

**APLICACIÓN WEB PARA GENERAR CERTIFICADOS DE CALIBRACIÓN PARA
EQUIPOS MÉDICOS.**



Jose Luis Espinosa Buitrago

Universidad Santo Tomás
Facultad de Ingeniería de Electrónica
Bogotá, Abril de 2017

**APLICACIÓN WEB PARA GENERAR CERTIFICADOS DE CALIBRACIÓN PARA
EQUIPOS MÉDICOS.**



Jose Luis Espinosa Buitrago

Monografía

Director: Ing. Jaime Vitola Oyaga

Universidad Santo Tomás
Facultad de Ingeniería de Electrónica
Bogotá, Abril de 2017

Nota de aceptación del jurado

Firma Jurado

Firma Jurado

Bogotá D.C. Febrero de 2017

DEDICATORIA

A mi madre Blanca Myriam Buitrago.

Por haberme apoyado en todo momento, aun en los tiempos más difíciles, por creer en mí, por su motivación constante, por los grandes valores que me inculco, y lo más importante, por su gran amor.

A mi padre Jose Espinosa Gómez.

Por los ejemplos de responsabilidad, honestidad y constancia, por su ayuda, por depositar su confianza en mí y por su amor.

A mis hermanas Mónica y Pilar.

Por su tiempo, sus enseñanzas, sus experiencias, por estar siempre pendiente de mí, por su incondicional apoyo y motivación, por ese gran cariño que siempre me han brindado y por inculcarme lo correcto e irme mostrando como ser una persona de bien.

A mis Familiares y Amigos.

Por compartir experiencias que constantemente nos enriquecían como personas.

A mis Profesores.

Por compartir su sabiduría y gracias a eso, permitirme ser un profesional.

RECTOR GENERAL

Fray Juan Ubaldo López Salamanca, O.P.

VICERECTOR ADMINISTRATIVO Y FINANCIERO

Fray Diego Orlando Serna Salazar, O.P.

VICERRECTOR ACADEMICO GENERAL

Fray Mauricio Antonio Cortés Gallego, O.P.

SECRETARIO GENERAL

Héctor Fabio Jaramillo Santamaria.

DECANO DE DIVISIÓN DE INGENIERÍAS

Fray Luis Antonio Alfonso Vargas, O.P.

SECRETARIA DE DIVISIÓN

Myriam Gómez Colmenares.

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA

Edgar Javier Barajas Herrera.

ADVERTENCIA

La universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados por los autores en el trabajo de grado, sólo velará por que no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III – Art. 46 del Reglamento de la Universidad Santo Tomás.

GLOSARIO

Apache Server: Software instalado en un servidor para entender el protocolo HTTP.

Calibración: Proceso de comparación entre un instrumento de medición y un patrón de referencia.

Certificado: Documento escrito en el que se da por verdadera una cosa.

DOM: Estructura jerárquica donde existen objetos que dependen unos de otros, generalmente