR EACH ROW
- BEGIN
- DECLARE idproducto INT;
- IF (idproducto) THEN
- SELECT stock FROM productos where id=idproducto;
- END IF;
- END;

- **Procedimientos almacenados**

Un procedimiento almacenado es un conjunto de comandos que pueden acumularse en el servidor. Realiza este proceso los clientes no necesitan relanzar los comandos individuales pero pueden en su lugar referirse al procedimiento almacenado.

- CREATE PROCEDURE Graba\_Venta (IN tipoventa INTEGER)
- BEGIN
- DECLARE tipo CHAR(10);
- IF tipoventa = 1 THEN
- SET tipo = 'Minorista';
- ELSE
- SET tipo = 'Mayorista';
- END IF;
- INSERT INTO ventas (tipo) VALUES (tipo);
- END

Luego se invoca de la siguiente manera

- CALL Graba\_venta(1)

Otro ejemplo de procedimiento almacenado

- CREATE PROCEDURE ProcedimientoInsertar
- (
- in Nombre varchar(50),
- in Telefono varchar(50)

- )
- BEGIN
- insert into contactos(nombre,telefono) values(Nombre,Telefono);
- END

- **Funciones**

Las funciones están pensadas para devolver parámetros, pueden ser llamadas dentro de sentencias como select, insert, etc. Así también, no pueden devolver consultas, solo sirven para realizar tareas donde se requiera algún tratamiento de datos, calculo o conversiones.

**Por ejemplo:**

- CREATE FUNCTION descuento(precio, porcentaje FLOAT)
- RETURNS FLOAT DETERMINISTIC
- BEGIN
- DECLARE importedesconto FLOAT;
- SET importedesconto = precio\*porcentaje/100;
- RETURN importedesconto;
- END \$\$

## 2.5 JAVASCRIPT <sup>21</sup>

Javascript permite a los desarrolladores expertos y a quienes se inician en la programación crear acciones en sus páginas web. Este lenguaje no requiere de un compilador porque funciona del lado del cliente.

Entre Java y JavaScript existe una confusión, pero estos lenguajes son totalmente diferentes. JavaScript tiene varias ventajas para páginas web una de las cuales es la visualización sin la necesidad de instalar otro programa.

---

<sup>21</sup> <http://www.maestrosdelweb.com/que-es-javascript/>

### **2.5.1 Historia**

La primera versión nació en los años de 1990, su creador fue Netscape que creó LiveScript; en sus inicios este lenguaje fue diseñado para un grupo pequeño de diseñadores Web sin tener la necesidad de un compilador o con experiencia en orientación objetos.

En diciembre de 1995, Netscape y Sun Microsystems creó la versión de JavaScript muy popular en el mercado.

A JavaScript, se le conoce como un lenguaje que permite la creación de pequeños programas con muchas posibilidades, los cuales posteriormente son insertados en una página web, programas más grandes, o programas de orientados a objetos mucho más complejos. Al mismo tiempo que también se puede crear diferentes efectos e interactuar con los usuarios.

### **2.5.2 Características de JavaScript**

Entre las características más notables que lo ha llevado a ser un lenguaje de mayor demanda y más conocido por los desarrolladores son:

- Es un lenguaje basado en acciones que posee menos restricciones.
- Gran parte de la programación lenguaje está centrada en describir objetos, escribir funciones (que no son necesarias los movimientos del mouse), aperturas, utilización de teclas, cargas de páginas entre otros.
- También se define 2 tipos de JavaScript, el primero es aquel que se ejecuta en el lado del cliente que se lo conoce como Navigator JavaScript. El segundo es aquel que se ejecuta en el lado del servidor y se denomina LiveWire JavaScript.
- Se usa una sintaxis similar al de Java y C.
- Es un lenguaje que no necesita instalación de ningún Framework
- Permite la programación orientada a objetos: `document.write("Hola");`



### 2.5.3 Usos de JavaScript

Entre los diferentes servicios o usos de este lenguaje denotan los siguientes:

- Correo
- Chat
- Buscadores de Información

Dentro de los usos de su código puede insertar características en páginas tales como:

- Reloj
- Contadores de visitas
- Fechas
- Calculadoras
- Validadores de formularios
- Detectores de navegadores e idiomas

### 2.5.4 ¿Cómo identificar código Javascript?

Este código se identifica de acuerdo a la inserción que ha tenido dentro de una página web en medio de las etiquetas

- `<body></body>`
- `<script></script>`.
- Se puede ubicar en ficheros externos usando: `<script type="text/javascript" src="micodigo.js"></script>`

### 2.5.5 Compatibilidad con navegadores

Al ser un lenguaje de código abierto este tiene una gran aceptación por varios tipos de navegadores como son:

- Internet Explorer
- Netscape
- Opera

- Mozilla Firefox
- Entre otros.

Aunque, JavaScript sea soportado en gran cantidad de navegadores, los usuarios pueden elegir la opción de Activar/Desactivar.

## **2.6 CSS3<sup>22</sup>**

### **2.6.1 Concepto**

Sus siglas en ingles son Cascade Style Sheets, conocidas como Hojas de Estilo en Cascada. Este es un lenguaje de marcado para sitios WEB, esto quiere decir que funciona en un conjunto de archivos HTML, para lo cual, se debe conocer este tipo de programación y CSS.

Es un lenguaje apto para ser desarrollado dentro de un documento HTML pero por motivos de productividad se suele realizar aparte una extensión .css.

### **2.6.2 Características**

#### **2.6.2.1 Es gratuito**

No requiere de ningún software costoso para poder realizar un programa en CSS, más que un block de notas para su codificación. Esto podría ser un problema, si se desea programar muchos estilos distintos, pues no permitía distinguir las etiquetas y dificultaría el encontrar errores.

#### **2.6.2.2 Todos los estilos en una sola página**

Se puede usar un solo archivo CSS para varias páginas. De modo que puede definir todos los estilos de un sitio web y vincularlos mediante las etiquetas respectivas según corresponda. Lo que cambiaría el estilo que se repite en todas las páginas.

---

<sup>22</sup> <http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>

### 2.6.2.3 Genera sitios más rápidos

Al no necesitar de un repositorio para su ejecución esto mejora el tiempo de respuesta del sitio web. Lo que permite la repetición del código en los archivos HTML, generando un ahorro de tiempo valioso y mayor peso en las páginas.

## 2.6.3 Usos de CSS3

### 2.6.3.1 Animaciones y transiciones en CSS

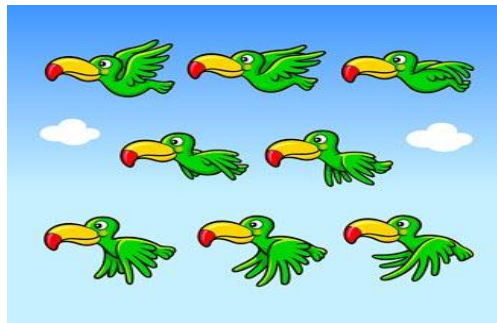


Figura 9. Modelo de animaciones y transiciones en CSS

Fuente: <http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>

Permite añadir en una página web, animaciones y transiciones estas pueden cambiar la apariencia, comportamiento de los elementos de una página. Estos comportamientos se consiguen en base a fotogramas.

De esta manera, se puede añadir mayor interactividad a cualquier sitio sin recurrir a JavaScript o Flash, sólo con el poder de CSS3.

### 2.6.3.2 Layout con columnas



Figura 10. Ejemplo de layout en columnas en CSS

Fuente: <http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>

En toda página web se trabaja en base a cuadrículas. En este caso el número de columnas es mayor esto es gracias a la propiedad “Multicolumn Layout” de CSS3. Esta propiedad no sólo permite especificar el número de columnas del layout, sino también el ancho de cada columna.

### 2.6.3.3 Gradientes



Figura 11. Ejemplo de gradientes en CSS

Fuente:<http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>

CSS3 no sólo permite modificar el color de los elementos que conforman página, sino que también permite crear gradientes. Los gradientes lineales, son creadas especificando la dirección o añadiendo un ángulo. Puedes elegir de dos a más colores para tus gradientes e incluso añadirle transparencia, de modo que, el nivel creativo realmente no tenga límites al momento de crear los gradientes CSS3.

### 2.6.3.4 Rotación de elementos



Figura 12. Relaciones entre elementos en CSS

Fuente:<http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>

La propiedad “Transform” de CSS3 no sólo permite rotar elementos, sino también cambiar su tamaño y moverlos. “Transform” también realiza modificaciones a elementos 3D, girarlos, cambiarlos de tamaño que se desee, se lo puede aplicar también a objetos en 2D.

#### 2.6.3.5 Uso de fuentes



Figura 13. Ejemplo de fuentes en CSS

Fuente:<http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>

Con CSS se obtiene un gran número de fuentes gratuitas, estas al ser colocadas en el archivo CSS que se mostrará en la página web.

**CAPÍTULO III**  
**3 ANÁLISIS Y DISEÑO**

### **3.1 RECOLECCIÓN DE LA INFORMACIÓN**

La información fue recopilada en el Cuerpo de Bomberos Mocha, donde se verificaron las necesidades que poseen sobre el control de materiales y equipamiento que posee la organización. Además de potencializar cada uno de los servicios destinados para el bien de la comunidad, cubiertos cada uno de los requerimientos mencionados en el desarrollo del sitio web.

Esta herramienta tecnológica tiene varias funciones, dentro de las primordiales se encuentran las siguientes:

- La promoción y difusión a la comunidad de los servicios y horarios que presta el Cuerpo de Bomberos Mocha.
- La definición de los usuarios para la administración del sitio web
- Otorga la manipulación de datos referentes a los materiales y equipos del Cuerpo de Bomberos Mocha
- Permite el almacenamiento de información mediante una base de datos.

#### **3.1.1 Alcance**

El sitio web permitirá el control de materiales, equipos y la promoción de los servicios que ofrece el Cuerpo de Bomberos Mocha, con la finalidad de servir a la comunidad mediante una herramienta útil, promoviendo el buen manejo de la información interna y se puedan tomar acciones importantes que generen un adecuado control de los recursos.

### **3.2 ANÁLISIS**

El análisis se lo realizó previo a la etapa de diseño donde se describen aquellas funcionalidades que permitirán cubrir las necesidades que posee el Cuerpo de Bomberos Mocha sobre la custodia o buen uso de sus materiales y equipos, además de ofrecer servicios publicitarios o datos informativos para que la comunidad pueda enterarse del desarrollo que posee dicha organización.

Siendo las principales funciones del sitio web las siguientes:

- El ingreso al sitio web se realizó mediante el uso de un link, lo que implica la generación de la arquitectura cliente-servidor.
- Permite el ingreso de los materiales y equipos al sitio web de forma sencilla
- Organiza la información ingresada y la almacena en su base de datos
- Potencializa la promoción del Cuerpo de Bomberos de Mocha mediante la incorporación de datos de los servicios, campañas, horarios, noticias, etc.
- Finalmente establece un reporte de las transacciones realizadas para establecer un control constante del inventario existente.

### **3.2.1 Estudio de factibilidad**

Por medio de las siguientes fases el desarrollo del sitio web se consideró viable para su diseño e implementación.

#### **3.2.1.1 Factibilidad operativa**

El sitio web fue desarrollado por la proponente del presente trabajo de investigación, quien por lo tanto está capacitado de difundir los conocimientos necesarios al personal sobre el manejo de esta herramienta tecnológica. Como el enfoque del presente trabajo es brindar soporte y apoyo en el manejo del inventario del establecimiento, el personal se encontró dispuesto a recibir la información descrita, haciendo que la realización del sitio web sea factible operativamente.

#### **3.2.1.2 Factibilidad Técnica**

El sitio web fue desarrollado bajo licencia Open Source mediante el uso del lenguaje de programación PHP orientado a objetos con IDE NetBeans y base de datos MySQL todas las herramientas son software libre que facilitan el desarrollo de sitios web .



Tabla 1. Factibilidad Técnica

| <b>HARDWARE</b>   |                |
|---|----------------|
| <b>Computador</b>   |                |
| - Modelo: HP G42 Notebook Pc.                               |                |
| - Processor: AMD Athlon (tm) II P360<br>Dual-Core Processor |                |
| - Memoria instalada (RAM): 2,00 GB                          |                |
| Sistema operativo de 32 bits                                |                |
| <b>SOFTWARE</b>   |                |
| <b>Windows 7</b>  |                |
| Xampp 5.5.27  | Software libre |
| Php   | Software libre |
| IDE NetBeans  | Software libre |
| MySQL   | Software libre |

Fuente: Elena Mariño

Con relación al Hardware el computador es de propiedad del proponente. Por su parte el software es de código abierto que quiere decir que cuenta con libre acceso y además cuenta con una curva baja de aprendizaje de esta manera los recursos, materiales y programas necesarios para la elaboración y ejecución son accesibles delimitando la factibilidad técnica.

### **3.2.1.3 Factibilidad económica**

Existieron todos los recursos económicos necesarios para la realización del sitio web, a pesar de que el software era de uso libre, se requirieron de otros componentes como el Hosting que fue cubierto en su totalidad por el investigador, por tanto la factibilidad económica fue establecida.

### **3.2.1.4 Factibilidad legal**

Por otra parte, el sitio web se encuentra con todos los derechos de autor y propiedad intelectual. Haciendo que de esta manera el proyecto resultará legalmente factible, es decir no infringe ninguna ley vigente, por lo cual pudo ser implementada por el Cuerpo de Bomberos Mocha.

### **3.2.2 Análisis de los requerimientos**

Los requerimientos representan aquellas características que el catálogo tendrá para el cumplimiento de las necesidades que posee el cuerpo de Bomberos Mocha.

Para ello, se definen dos requerimientos que a continuación se detalla:

### 3.2.2.1 Requerimientos funcionales del sistema

Esta clase de requerimientos delimita la funcionalidad del sitio web con su entorno y como actúa dependiendo la actividad impuesta.

- **Presentar Página Informativa.-** El sitio Web presenta una página con las actividades, servicios, números de contacto y demás información relevante del Cuerpo de Bomberos Mocha
- **Ingresar, Listar, Actualizar y Dar de Baja los Bienes al Registro.-** El Sitio Web permite el registro de nuevos bienes que estén bajo custodia del personal del Cuerpo de Bomberos Mocha, consintiendo la actualización de la información registrada o dar de baja un bien del registro.
- **Ingresar, Listar, Actualizar y Dar de Baja a los Custodios del sistema.-** El Sitio Web registra nuevos Custodios que ingresen al Cuerpo de Bomberos Mocha, además permite la actualización de la información registrada o darlos de baja del registro.
- **Ingresar, Listar, Actualizar y Dar de Baja a los Usuarios del sistema.-** El Sitio Web, admite el registro de nuevos usuarios autorizados para ingresar a la parte administrativa, así también permite la actualización de información o eliminar ciertos datos.
- **Asignar Custodio al Bien y Listar Bienes del Custodio.-** Se podrá asignar Custodios a los Bienes, quienes tendrán a su cargo el cuidado y conservación de los mismos. Además se podrá obtener un listado con los bienes de cada Custodio.
- **Permitir Acceso a Usuarios.-** Mediante el uso de un usuario y clave, se tendrá el acceso a la parte administrativa del sitio.
- **Lista y carga de Archivos Leyes.-** Para dar cumplimiento a la Ley de Transparencia y Acceso a la Información pública, el usuario Administrador podrá cargar archivos y presentarlos en una lista a los usuarios.

### 3.2.2.2 Requerimientos no funcionales del sistema

Representan las restricciones y características del desarrollo del proyecto, indican las dificultades que pueden afrontarse y especifican los criterios bajo los cuales se va a evaluar la operación del sitio web.

- **Disponibilidad:** El sitio web del Cuerpo de Bomberos Mocha está disponible y al servicio de los usuarios las 24 horas al día, los 365 días del año. Tomando en consideración la realización de mantenimientos e implementación de mejoras que pueden realizarse en línea.
- **Escalabilidad:** El sitio Web del Cuerpo de Bomberos Mocha tendrá una alta capacidad para soportar cargas crecientes de trabajo, sin desmejorar sus tiempos de respuesta.
- **Seguridad:** El Sitio Web del Cuerpo de Bomberos Mocha cuenta con las seguridades necesarias para garantizar la integridad y disponibilidad de los datos almacenados.
- **Eficiencia:** El Sitio Web tiene la capacidad para realizar adecuadamente las funciones requeridas por los usuarios, presentando en poco tiempo una respuesta efectiva al requerimiento solicitado.

### 3.2.3 Definición de Casos de Uso

- **Actores**

**Usuario:** Se refiere a la persona que accede al sitio web sin necesidad de estar registrado en el sistema

**Administrador:** Se refiere a la persona que está registrada en el sistema y tiene permiso para realizar los procesos.

- **Presentar página informativa**

**Objetivo:** Mostrar en el navegador web la página de inicio del sitio web.

**Proceso:**

1. El usuario accede en el navegador web a la dirección URL del sitio.
2. El servidor Web consulta en la base de datos la información.
3. El servidor Web presenta la página de Inicio al usuario.

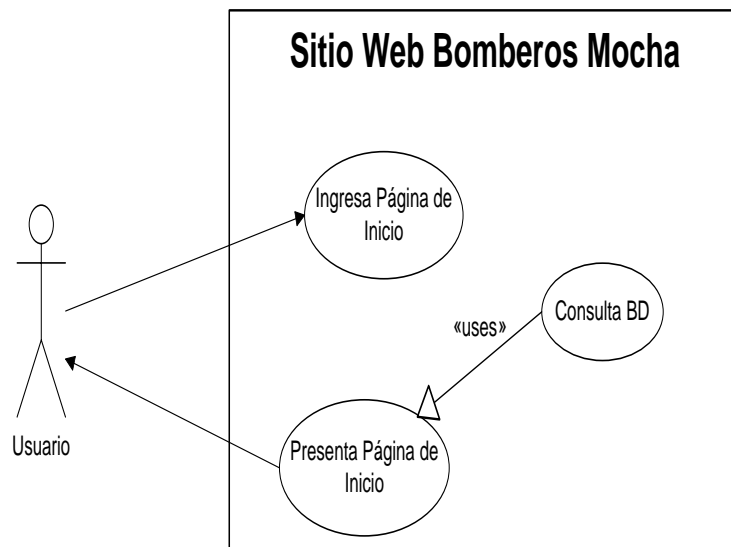


Figura 14. Caso de uso página informativa

Fuente: Elena Mariño

- **Ingresar al sistema**

**Objetivo:** Acceder al componente de la administración del sitio Web

**Proceso:**

1. El Administrador selecciona la opción de Ingreso
2. El Sistema presenta la página de inicio.
3. El Administrador Ingresa las Credenciales de Acceso.
4. El Sistema verifica en la base de datos que los datos ingresados sean válidos.
5. Si las credenciales son válidas, permite el acceso.
6. Si las credenciales no son válidas presenta mensaje de error.

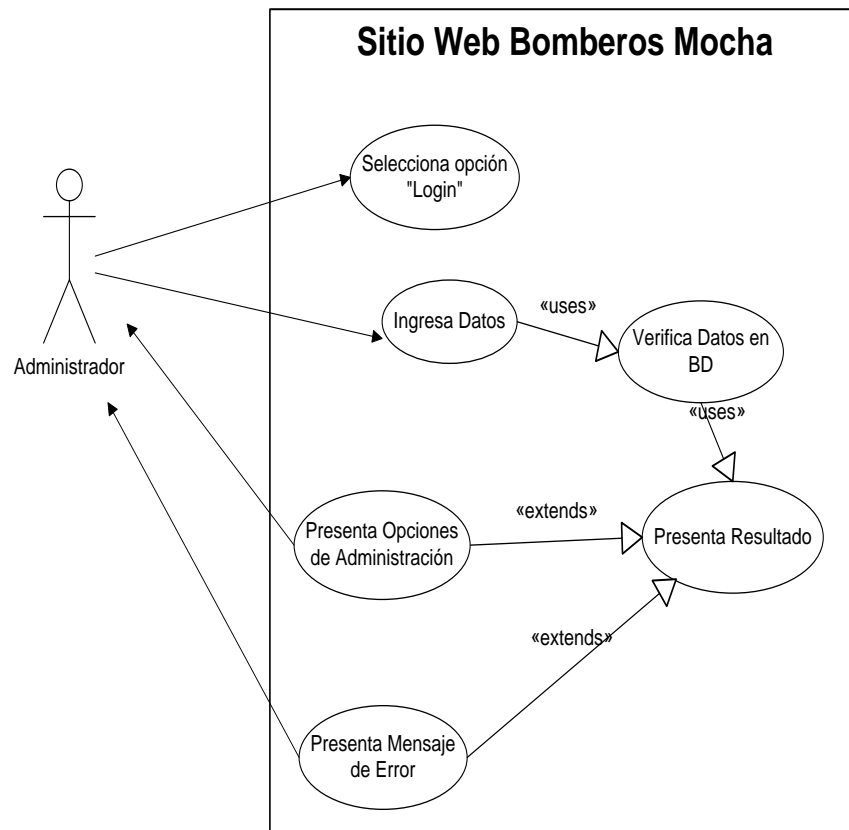


Figura 15. Caso de uso ingresar al sistema

Fuente: Elena Mariño

- **Ingresar y Actualización de datos al registro**

**Objetivo:** Grabar en la base de datos la información ingresada y actualizada

**Proceso:**

1. El Administrador selecciona una opción.
2. El Administrador elige los campos a modificar o ingresar
3. El Administrador los campos de acuerdo con el menú que desee
4. Modifica o Ingresa los datos a actualizar
5. El Sistema valida los datos ingresados
  - Si los datos son incorrectos muestra mensaje de error.
  - Si los datos son correctos almacena la información en la base de datos.
6. El sistema presenta el resultado.

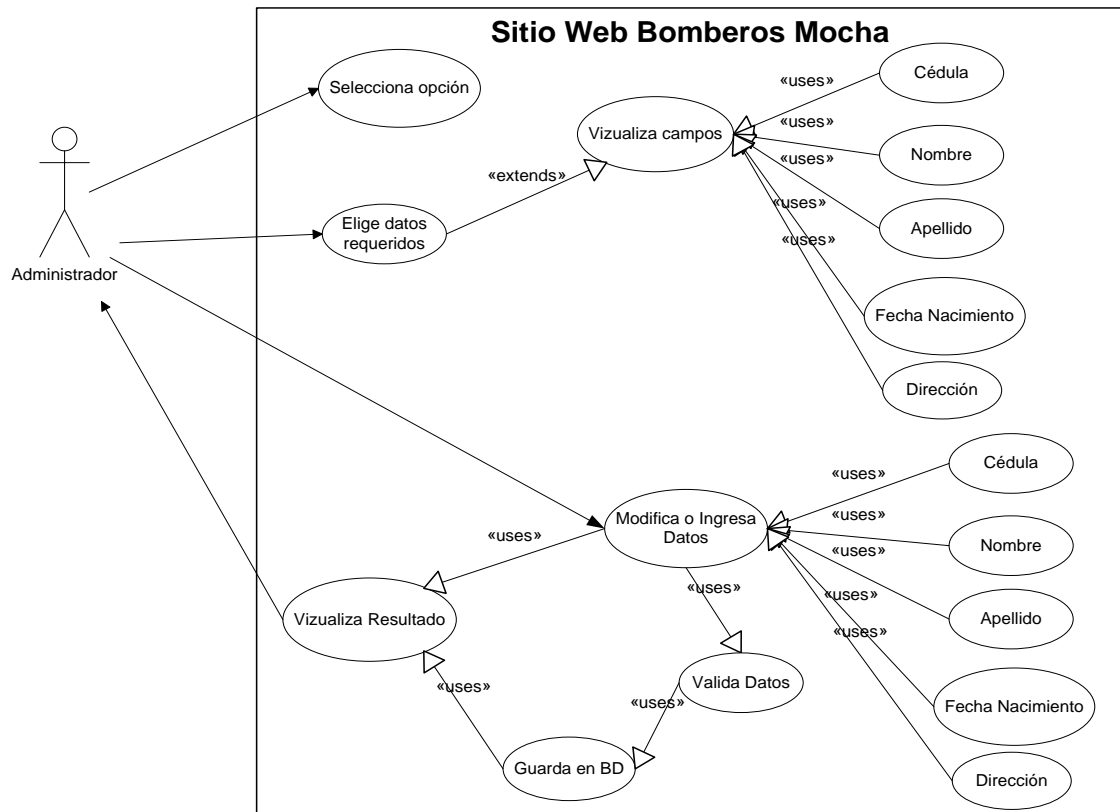


Figura 16. Caso de uso ingresar o actualización datos de registro

Fuente: Elena Mariño

- **Generar reportes**

**Objetivo:** Presentar un Reporte según el criterio seleccionado

**Proceso:**

1. El Administrador selecciona la opción de consulta
2. El sistema presenta las opciones disponibles
3. El Administrador ingresa el criterio de consulta
4. El sistema presenta los datos solicitados
5. El Administrador selecciona la opción generar reporte
6. El sistema presenta el reporte solicitado.

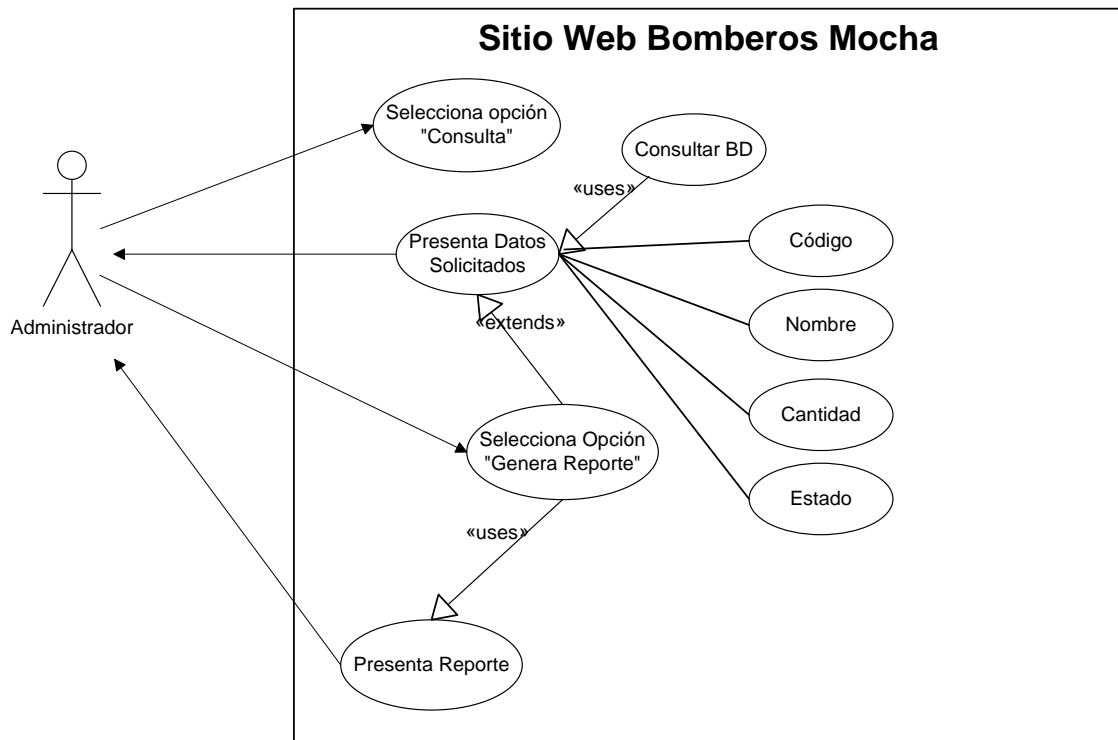


Figura 17. Caso de uso generar reportes

Fuente: Elena Mariño

### 3.3 DISEÑO

El diseño basa su desarrollo en la explicación del funcionamiento de la base de datos en el sitio web, particularidad que servirá para el almacenamiento y la recuperación de la información, para lo cual se describen los siguientes aspectos.

### 3.3.1 Diseño conceptual

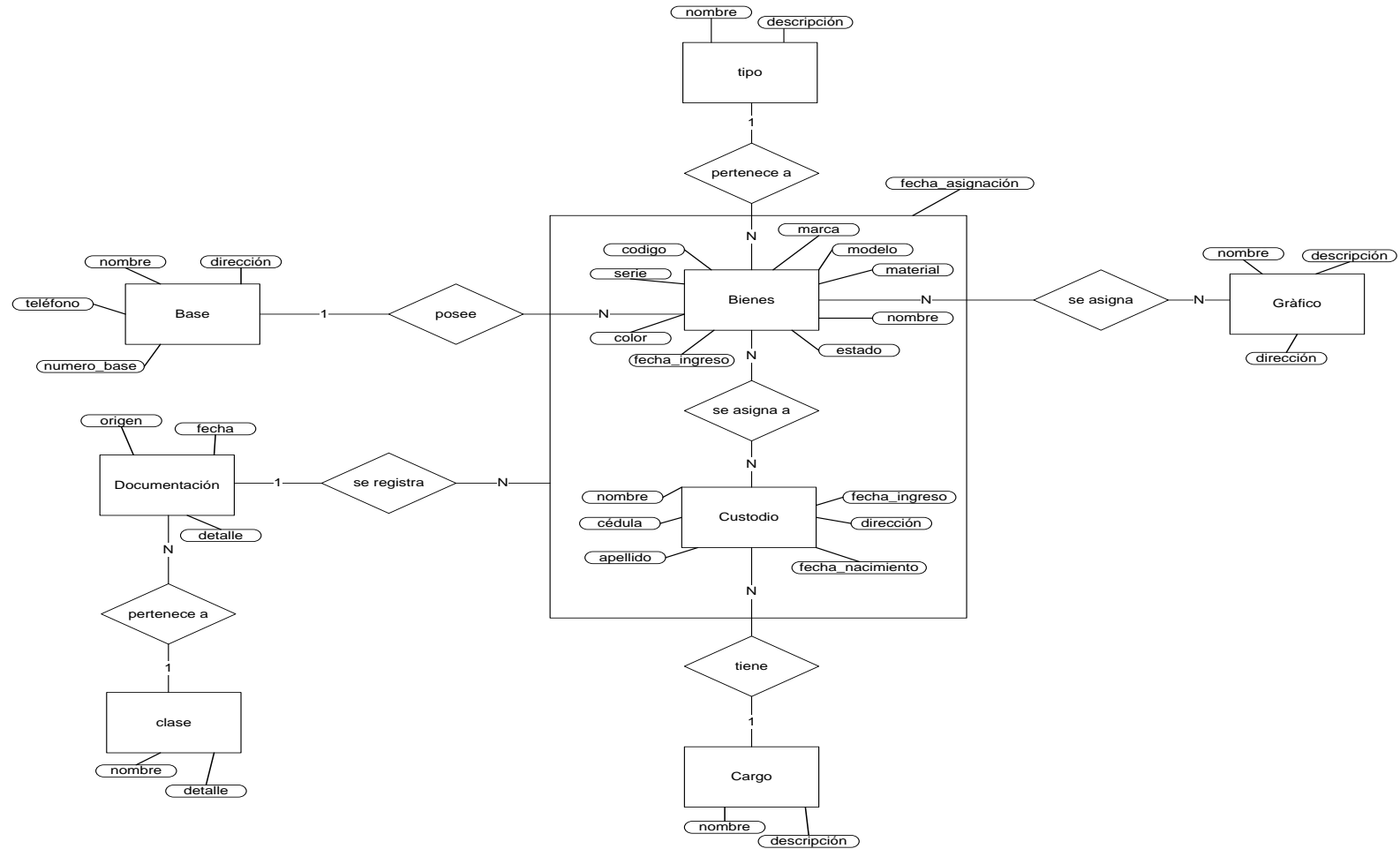


Figura 18. Diseño conceptual del sitio web

Fuente: Elena Mariño



### 3.3.2 Modelo relacional

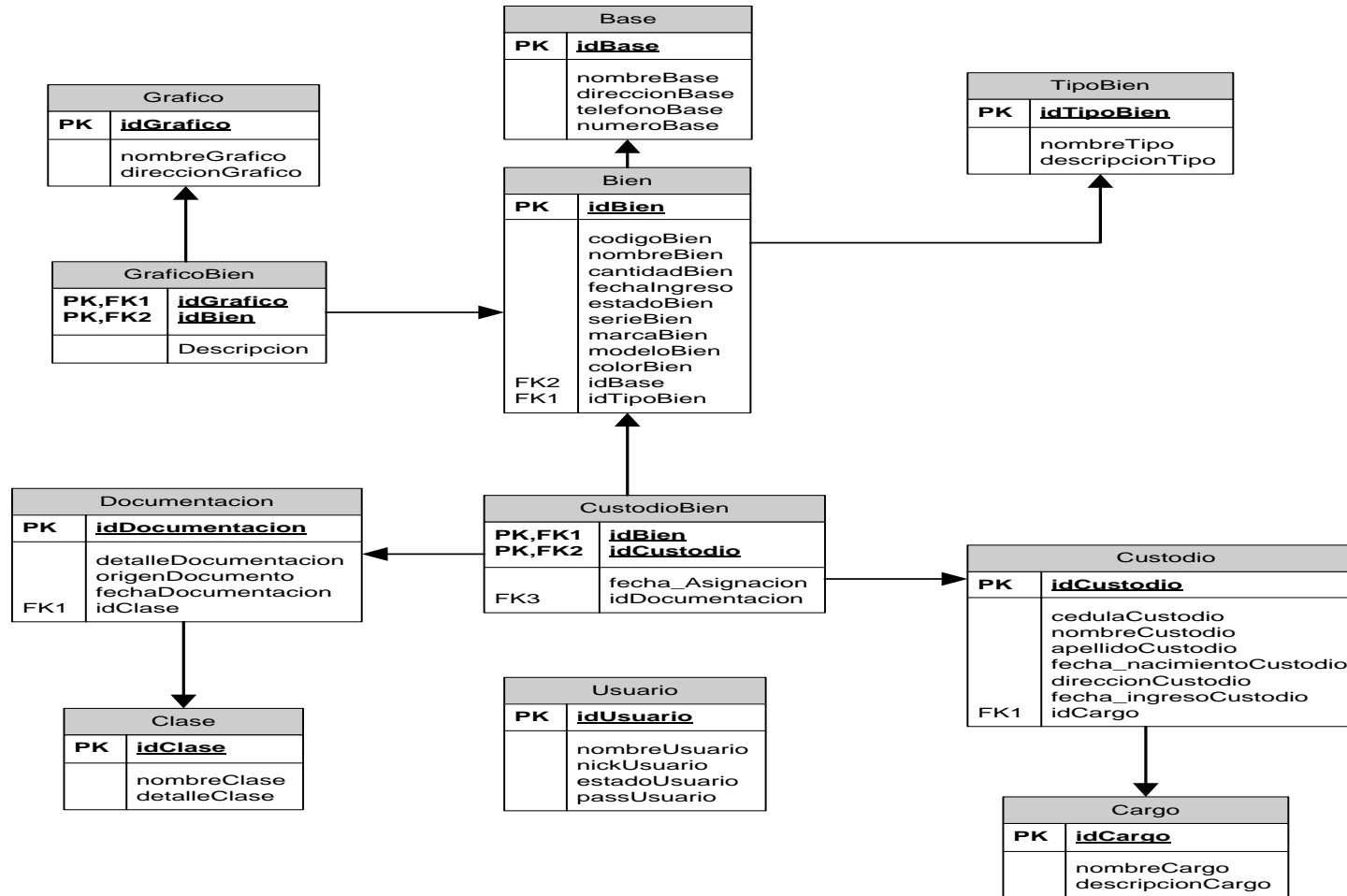


Figura 19. Modelo relacional del sitio web

Fuente: Elena Mariño

### 3.3.3 Diccionario de datos

El diccionario de datos describen las características lógicas de cada las características generales del sitio web y de las tablas necesarias para el desarrollo de la base de datos.

Tabla 2. Diccionario de datos referenciales del sitio web

| Diccionario de Datos Referencial |              |        |             |                 |   |
|----------------------------------|--------------|--------|-------------|-----------------|---|
| Denominación de Uso              | Tipo de Dato | Tamaño | V. Defecto  | Restricciones   | Descripción   |
| Nombres                          | String       | 50     |             | Alfanumérico    | Se usa para almacenar los nombres de personas, bienes o instituciones             |
| Direcciones                      | String       | 200    |             | Alfanumérico    | Se usa para registrar las direcciones domiciliarias o ubicaciones de los archivos |
| Teléfonos                        | String       | 16     |             | [0..9],[(..)]   | Se usa para registrar los números de teléfono                                     |
| Números                          | String       | 3      |             | [0..9]          | Se usa para registrar números en formato texto                                    |
| Descripciones                    | String       | 500    |             | Alfanumérico    | Se usa para detallar las características de un bien o recurso                     |
| Códigos                          | String       | 100    |             | [0..9],[.]      | Se usa para registrar los codigos asignados a los bienes                          |
| Cantidades                       | Integer      |        | 0           | Números Enteros | Se usa para registrar las cantidades de bienes existentes                         |
| Fechas                           | Date         |        |             |                 | Se usa para registrar las fechas  |
| Estados                          | String       | 15     |             | Alfanumérico    | Se usa para registrar el estado de los bienes o recursos                          |
| Características                  | String       | 100    |             | Alfanumérico    | Se usa para registrar las diferentes características de los bienes                |
| Colores                          | String       | 20     | "sin color" | Alfanumérico    | Se usa para registrar los colores de los bienes                                   |
| Identificadores                  | Integer      |        |             | Autoincremental | Se usa para identificar cada uno de los registros de las tablas                   |
| Nick                             | String       | 20     |             | Alfanumérico    | Se usa para identificar a los usuarios del sistema                                |
| Passwords                        | String       | 50     |             | Alfanumérico    | Se usa para restringir el acceso al sistema                                       |
| Cédulas                          | String       | 13     |             | [0..9]          | Se usa para registrar las cédulas de identidad de los custodios                   |

Fuente: Elena Mariño

Tabla 3. Diccionario de datos tabla base

| Tabla | Campo         | Tipo    | Tamaño       | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|-------|---------------|---------|--------------|------------|----------------|-----------------|-----------|
| Base  | idBase        | Integer | autodefinido |            | Si             | Si              | Si        |
| Base  | nombreBase    | String  | 50           |            |                |                 | Si        |
| Base  | direccionBase | String  | 200          |            |                |                 | No        |
| Base  | telefonoBase  | String  | 16           |            |                |                 | No        |
| Base  | numeroBase    | String  | 3            |            |                |                 | No        |

Fuente: Elena Mariño

Tabla 4. Diccionario de datos tabla tipoBien

| Tabla    | Campo           | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|----------|-----------------|---------|--------|------------|----------------|-----------------|-----------|
| TipoBien | idTipoBien      | Integer |        |            | Si             | Si              | Si        |
| TipoBien | nombreTipo      | String  | 50     |            |                |                 | Si        |
| TipoBien | descripcionTipo | String  | 500    |            |                |                 | Si        |

Fuente: Elena Mariño

Tabla 5. Diccionario de datos tabla Gráfico

| Tabla   | Campo            | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|---------|------------------|---------|--------|------------|----------------|-----------------|-----------|
| Grafico | idGrafico        | Integer |        |            | Si             | Si              | Si        |
| Grafico | nombreGrafico    | String  | 50     |            |                |                 | Si        |
| Grafico | direccionGrafico | String  | 200    |            |                |                 | Si        |

Fuente: Elena Mariño

Tabla 6. Diccionario de datos tabla Bien

| Tabla | Campo        | Tipo    | Tamaño | V. Defecto  | Clave Primaria | Autoincremental | Requerido |
|-------|--------------|---------|--------|-------------|----------------|-----------------|-----------|
| Bien  | idBien       | Integer |        |             | Si             | Si              | Si        |
| Bien  | codigoBien   | String  | 100    |             |                |                 | Si        |
| Bien  | nombreBien   | String  | 50     |             |                |                 | Si        |
| Bien  | cantidadBien | Integer |        | 0           |                |                 | No        |
| Bien  | fechaIngreso | Date    |        |             |                |                 | No        |
| Bien  | estadoBien   | String  | 15     |             |                |                 | No        |
| Bien  | serieBien    | String  | 100    | "s/n"       |                |                 | No        |
| Bien  | marcaBien    | String  | 100    | "sin marca" |                |                 | No        |
| Bien  | modeloBien   | String  | 100    | "genérico"  |                |                 | No        |
| Bien  | colorBien    | String  | 20     | "sin color" |                |                 | No        |
| Bien  | idBase       | Integer |        |             |                |                 | Si        |
| Bien  | idTipoBien   | Integer |        |             |                |                 | Si        |

Fuente: Elena Mariño

Tabla 7. Diccionario de datos tabla Documentación

| Tabla         | Campo                | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|---------------|----------------------|---------|--------|------------|----------------|-----------------|-----------|
| Documentacion | idDocumentacion      | Integer |        |            | Si             | Si              | Si        |
| Documentacion | detalleDocumentacion | 500     |        |            |                |                 | Si        |
| Documentacion | origenDocumento      | 100     |        |            |                |                 | No        |
| Documentacion | fechaDocumentacion   | Date    |        |            |                |                 | No        |
| Documentacion | idClase              | Integer |        |            |                |                 | Si        |

Fuente: Elena Mariño

Tabla 8. Diccionario de datos tabla CustodioBien

| Tabla        | Campo           | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|--------------|-----------------|---------|--------|------------|----------------|-----------------|-----------|
| CustodioBien | idBien          | Integer |        |            | Si             | Si              | Si        |
| CustodioBien | IdCustodio      | Integer |        |            | Si             | Si              | Si        |
| CustodioBien | fechaAsignacion | Date    |        |            |                |                 | Si        |
| CustodioBien | idDocumentacion | Integer |        |            |                |                 | Si        |

Fuente: Elena Mariño

Tabla 9. Diccionario de datos tabla Clase

| Tabla | Campo        | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|-------|--------------|---------|--------|------------|----------------|-----------------|-----------|
| Clase | idClase      | Integer |        |            | Si             | Si              | Si        |
| Clase | nombreClase  | String  | 50     |            |                |                 | Si        |
| Clase | detalleClase | String  | 500    |            |                |                 | Si        |

Fuente: Elena Mariño

Tabla 10. Diccionario de datos tabla Custodio

| Tabla    | Campo                    | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|----------|--------------------------|---------|--------|------------|----------------|-----------------|-----------|
| Custodio | idCustodio               | Integer |        |            | Si             | Si              | Si        |
| Custodio | cedulaCustodio           | String  | 13     |            |                |                 | Si        |
| Custodio | nombreCustodio           | String  | 50     |            |                |                 | Si        |
| Custodio | apellidoCustodio         | String  | 50     |            |                |                 | Si        |
| Custodio | fecha_nacimientoCustodio | Date    |        |            |                |                 | No        |
| Custodio | DireccionCustodio        | String  | 200    |            |                |                 | No        |
| Custodio | fecha_ingresoCustodio    | Date    |        |            |                |                 | No        |
| Custodio | idCargo                  | Integer |        |            |                |                 | Si        |

Fuente: Elena Mariño

Tabla 11. Diccionario de datos tablaCargo

| Tabla | Campo            | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|-------|------------------|---------|--------|------------|----------------|-----------------|-----------|
| Cargo | IdCargo          | Integer |        |            | Si             | Si              | Si        |
| Cargo | NombreCargo      | String  | 50     |            |                |                 | Si        |
| Cargo | descripcionCargo | String  | 500    |            |                |                 | No        |

Fuente: Elena Mariño

Tabla 12. Diccionario de datos Usuario

| Tabla   | Campo         | Tipo    | Tamaño | V. Defecto | Clave Primaria | Autoincremental | Requerido |
|---------|---------------|---------|--------|------------|----------------|-----------------|-----------|
| Usuario | idUsuario     | Integer |        |            | Si             | Si              | Si        |
| Usuario | nombreUsuario | String  | 50     |            |                |                 | Si        |
| Usuario | nickUsuario   | String  | 20     |            |                |                 | Si        |
| Usuario | estadoUsuario | String  | 15     |            |                |                 | Si        |
| Usuario | passUsuario   | String  | 50     |            |                |                 | Si        |

Fuente: Elena Mariño

### 3.3.4 Diseño de interfaces

Representan la presentación que tendrá el sitio web de una manera tentativa y que son dispuestas por el desarrollador.

**Página Principal:** En esta pantalla se muestran las opciones con las que contará el usuario

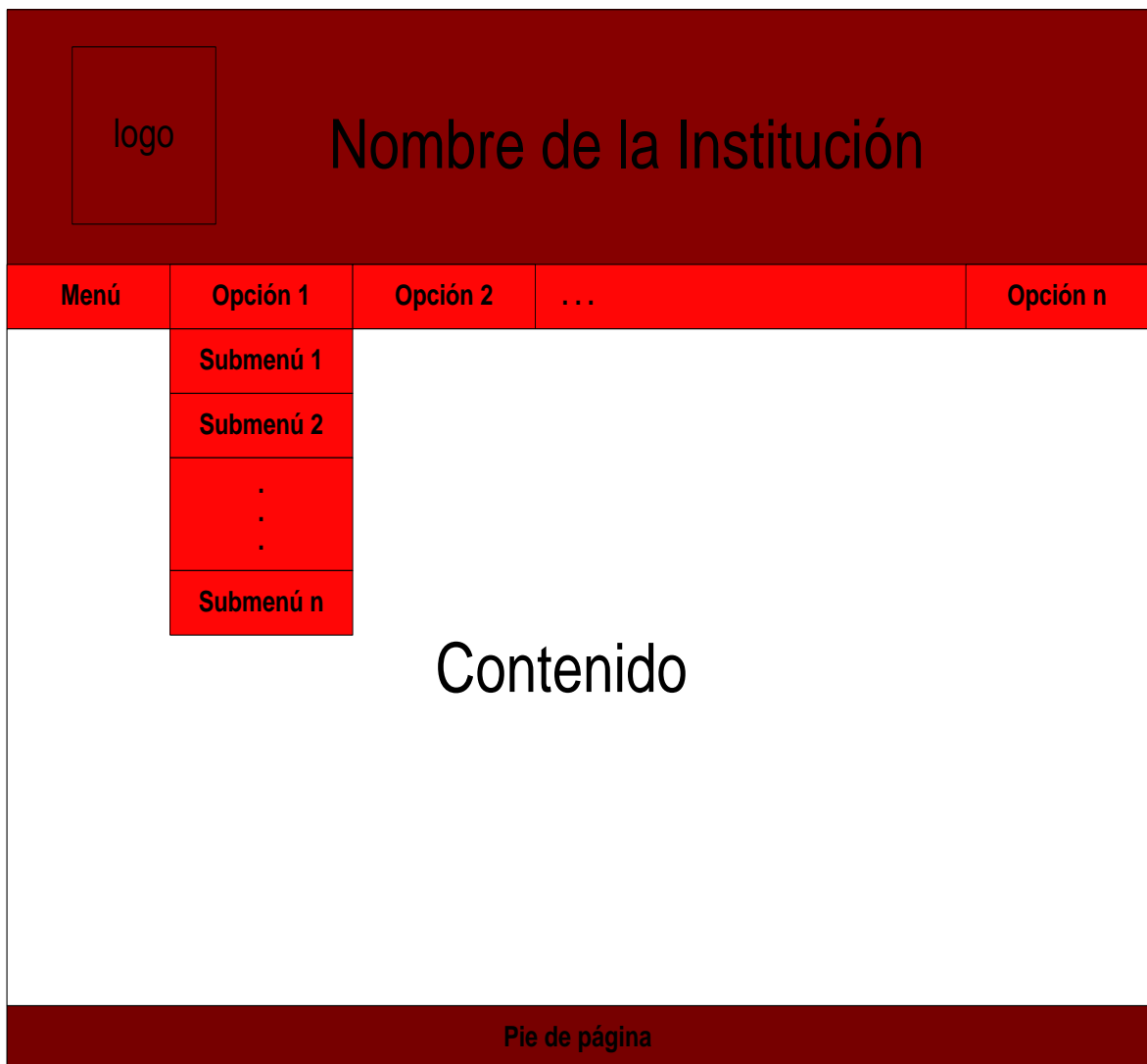


Figura 20. Pantalla de inicio

Fuente: Elena Mariño

**Acceso al sistema:** Muestra las opciones de seguridad para el ingreso al sistema



Logo

Ingresar datos de acceso


Usuario:

Password:

Figura 21. Acceso al sistema

Fuente: Elena Mariño

**Ingreso de datos:** Indica las opciones y secuencia de instrucciones que se deben seguir para el ingreso de la información por parte del usuario



Título de la Página

Elemento 1:

Elemento 2:

⋮

Elemento n:

Figura 22. Ingreso de datos

Fuente: Elena Mariño

**Consulta de datos:** Se muestran los campos que se requieren para la consulta de datos en el sitio web.

### Título de la Página

Campo a Buscar:

| Nombre Campo 1    | Nombre Campo 2 | ... | Nombre Campo n | Acción  |
|-------------------|----------------|-----|----------------|---|
| Valor Resultado 1 |                |     |                | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> |
| Valor Resultado 2 |                |     |                | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> |
| ...               |                |     |                |   |
| Valor Resultado n |                |     |                | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> |

Figura 23. Consulta de datos

Fuente: Elena Mariño

**Actualización de datos:** detalla los campos necesarios para la actualización de datos.

### Título de la Página

**Elemento 1:**

**Elemento 2:**

...

**Elemento n:**

Figura 24. Actualización de datos

Fuente: Elena Mariño



**Asignación de bienes al custodio:** Determina aquellos datos necesarios para la entrega de bienes a custodios

### Título de la Página

Nombre del custodio:  ▼

Custodio 1  
Custodio 2  
.  
.  
Custodio n

Nombre del bien:  ▼ Cantidad:

Bien 1  
Bien 2  
.  
.  
Bien n

| Campo 1           | Campo 2 | ... | Campo n | Acción                                |
|-------------------|---------|-----|---------|---------------------------------------|
| Valor Resultado 1 |         |     |         | <input type="button" value="Remove"/> |
| Valor Resultado 2 |         |     |         | <input type="button" value="Remove"/> |
| Valor Resultado n |         |     |         | <input type="button" value="Remove"/> |

Figura 25. Asignación de los bienes al custodio

Fuente: Elena Mariño

## **CAPÍTULO IV**

### **4 IMPLEMENTACIÓN DEL SITIO WEB**

## 4.1 CONFIGURACIÓN DE LAS HERRAMIENTAS DE DESARROLLO

Para realizar el Sitio Web de Bomberos Mocha, se utilizó el entorno de desarrollo Netbeans 8.2, el cual se puede descargar desde su página web.

Con la descarga del programa se procede a abrir el archivo y a su ejecución

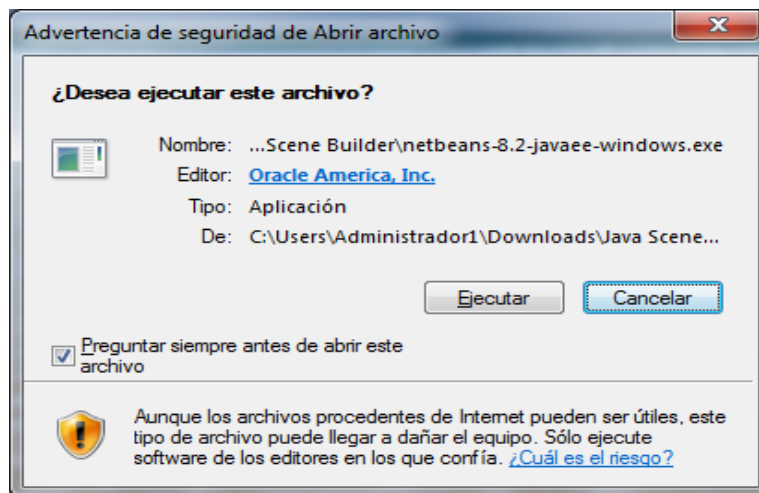


Figura 26. Ejecución de Netbeans

Fuente: Elena Mariño

Con la ejecución del programa se despliega una pantalla que brinda las opciones por defecto.

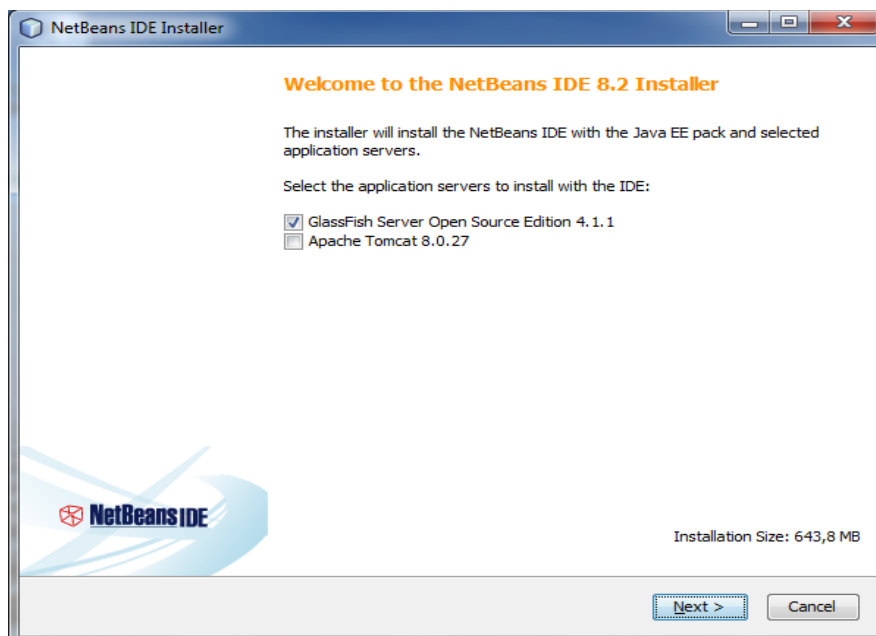


Figura 27. Instalador de Netbeans

Fuente: Elena Mariño

Ninguna de las opciones serán usadas en el proyecto, por lo cual las desactivamos y se procede a autorizar el contrato de uso del Software.

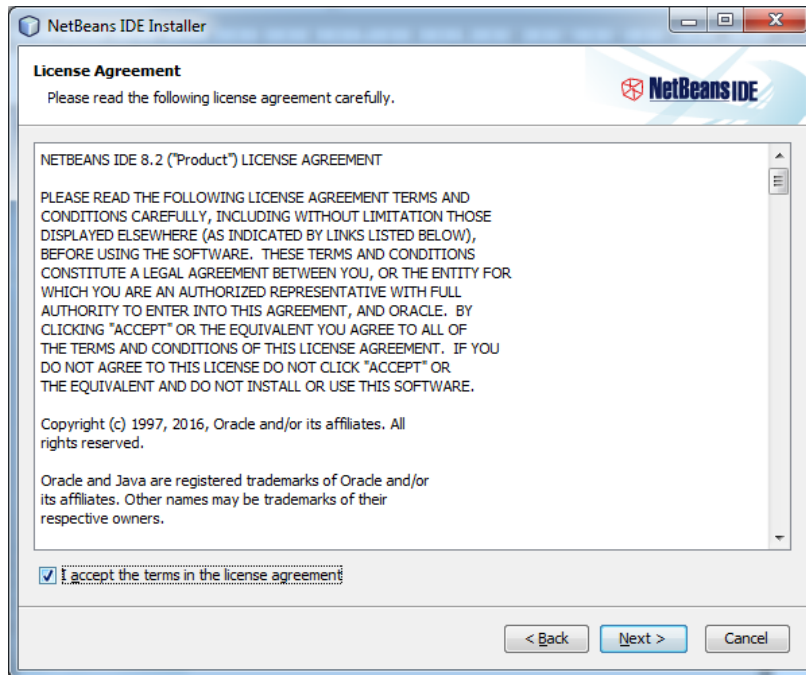


Figura 28. Contrato de uso de Netbeans

Fuente: Elena Mariño

En la siguiente pantalla se selecciona la carpeta donde almacenar los archivos necesarios para la ejecución del programa.

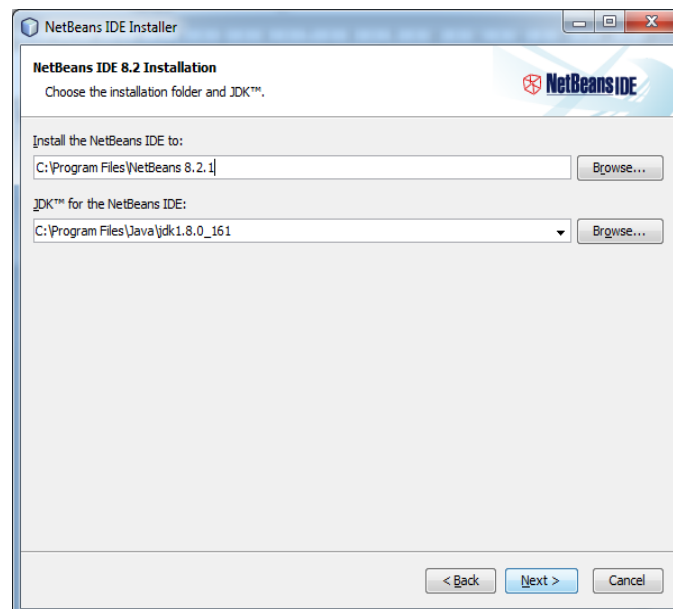


Figura 29. Ubicación de Netbeans en una carpeta

Fuente: Elena Mariño

A continuación se presenta el resumen de las opciones elegidas

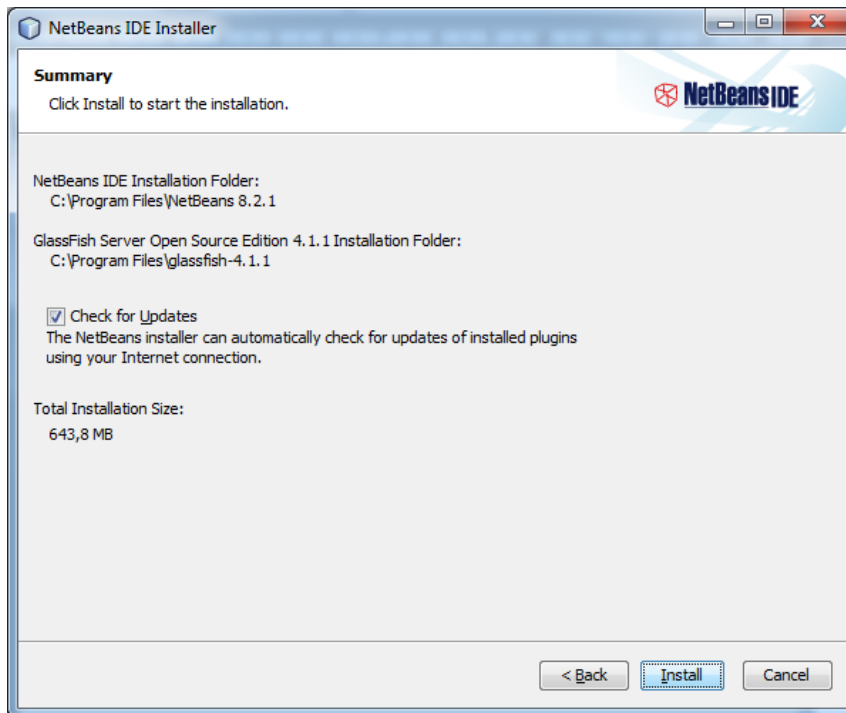


Figura 30. Resumen de opciones Netbeans

Fuente: Elena Mariño

Con todos los datos descritos, se encuentra una pantalla que permite la instalación del programa

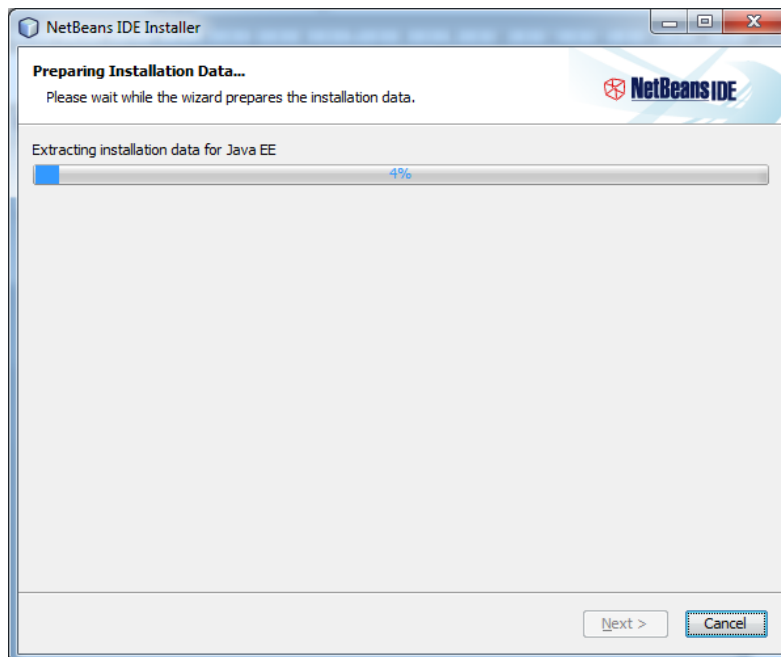


Figura 31. Instalación de Netbeans

Fuente: Elena Mariño

Una vez finalizado el proceso de instalación se abre el entorno de desarrollo NetBeans 8.2



Figura 32. Netbeans 8.2

Fuente: Elena Mariño

Además, es necesario utilizar el Servidor Web XAMPP, requiriendo de igual forma la descarga desde la página web de este servidor.

Se procede después de la descarga a su respectiva ejecución

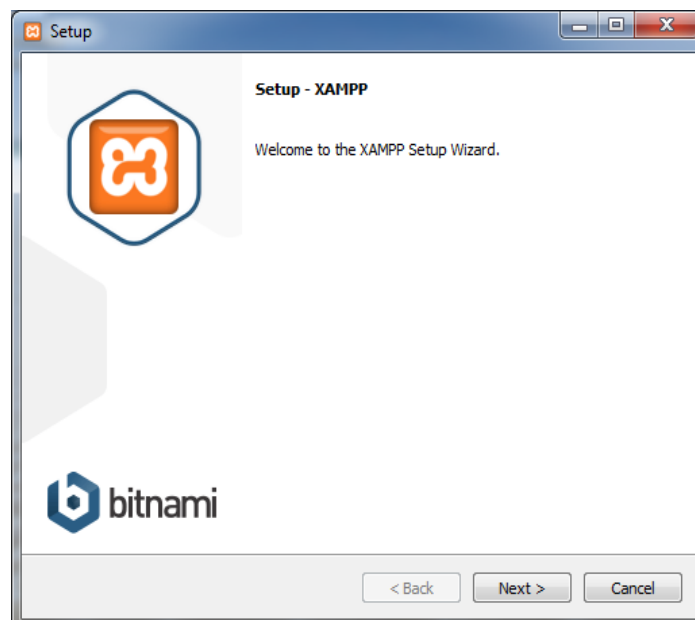


Figura 33. Ejecución de XAMPP

Fuente: Elena Mariño

Avanzando a la siguiente página se eligen los componentes necesarios para la instalación

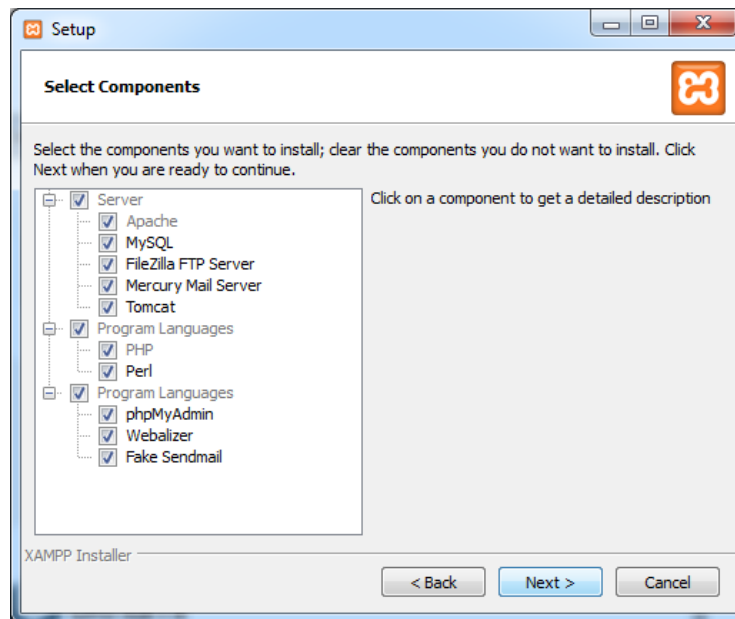


Figura 34. Elección de componentes en XAMPP

Fuente: Elena Mariño

Posteriormente solicita para la correcta instalación que se ubique en una carpeta para que puedan ser archivados los datos en el computador

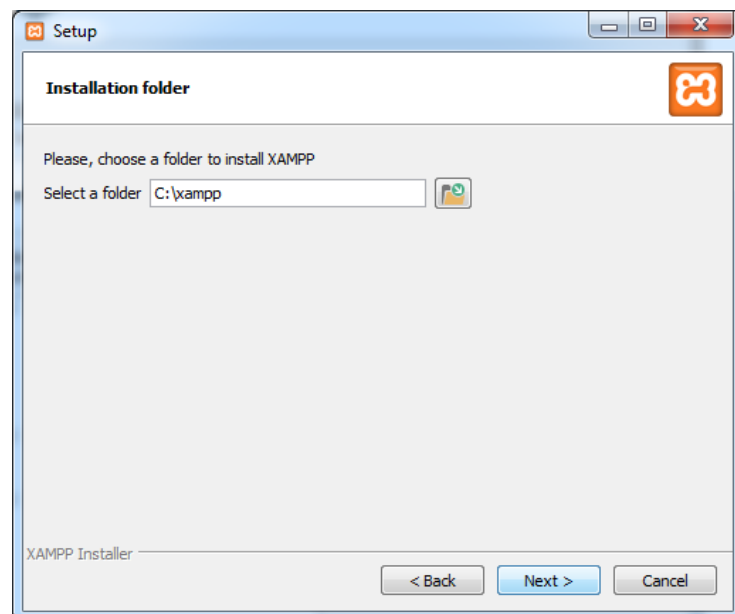


Figura 35. Ubicación en una carpeta el programa XAMPP

Fuente: Elena Mariño

A continuación aparece la pantalla de confirmación, para de esta manera dar inicio a la instalación

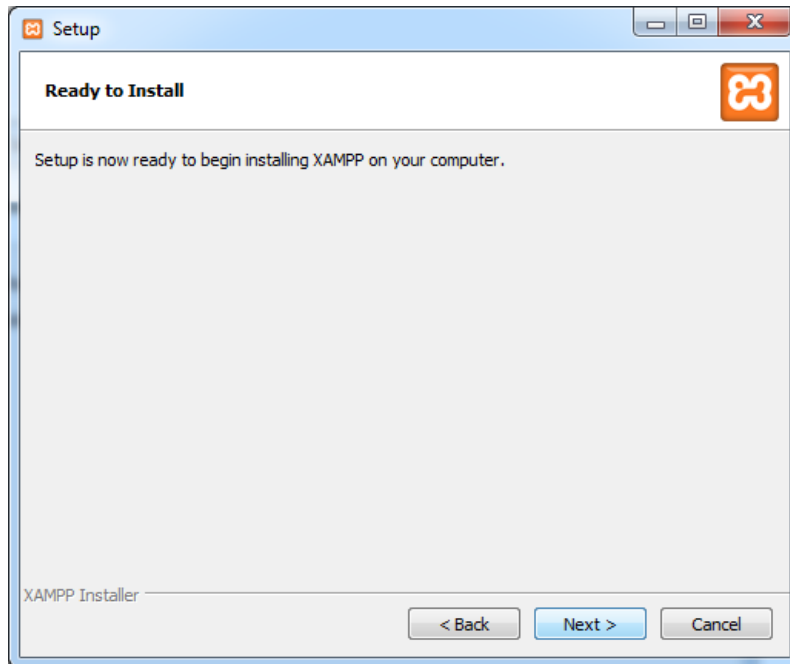


Figura 36. Pantalla de confirmación de XAMPP

Fuente: Elena Mariño

Una vez que finalice la instalación, se ejecuta XAMPP Control Panel, que aparece en el menú de Inicio del programa como se muestra a continuación

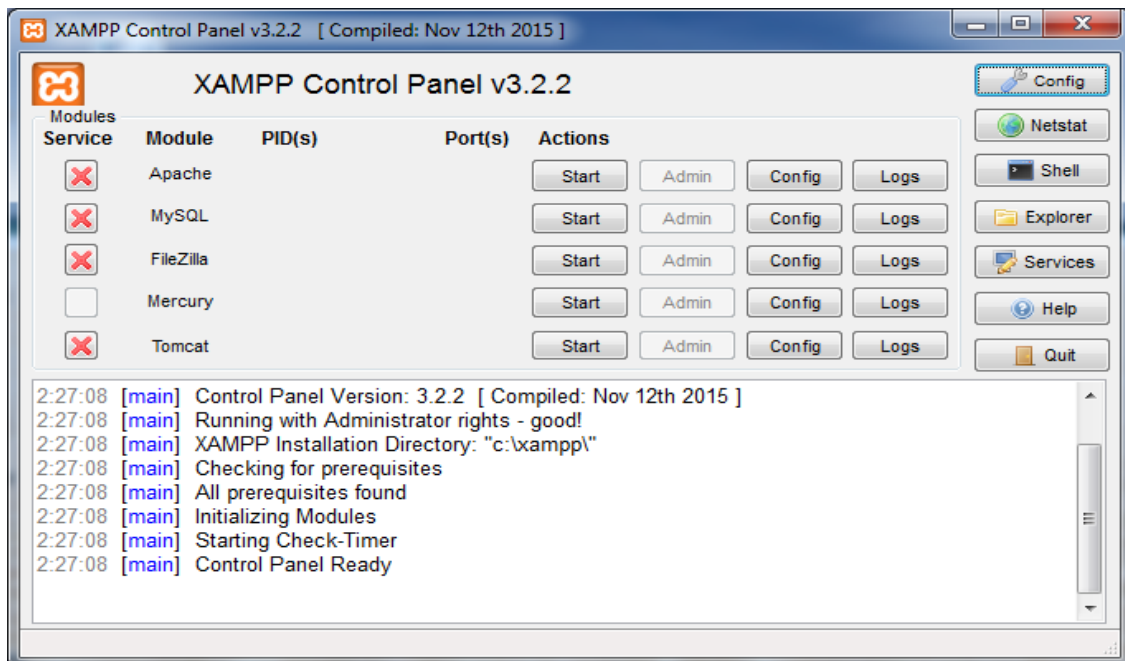


Figura 37. Ejecución de XAMPP Control Panel

Fuente: Elena Mariño



Aparecerá la pantalla con la lista de módulos instalados, verificando que se encuentran listos cuando se escoge la opción en color verde

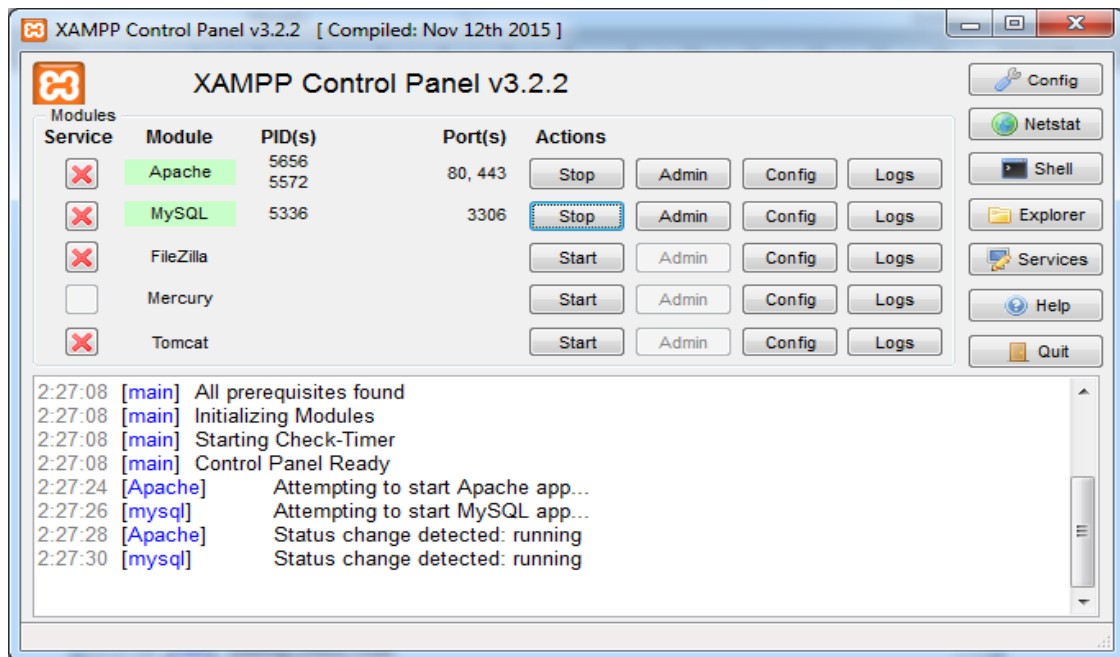


Figura 38. Elección de módulos

Fuente: Elena Mariño

A continuación se ingresa a la siguiente dirección <http://localhost> para verificar el servidor web está en modo activo



Figura 39. Verificación del servidor en modo activo

Fuente: Elena Mariño

Además, con la siguiente dirección <http://localhost/phpmyadmin> se podrá trabajar con el servidor de base de datos MySQL

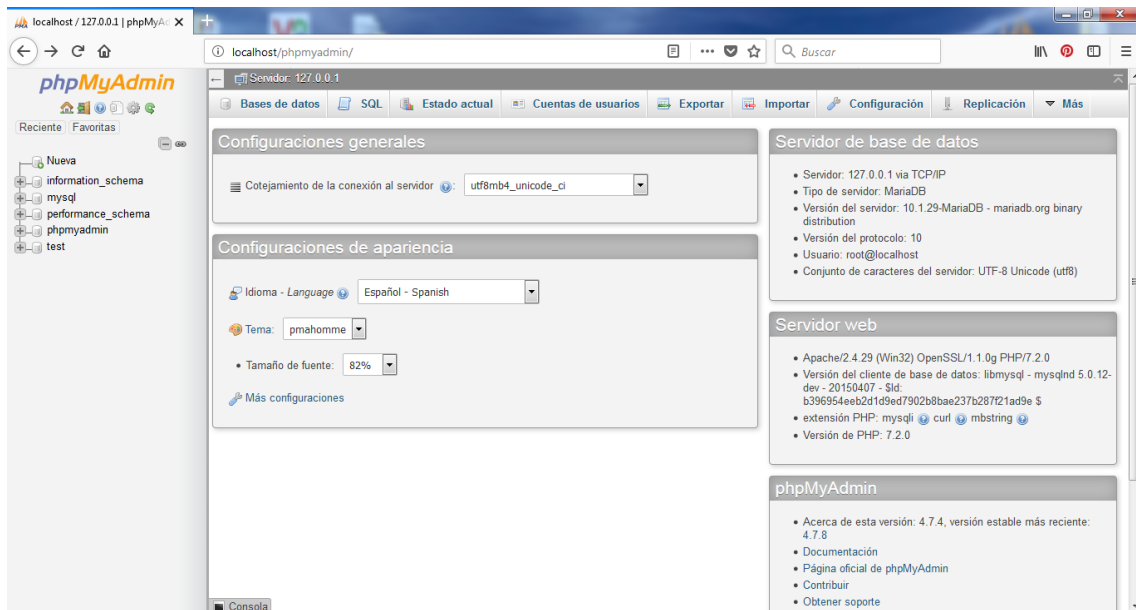


Figura 40. PhpMyAdmin

Fuente: Elena Mariño

Una vez en phpMyAdmin se elige la opción nuevo que es la que permite el ingreso de la nombre de la base de datos que desea crear.

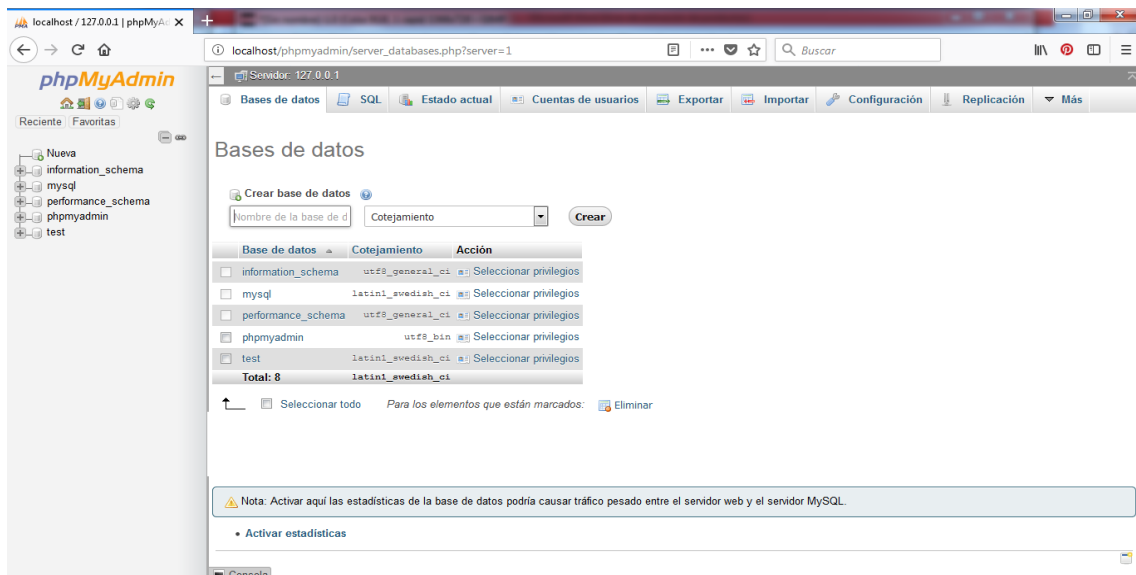


Figura 41. Creación de una nueva base de datos

Fuente: Elena Mariño

Una vez creada la Base de Datos, seleccionamos la Opción Nuevo, dentro de la Base de Datos creada y se llenan los campos que se encuentran en blanco

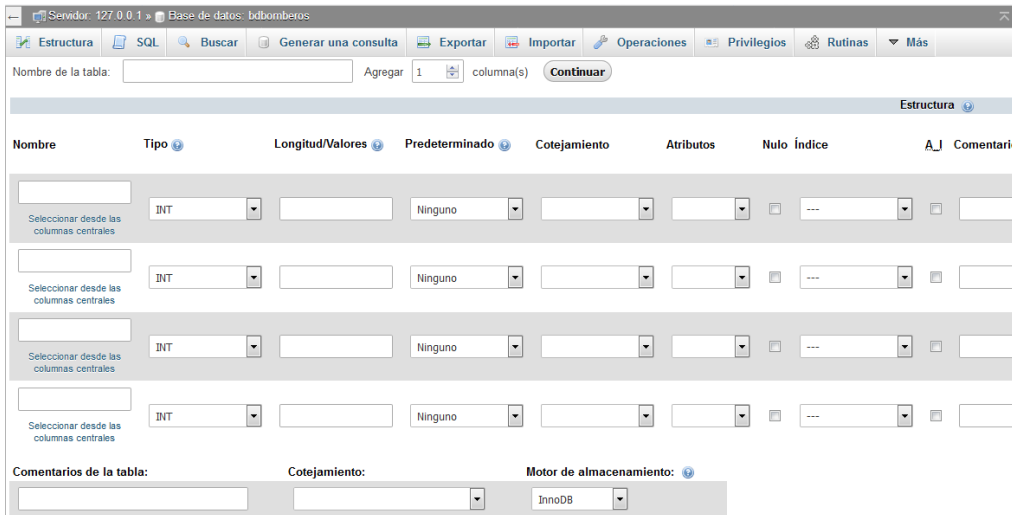


Figura 42. Ingreso de los campos para la base de datos creada

Fuente: Elena Mariño

Por tanto esta herramienta permitió la creación de todas las tablas necesarias en la base de datos correspondiente y requerida para el sitio web.

## 4.2 ARQUITECTURA DEL SISTEMA

Una Arquitectura cliente servidor es considerada como el mayor soporte en una comunicación por redes. La misma se describe a continuación:

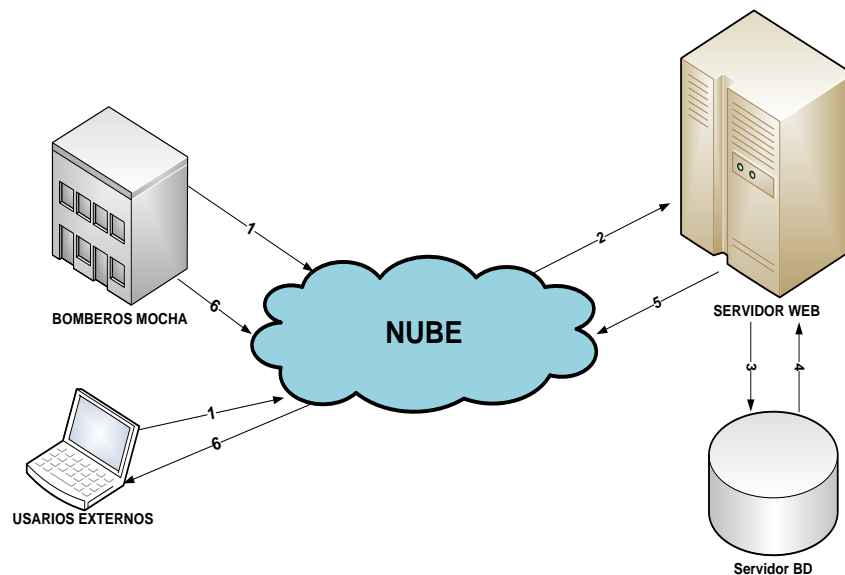


Figura 43. Arquitectura del sitio web

Fuente: Elena Mariño

1. Los usuarios realizan la solicitud aplicación web, de un requerimiento hacia la nube. Esta ingresa a la nube como una dirección TCP/IP.
2. En esta se transforma en un Post Get el cual es enviado al Servidor Web.
3. En este servidor transforma esta solicitud en un lenguaje que el Servidor de BD conozca y lo envía
4. El servidor web procesa la información del Query recibido, lo analiza y regresa un Dataset con la información requerida.
5. El Servidor Web analiza la información y lo transforma en un ambiente que el usuario conoce (HTML) y es devuelta a la nube.
6. La información regresa al usuario en forma de una dirección TCP/IP pero este es codificado y lo muestra en forma de página HTML

Todo este proceso el usuario no lo visualiza ya que las respuestas a estos procesos son inmediatas, y lo único que el usuario observa son aplicaciones WEB.

### **4.3 IMPLEMENTACIÓN DEL SISTEMA**

La implementación no es más que la puesta en práctica a los requerimientos que el usuario proporciona al desarrollador, para poder resolver A o B problema que este desea automatizar procesos para hacerlos más eficientes.

#### **4.3.1 Definición de Módulos**

Este tipo de Módulos, están relacionados en forma cíclica, la misma que define una seguridad confiable, y que están relacionados entre sí, con la debida información o autenticación para los anteriores y siguientes.

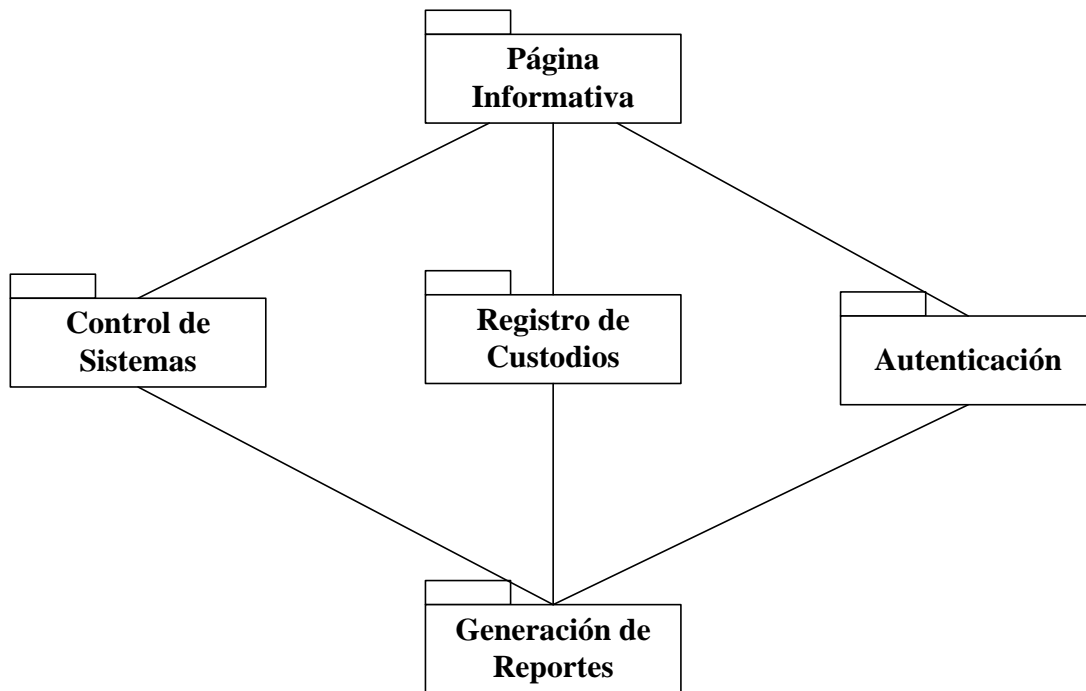


Figura 44. Definición de módulos del sistema web

Fuente: Elena Mariño

#### 4.3.2 Desarrollo de la aplicación implementación

Mediante las estepas de diseño y de análisis se procedió a la instalación del sitio web en el Cuerpo de Bomberos Mocha, el mismo se encuentra especificado mediante el manual de usuario dispuesto en el Anexo 1.

#### 4.3.3 Implementación de la aplicación

Para el máximo desempeño del sitio web se requiere de una serie de actividades para su configuración las cuales se detallan en el manual técnico especificado en el Anexo 2.

#### 4.4 PRUEBAS DE LA APLICACIÓN

Existe un sinnúmero de pruebas que se puede realizar al sitio web dentro de las cuales se detallan las necesarias:

- **Navegabilidad:** Los links cumplen con el estándar definido, no existe errores en estos que produzcan errores en el sistemas.

- **Funcionalidad:** El funcionamiento del sistema cumple con los requerimientos funcionales expuestos para dar solución a la petición de los clientes.

Tabla 13. Pruebas de los requerimientos Funcionales

| <b>Funcionalidad</b>  | <b>SI</b> | <b>NO</b> |
|---|-----------|-----------|
| <b>Presentar Página Informativa</b>   | <b>X</b>  |           |
| <b>Ingresar, Listar, Actualizar y Dar de Baja a los Bienes al Registro</b>    | <b>X</b>  |           |
| <b>Ingresar, Listar, Actualizar y Dar de Baja a los Custodios del sistema</b> | <b>X</b>  |           |
| <b>Ingresar, Listar, Actualizar y Dar de Baja a los Usuarios del sistema</b>  | <b>X</b>  |           |
| <b>Asignar Custodio al Bien y Listar Bienes del Custodio</b>                  | <b>X</b>  |           |
| <b>Permitir Acceso a Usuarios</b>   | <b>X</b>  |           |
| <b>Lista y carga de Archivos Leyes</b>  | <b>X</b>  |           |

Fuente: Elena Mariño

- **Seguridad:** Realiza un control integral en fallos, de carácter, ortográfico como de seguridad al ingreso de usuarios, así como los requerimientos no funcionales.

Tabla 14. Pruebas de los requerimientos no funcionales

| <b>Descripción</b> | <b>SI</b> | <b>NO</b> |
|--------------------|-----------|-----------|
| Disponibilidad     | X         |           |
| Escalabilidad      | X         |           |
| Seguridad          | X         |           |
| Eficiencia         | X         |           |

Fuente: Elena Mariño

## 4.5 CAPACITACIÓN AL PERSONAL

- **Personal**

**Impartidor:** Elena Mariño desarrolladora del sistema

**Destinatarios:** Personal administrativo, serán los encargados de obtener los conocimientos necesarios para un correcto manejo del sistema.

- **Herramientas**

Se utilizara presentaciones, ejemplos y demostraciones del sistema, la misma que tendrá como apoyo:

**Dispositivos electrónicos**

- Computador
- Proyector de pantalla

**Herramientas tecnológicas**

- Power Point (diapositivas)
- Firefox
- Internet

- **Tiempo**

**Tiempo estimado:** 1 mes

## 4.6 MANTENIMIENTO DE LA APLICACIÓN

El manteniendo de la aplicación fue definido por los actores del sistema, dando así tiempos y precisando los nuevos requerimientos de los usuarios.

Una vez implementado el sistema, sus actualizaciones se basaron en los requerimientos que a futuro solicite el cuerpo de bomberos Mocha, esto dependerá de los usuarios, los mismos que permitirán dar tiempos de mantenimiento.

Dando paso así a un mejor control de bienes y seguridad de la aplicación mediante el resguardo de la información.

## **CAPÍTULO V**

### **5 CONCLUSIONES Y RECOMENDACIONES**



## 5.1 CONCLUSIONES

- El uso de PHP orientado a objetos y el gestor de base de datos MySQL facilita el desarrollo de un sitio web de forma dinámica, por su bajo costo y su curva de aprendizaje baja.
- La funcionalidad del sitio web destinado para el control de materiales y equipos del Cuerpo de Bomberos Mocha permite el registro de dichos bienes de manera automatizada, segura y de fácil acceso. Logrando con ello la optimización de tiempo y de los recursos de la institución.
- La implementación del sitio Web permitió plasmar todo el análisis realizado, obteniendo así un mejor control de los materiales y equipos mediante la automatización de los procesos manuales que realiza en la actualidad el Cuerpo de Bomberos Mocha para el registro de los bienes.
- Mediante la utilización de la herramienta de administración phpMyAdmin se creó la base de datos de acuerdo a los requerimientos del Cuerpo de Bomberos Mocha y el desarrollo de la aplicación se lo realizó en el lenguaje de programación Php orientado a objetos, logrando de esta manera que todo el proceso de carga de información sea transparente para el usuario

## 5.2 RECOMENDACIONES

- Es recomendable la utilización del lenguaje de programación PHP orientado a objetos y MySQL basados en software libre a fin de minimizar los costos de desarrollo de los sitios web.
- Se recomienda al Cuerpo de Bomberos Mocha facilitar información pertinente y veraz con la finalidad de que los proyectos se puedan desarrollar para bien de la entidad sean más eficaces.
- Se debe realizar la actualización y el mantenimiento de la información contenida en la base de datos, además de realizar copias de seguridad periódicas con la finalidad de garantizar la veracidad de la información.
- Se sugiere, realizar pruebas trimestrales al sitio web permitiendo al administrador y al técnico tener un mayor control de la funcionalidad lo que permitirá prevenir fallos por manejos inadecuados del sistema.

## BIBLIOGRAFÍA

- Aula Informativa. (30 de Junio de 2017). Obtenido de <http://blog.aulaformativa.com/definicion-usos-ventajas-lenguaje-css3/>
- Baluhart.NET. (29 de Noviembre de 2010). *Ocho expertos comentan los Pros y Contras de programar con PHP*. Obtenido de <http://www.baluart.net/articulo/ocho-expertos-comentan-los-pros-y-contras-de-programar-con-php>
- Barzanallana, R. (2 de Agosto de 2009). Obtenido de [http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Prog\\_Obj01.html](http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/Prog_Obj01.html)
- Delgado, H. (17 de Mayo de 2017). *Tipos de Datos de SQL en MySQL*. Obtenido de <https://disenowebakus.net/tipos-de-datos-mysql.php>
- Denner, C. (14 de Junio de 2015). *10 poderosos beneficios de tener un buen sitio web* . Obtenido de <https://www.infopresario.com/10-beneficios-tener-un-sitio-web/>
- Estudio Seijo. (23 de Julio de 2017). *Tipos de Sitios Web*. Obtenido de <http://www.estudioseijo.com/noticias/tipos-de-sitios-web.htm>
- Gaitan, F. (12 de Diciembre de 2016). *Php orientado a objetos, parte 2: Método constructor y destructor – Fernando Gaitán*. Obtenido de <https://fernandogaitan.com.ar/php-orientado-a-objetos-parte-2-metodo-constructor-y-destructor/>
- García, A. (11 de Noviembre de 2016). *Etapas de la Creacionde un Sitio Web*. Obtenido de <http://www.redalyc.org/html/161/16114408/>
- Gonzalez, J. (26 de Abril de 2005). *Desarrollo Web con PHP y MySql*. Obtenido de <http://educagratis.cl/moodle/mod/resource/view.php?id=4105&usg=AOvVaw2V9ryQz7Xw2ijt2jvVZOKD>
- Klenk. (5 de Mayo de 2016). *Diseño de Materiales multimedia Web 2.0*. Obtenido de <http://klenk.com.ar/Estructura%20de%20un%20sitio%20web.pdf>
- Morgan, C. (10 de Julio de 2004). *PHP5 and MySQL Bible*. Obtenido de [https://www.amazon.com/PHP5-MySQL-Bible-Tim-Converse/dp/0764557467&usg=AOvVaw3jPzqnw\\_eipJF\\_zU5\\_wTIy](https://www.amazon.com/PHP5-MySQL-Bible-Tim-Converse/dp/0764557467&usg=AOvVaw3jPzqnw_eipJF_zU5_wTIy)
- MySQL. (13 de Abril de 2010). Obtenido de <http://ftp.tcrc.edu.tw/MySQL/doc/refman/5.0/es/features.htm>
- Oracle-MySQL. (24 de Mayo de 2014). Obtenido de <https://downloads.mysql.com/docs/refman-5.0-es.pdf>

- Palomo, M. (9 de Julio de 2014). Obtenido de [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)
- Pérez, D. (3 de Julio de 2007). *¿Qué es Javascript?* Obtenido de <http://www.maestrosdelweb.com/que-es-javascript/>
- Pérez, J. (1 de Febrero de 2010). *Definición de Sitio Web*. Obtenido de <https://definicion.de/sitio-web/>
- Red Gráfica Latinoamérica. (11 de Marzo de 2013). *El Lenguaje de Programación PHP*. Obtenido de <http://redgrafica.com/El-lenguaje-de-programacion-PHP>
- Roldan, A. (11 de Octubre de 2017). *Ventajas de la Programación Orientada a Objetos*. Obtenido de [http://www.ciberaula.com/articulo/ventajas\\_poo](http://www.ciberaula.com/articulo/ventajas_poo)
- Solvetic. (17 de Marzo de 2014). *Herramientas y estructura de MYSQL 5*. Obtenido de <https://www.solvetic.com/tutoriales/article/559-herramientas-y-estructura-de-mysql-5/>
- Taller de Base de Datos. (13 de Septiembre de 2015). *TallerBD - - Estructura de MySQL*. Obtenido de <https://tallerbd.wikispaces.com/-+Estructura+de+MySQL>
- UNET. (29 de NOVIEMBRE de 2019). *Programación Orientada a Objetos(POO)*. Obtenido de <https://compu2poo.wordpress.com/2013/11/29/programacion-orientada-a-objetos-poo-definicion-origen-y-caracteristicas/>
- Velázquez, M. (20 de Enero de 2017). *¿Cuáles son las Características de un Sitio Web Confiable?* . Obtenido de <https://colombiadigital.net/actualidad/articulos-informativos/item/9461-cuales-son-las-caracteristicas-de-un-sitio-web-confiable.html>

# **ANEXOS**

## **ANEXO 1. MANUAL DEL USUARIO**

# **SISTEMA DESIGNACIÓN DE BIENES DEL CUERPO DE BOMBEROS MOCHA**

**MANUAL DE USUARIO**



**Versión 1.0**

**Autor: Elena Mariño**

## 1. Introducción

El Sistema Designación de Bienes del Cuerpo de Bomberos Mocha está dirigido para el Cuerpo de Bomberos Mocha. Que permitirá facilitar el trabajo que en la actualidad lo hacen manualmente

### HISTORIA



En el Cantón Mocha el alcalde Ec. Sipriano Ocaña Valle después de ver la necesidad de instaurar en el Cantón Mocha un organismo de socorro que vele por la ciudadanía las veinte y cuatro horas del día, y después de haber gestionado, la petición en la Secretaria Nacional de Gestión de Riesgos a través de la Coordinación Nacional de Cuerpos de Bomberos, y después de ocho meses los organismos acuden al Cantón Mocha para analizar la propuesta conjuntamente con el señor Alcalde.

Después de un largo análisis y al ser remitida toda la información legal necesaria y de igual manera la terna que integrarían la jefatura, entonces luego de ocho meses se recibe la tan anhelada noticia positiva de creación de la institución.

El 16 de Julio del 2010, bajo Resolución No. 107 SNGR, El Sistema Designación de Bienes del Cuerpo de Bomberos Mocha está desarrollado para el cuerpo de Bomberos Mocha.

## 2. Objetivo

Ayudar a un control óptimo el que permitirá facilitar a los usuarios administrativos un mejor control de bienes y custodios. Al igual dar a conocer a la ciudadanía las diferentes actividades de Cuerpo de Bomberos Mocha

## 3. Requerimientos del Sistema

- Navegador Mozilla Firefox Quantum [versión 30 o superior], Chrome [versión 49 o superior], Opera [versión 36 o superior], Zafari [versión 9.1 o superior].
- Acrobat Reader.
- Resolución recomendada 1266 x 778.
- Acceso a Internet.
- Obtener una APIKEY de Google para la utilización de Google Maps.
- Contratar un servicio de Hosting, que permite la visualización de la página a través de internet.

## 4. Acceso al Sistema

Para acceder al sistema se debe ingresar a la dirección web [www.bomberosmocha.com](http://www.bomberosmocha.com)

## 5. Página de Inicio



Figura. 1. Página Principal  
Fuente: Elena Mariño



## 6. Ingreso al Sistema

6.1. Para ingresar al sistema nos dirigimos al menú usuarios

### IMPORTANTE

Un usuario debe tener sus credenciales de acceso el mismo que le permitirá el ingreso, el cual solo puede ser dada por el Administrador del Sistema

6.1.1. Para acceder al sistema vamos al Menú Usuarios y se despliega el despliega el submenú

Damos clic en Acceso al Sistema, ingresamos nombre y usuario e ingresar

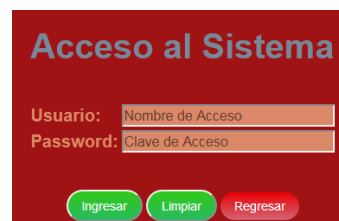
El formulario tiene un fondo rojo oscuro. En la parte superior, el título "Acceso al Sistema" está escrito en letras blancas y azules. Debajo del título, hay dos campos de entrada de texto con etiquetas "Usuario:" y "Password:". El campo "Usuario:" contiene el texto "Nombre de Acceso" y el campo "Password:" contiene "Clave de Acceso". En la parte inferior del formulario, hay tres botones: "Ingresar" (verde), "Limpiar" (verde) y "Regresar" (rojo).

Figura. 2. Acceso al Sistema

Fuente: Elena Mariño

Si el usuario está registrado aparece el mensaje de Bienvenida.

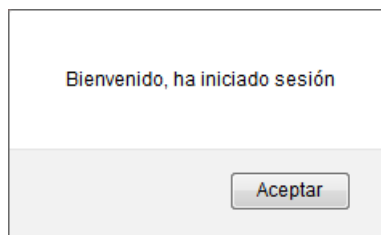
El mensaje de bienvenida está en un cuadro con fondo blanco y una franja gris en la parte inferior. El texto "Bienvenido, ha iniciado sesión" está centrado en la parte superior. En la franja gris inferior, hay un botón "Aceptar" con un fondo gris claro y un borde gris.

Figura. 3. Mensaje de Bienvenida

Fuente: Elena Mariño

Si en caso de que el usuario no este registrado el sistema nos devolverá el siguiente mensaje

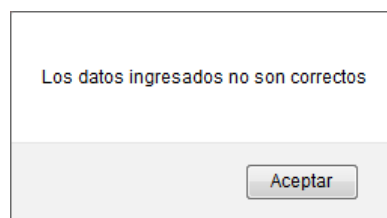
El mensaje de error está en un cuadro con fondo blanco y una franja gris en la parte inferior. El texto "Los datos ingresados no son correctos" está centrado en la parte superior. En la franja gris inferior, hay un botón "Aceptar" con un fondo gris claro y un borde gris.

Figura. 4. Mensaje de Error

Fuente: Elena Mariño

# 1. Elementos del Menú

## 1.1. Institución se desplazara el submenú el cual dará a conocer datos sobre el Cuerpo de Bomberos Mocha



Figura. 5. Menú y Submenú de Institución  
Fuente: Elena Mariño

### 1.1.1. Misión y Visión



Figura. 6. Misión y Visión  
Fuente: Elena Mariño

### 1.1.2. Historia



Figura. 7. Misión y Visión  
Fuente: Elena Mariño

## 1.2. Servicio

### 1.2.1. Información

Se presenta los archivos de la base legal y archivos de interés general. Las cuales se pueden subir y eliminar.



Figura. 8. Información  
Fuente: Elena Mariño

## 1.3. Galería

Las fotos presentadas son en formato JPG.



Figura. 9. Galería de Fotografías  
Fuente: Elena Mariño

## 1.4. Configuración

### 1.4.1. Base

Ingresa la información que permite visualizar en contáctenos



Figura. 10. Menú y Submenú de Base  
Fuente: Elena Mariño

## Tipo de Bien

Ingresar nuevos y visualizar los tipos de bien



Figura. 11. Menú y Submenú de Tipo Bien  
Fuente: Elena Mariño

Figura. 12. Ingreso de Datos del Tipo Bien  
Fuente: Elena Mariño

| Nombre de Tipo                | Descripción                                     | Acción                     |
|-------------------------------|---|----------------------------|
| Bien de Control               | Bienes de Control                               | <a href="#">Actualizar</a> |
| Activo Fijo                   | Activo Fijo                                     | <a href="#">Actualizar</a> |
| Activo de Control Actualizado | Activo de Control Descripción con actualización | <a href="#">Actualizar</a> |

Figura. 13. Mostrar Tipo Bien  
Fuente: Elena Mariño

### 1.4.2. Documentación permite buscar un documento en específico

| Nombre de Documento   | Origen | Fecha de Emisión | Clase de Documento     |
|-----------------------|--------|------------------|------------------------|
| ActaEntrega1.pdf      | Base 1 | 2018-03-16       | Sin Documento          |
| estructuradedatos.pdf | Base 1 | 2018-03-12       | Acta Entrega Recepción |

Figura. 14. Lista de Documentos  
Fuente: Elena Mariño

## 1.5. Bienes

### 1.5.1. Asignar Bienes Ingresamos un custodio y un bien y luego en guardar.

Figura. 15. Asignar Bienes  
Fuente: Elena Mariño

### 1.5.2. Ingresar Bienes ingresamos todos los datos que solicita e ingresar

**Registrar un Nuevo Bien**

Código:

Nombre:

Cantidad Ingresada:

Fecha de Ingreso: 16/03/2018

Estado: Bueno

Numero de Serie:

Marca:

Modelo:

Color: Blanco

Tipo: Bien de Control

Gráfico: sin\_imagen.jpg

Figura. 16. Nuevo Bien  
Fuente: Elena Mariño

Una vez que el bien se ingresa con éxito nos presenta el siguiente mensaje

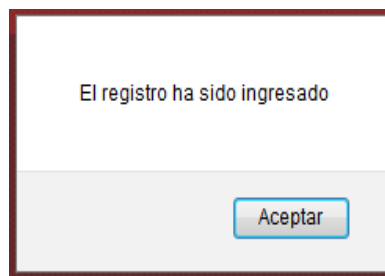


Figura. 17. Mensaje de registro exitoso  
Fuente: Elena Mariño

### 1.5.3. Lista Bienes

Puede Actualizar o Eliminar un registro

| Código            | Nombre                                   | Cantidad | Fecha de Ingreso | Estado del Bien | Numero de Serie | Marca              | Modelo  | Color    | Gráfico | Acción  |
|-------------------|--|----------|------------------|-----------------|-----------------|--------------------|---------|----------|---------|---|
| 32234233432234234 | MUEBLES DE USO GENERAL/ARMARIO/METAL1    | 4        | 2018-02-01       | Malo            | SD3422DSDRR3344 | Sin Marca          | Unico   | Amarillo |         | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> |
| 300800010001      | MUEBLES DE USO GENERAL/CAMA LITERA/METAL | 1        | 2010-12-22       | Bueno           | 0010001         | Estructura cromada | Cromada | Azul     |         | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> |

Figura. 18. Mensaje de registro exitoso  
Fuente: Elena Mariño

## 1.6. Custodios

### 1.6.1. Ingresar Custodio



Registrar Nuevo Custodio

Cédula:

Nombre:

Apellido:

Fecha de Nacimiento: 19/03/2000

Dirección:

Fecha de Ingreso: dd/mm/aaaa

Cargo: Administrador

Figura. 19. Registro de Nuevos Custodios  
Fuente: Elena Mariño

### 1.6.2. Lista de Custodio

Al ingresar un nombre verifica y mostrara los datos de los custodios ingresados en el sistema.



Lista de Custodios

Nombre del Custodio:

| Cédula      | Nombre                     | Fecha de Ingreso | Dirección            | Cargo         | Acción  |
|-------------|----------------------------|------------------|----------------------|---------------|---|
| 1234567890  | Jose Luis Castelo          | 2018-02-01       | Ambato               | Administrador | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> <input type="button" value="Editar"/> |
| 1700231222  | Carlos Pareja              | 2000-01-01       | Ambato, Huachi Chico | Administrador | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> <input type="button" value="Editar"/> |
| 06030122324 | Ángel Arturo Lema González | 2018-03-07       | Mocha                | Voluntario    | <input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/> <input type="button" value="Editar"/> |

Figura. 20. Lista de Custodios  
Fuente: Elena Mariño

### 1.6.3. Cargos, se asignara cargos para los custodios



Lista de Cargos

Nombre del Cargo:

| Nombre del Cargo | Descripcion del Cargo   | Acción                                    |
|------------------|-------------------------|---|
| Voluntario       | Personal Voluntario     | <input type="button" value="Actualizar"/> |
| Asistente        | Asistente               | <input type="button" value="Actualizar"/> |
| Administrador    | Administrador del Sitio | <input type="button" value="Actualizar"/> |

Figura. 21. Lista de Cargos  
Fuente: Elena Mariño

### 1.6.4. Nuevo Cargo ingreso de nuevos cargos esto solo lo puede realizar el administrador



Figura. 22. Lista de Cargos  
Fuente: Elena Mariño

## 1.7. Usuarios

### 1.7.1. Nuevo Usuario ingresando los datos solicitados e ingresar

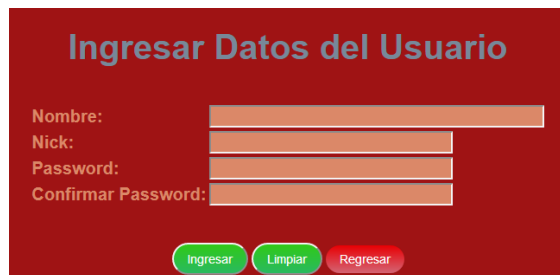


Figura. 23. Lista de Cargos  
Fuente: Elena Mariño

### 1.7.2. Salir del Sistema



Figura. 24. Salir Sistema  
Fuente: Elena Mariño

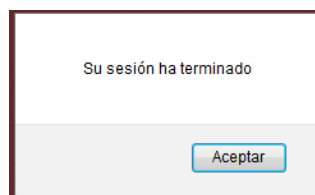


Figura. 25. Mensaje de finalización  
Fuente: Elena Mariño

## 1.8. Contáctanos encontraremos en donde podemos ubicar al Cuerpo de Bomberos Mocha

**CUERPO DE BOMBEROS MOCHA**

Bienvenido Admin [\(Salir\)](#)

Inicio Institucion Servicios Noticias Galeria Configuración Bienes Custodios Usuarios Contactenos

## Contáctenos

**Estación N. 1**

Dirección: Centro del Cantón Mocha

Teléfono: 98765432

**Cuerpo de Bomberos Mocha Base 2**

Dirección: Dirección en el cantón Mocha Base 2

Teléfono: 032987654

Figura. 26. Contáctenos  
Fuente: Elena Mariño



## **ANEXO 2. MANUAL TÉCNICO**

# **SISTEMA DESIGNACIÓN DE BIENES DEL CUERPO DE BOMBEROS MOCHA**

**MANUAL TÉCNICO**



**Versión 1.0**

**Autor: Elena Mariño**

## 7. Introducción

El Sistema Designación de Bienes del Cuerpo de Bomberos Mocha, permitirá a la Institución mantener un control actualizado y efectivo de las asignaciones de bienes, el mismo que será accesible en línea lo que facilitará el trabajo realizado por el personal encargado de dicha tarea. Con la utilización de tecnología actual y de punta.

## 8. Requerimiento del sistema

Software utilizado para la creación de la aplicación:

- PHP 4.7.4
- MYSQL 5.1
- XAMPP 7.2
- NetBeans IDE 8.2

### CREACIÓN DE LA BD

1. Se abre el Mozilla Firefox y colocar la dirección **localhost/bombersmocha**
2. Nos aparece la página principal de phpMyadmin

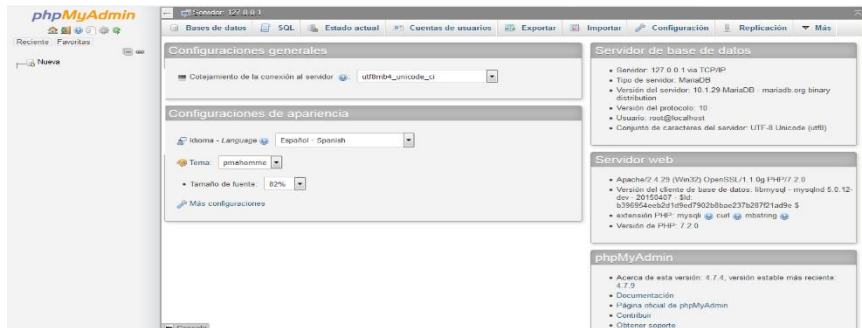


Figura. 1. Página de inicio phpMyadmin

Fuente: Elena Mariño

3. Creamos un nueva base de datos y colocamos el nombre de bdbomberos



Figura. 2. Creación de BD en MYSQL

Fuente: Elena Mariño

4. Y empezamos a crear tablas y datos.

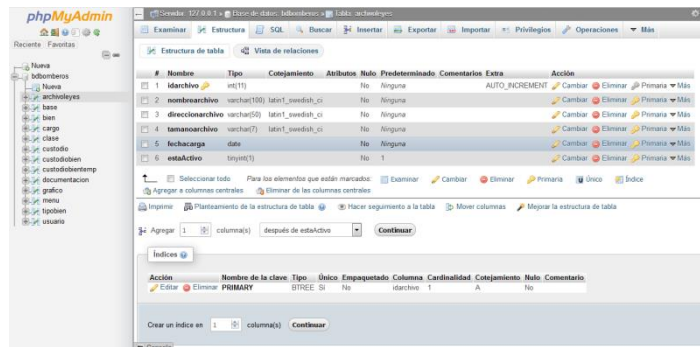


Figura. 3. Creación de tablas y datos para la BD

Fuente: Elena Mariño

5. Una vez creada la BD completa para poder trasladarla al servidor sacamos una copia de seguridad.

5.1. Damos clic en EXPORTAR y CONTINUAR

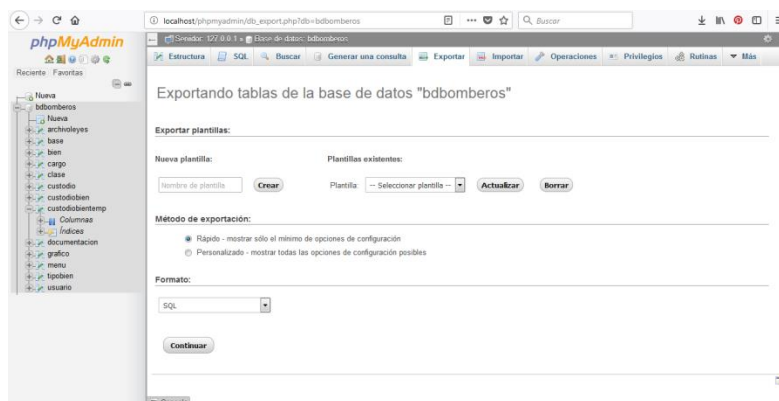


Figura. 4. Exportar la BD

Fuente: Elena Mariño

6. Aceptar y Guardamos el Archivo

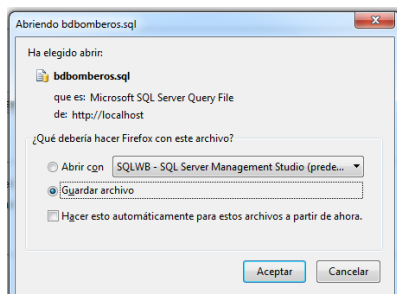


Figura. 5. Pantalla para guardar la BD

Fuente: Elena Mariño

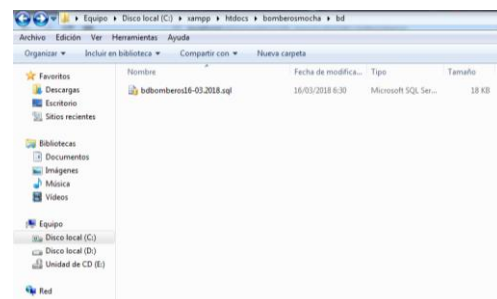


Figura. 6. Ubicación de donde se encuentra la bd exportada.

Fuente: Elena Mariño

7. Restauración de BD desde archivo en nuestro servidor creamos la BD con el mismo nombre, y damos clic IMPORTAR

8. Examinar y escogemos la copia de la BD.

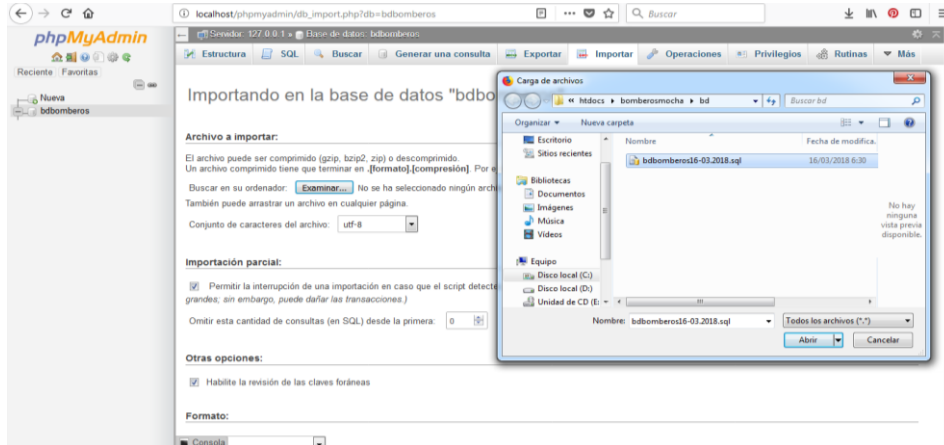


Figura. 7. Importación de la BD.

Fuente: Elena Mariño

9. Y nos confirma que se encuentra importada



Figura. 8. Mensaje Importación exitosa.

Fuente: Elena Mariño

## Codificación del sistema

10. Prendemos el XAMPP e iniciamos los componentes que necesitamos

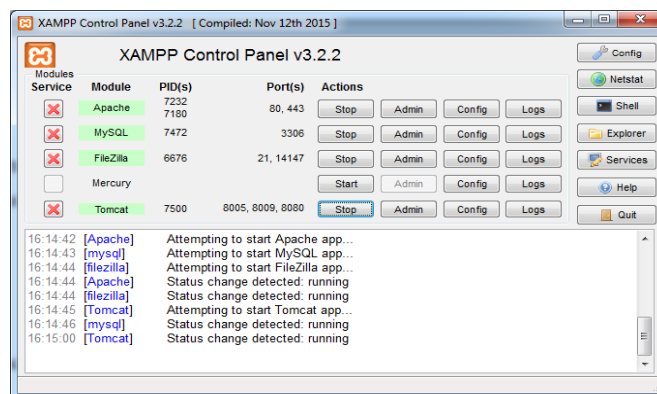


Figura. 9. Pantalla del XAMPP.

Fuente: Elena Mariño

11. Abrimos el Notepad++



```

$inicioSesion=0;
if(!empty($_SESSION['nick'])) {
    $inicioSesion = 1;
}
/* Si no hay una sesión creada, redireccionar al login. */
if(empty($_SESSION['nick'])) {
    header('Location: login.php');
}

include 'include/objUsuario.php';
include 'include/objConexion.php';
include 'include/objPagina.php';

//$idBien = $_GET['idBien'];

$pagina= new objPagina();

echo $pagina->gethead();
echo $pagina->getbody_ini();
echo $pagina->getmenu($inicioSesion);

if (isset($_GET['resultado']))
{
    $resultado = $_GET['resultado'];
    if ($resultado==1)
    {
        echo '<script type = "text/javascript"> alert("El registro ha sido
actualizado");</script>';
    }
    else
    {
        echo '<script type = "text/javascript">alert("No se pudo actualizar el
registro");</script>';
    }
}

```

```

    }
}

echo $pagina->getcont_ini();

//Insertar aqui el contenido de la pagina

$pdo = objConexion::connect();
$pdo1 = objConexion::connect();

if (!isset($_GET['idBase']))
    echo "Error, parametro no definido";
else
    $idBase=$_GET['idBase'];

$sql = "SELECT * FROM base where idBase = '".$_GET['idBase']."'";
foreach ($pdo->query($sql) as $row) {
    $base = new
objBase($row['idBase'],$row['telefonoBase'],$row['nombreBase'],$row['numeroBas
e'],
        $row['direccionBase'],$row['longitud'],$row['latitud']);
}
objConexion::disconnect();

?>

<div>
    <h3 class="fnt_tit_secundario">Actualizar Datos de la Base</h3>
</div>

<form action="registraenBD.php" method="post">
    <table align="center">
        <tr>

```



```

        <th class="fnt_encabezado texto_negrita" align="left">Numero</th>
        <td>
            <input class="caja_texto" name="numeroBase" type="text"
required maxlength="2" size="3"
            value="<?php echo $base->getNumero();?>" pattern="[0-9]+"
        >
        </td>
    </tr>
    <tr>
        <th class="fnt_encabezado texto_negrita"
align="left">Nombre</th>
        <td>
            <input class="caja_texto" name="nombreBase" type="text"
required maxlength="100" size="60"
            value="<?php echo $base->getNombre();?>"
        </td>
    </tr>
    <tr>
        <th class="fnt_encabezado texto_negrita" align="left">
Direcci&oacute;n
        <td>
            <input class="caja_texto" name="direccionBase" type="text"
required maxlength="200" size="100"
            value="<?php echo $base->getDireccion();?>"
        </td>
    </tr>
    <tr>
        <th class="fnt_encabezado texto_negrita"
align="left">Telefono</th>
        <td>
            <input class="caja_texto" name="telefonoBase" type="text"
required maxlength="13" size="13"

```

```

        value="<?php echo $base->getTelefono() ?>" required
pattern="[0-9]+">
        </td>
    </tr>
    <tr>
        <th class="fnt_encabezado" texto_negrita"
align="left">Longitud</th>
        <td>
            <input class="caja_texto" name="longitudBase" type="text"
required maxlength="13" size="13"
            value="<?php echo $base->getLongitud() ?>" required
pattern="[0-9\.\-]+">
        </td>
    </tr>
    <tr>
        <th class="fnt_encabezado" texto_negrita"
align="left">Latitud</th>
        <td>
            <input class="caja_texto" name="latitudBase" type="text"
required maxlength="13" size="13"
            value="<?php echo $base->getLatitud() ?>" required
pattern="[0-9\.\-]+">
        </td>
    </tr>
</table>

<br><br>
<div align="center">
    <input type="hidden" name="opcion" value="15">
    <input type="hidden" name="idBase" value="<?php echo $base-
>getId(); ?>">
    <button class="btn_verde" type="submit" >Actualizar</button>
    <button class="btn_verde" type="reset" >Restaurar</button>
    <a class="btn_rojo" href="consultarbase.php">Regresar</a>

```

```
</div>
```

```
</form>
```

```
<?php  
echo $pagina->getcont_fin();  
echo $pagina->getpie();  
echo $pagina->getbody_fin();  
echo $pagina->gethtml_fin();  
?>
```

#### 14. Conexión para mostrar Google Maps **contacto.php**

```
<script>  
function initMap() {  
    <?php  
    for($d=0;$d<$c;$d++){  
        echo 'var uluru' . ($d+1) . ' = {lat: ' . $datosBase[$d][ "latitud" ] . ' , lng:  
' . $datosBase[$d][ "longitud" ] . '};';  
    }  
    ?>  
    var map = new google.maps.Map(document.getElementById('map'), {zoom:  
15, center: uluru1 });  
    <?php  
    for($d=0;$d<$c;$d++)  
    {  
        echo 'var marker' . ($d+1) . ' = new google.maps.Marker({ position:  
uluru' . ($d+1) . ',map: map});';  
    }  
    ?>  
    }  
</script>  
<script async defer  
  
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyDydmiwBmhPaMLiJ68  
Gt8hDm_v1DGdJjWM&callback=initMap">  
</script>
```

#### 15. Ingresar al sistema **ingresoUsuario.php**

```
<?php  
session_start();  
$inicioSesion=0;  
if(!empty($_SESSION['nick'])) {  
    $inicioSesion = 1;  
}  
include 'include/objUsuario.php';  
include 'include/objConexion.php';
```

```

include 'include/objPagina.php';
include 'include/funciones.php';
$pagina= new objPagina();
echo $pagina->gethead();
echo $pagina->getbody_ini();
echo $pagina->getmenu($inicioSesion);

if (isset($_GET['resultado']))
{
    $resultado = $_GET['resultado'];
    if ($resultado==1)
        echo '<script type = "text/javascript"> alert("El registro ha sido
ingresado");</script>';
    else

        if ($resultado==2)
            echo '<script type = "text/javascript"> alert("Las Contraseñas deben ser
iguales");</script>';
        else
            echo '<script type = "text/javascript">alert("No se pudo ingresar el
registro");</script>';
    }
}

echo $pagina->getcont_ini();

//Insertar aquí el contenido de la pagina

?>
<div>
    <h3 class="fnt_tit_secundario">Ingresar Datos del Usuario</h3>
</div>

<form action="registraenBD.php" method="post">
    <table align="center">
        <tr>
            <td>
                <label class="fnt_encabezado texto_negrita">Nombre:</label>
            </td>
            <td>
                <input id='NombreUsuario' class="caja_texto" name="nombre"
type="text" required
value=""      onchange="validaCaracter(this)"      size="50"
maxlength="30">
            </td>
        </tr>
        <tr>
            <td>
                <label class="fnt_encabezado texto_negrita">Nick:</label>
            </td>
            <td>

```

```

        <input class="caja_texto" name="nick" type="text" value=""
size="35" required maxlength="20">
    </td>
</tr>
<tr>
    <td>
        <label class="fnt_encabezado texto_negrita">Password:</label>
    </td>
    <td>
        <input class="caja_texto" name="pass" id="password"
type="password" value=""
required size="35" maxlength="25">
    </td>
</tr>
<tr>
    <td>
        <label class="fnt_encabezado texto_negrita">Confirmar
Password:</label>
    </td>
    <td>
        <input class="caja_texto" name="confirmacion"
id="confPassword" type="password" required value=""
size="35"
maxlength="25"
onchange="validaCadenasIguales(this,document.getEleme
ntById('password'))">
    </td>
</tr>
</table>
<br><br>
<div align="center">
    <input type="hidden" name="opcion" value="2">
    <button class="btn_verde" type="submit" >Ingresar</button>
    <button class="btn_verde" type="reset" >Limpiar</button>
    <a class="btn_rojo" href="index.php">Regresar</a>
</div>
</form>

```

16. Una vez creada la aplicación con su codificación se debe tomar en cuenta que todas las carpetas y los archivos estén en una sola carpeta que la colocaremos en c:\xampp\htdocs\bomberosmocha

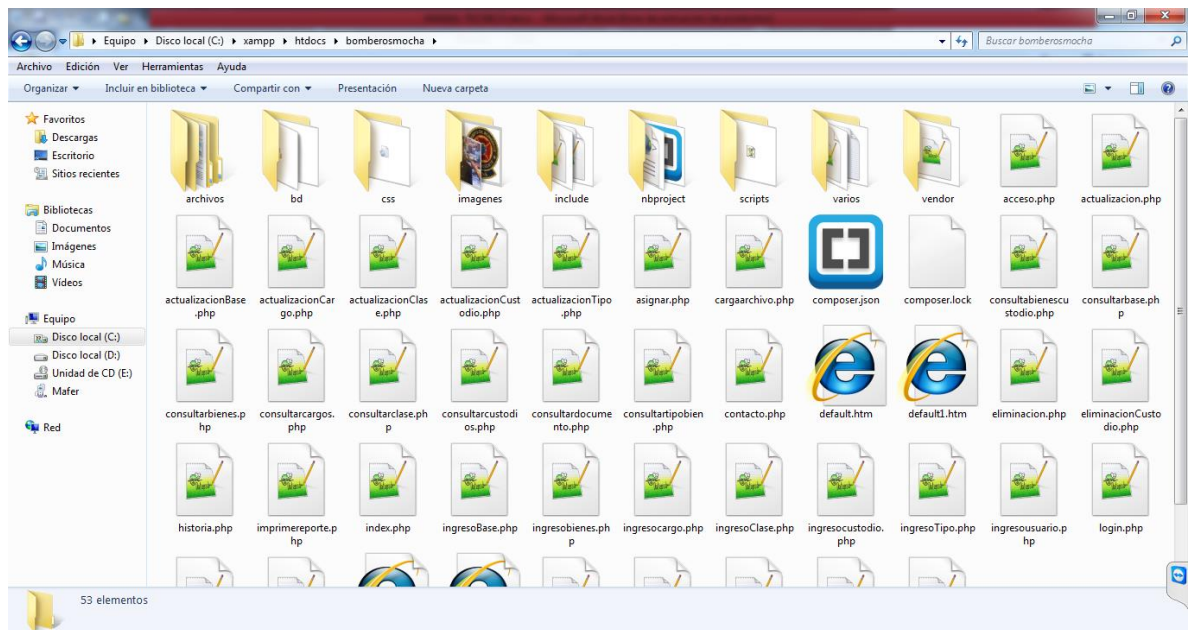


Figura. 11. Carpetas y archivos del sistema.

Fuente: Elena Mariño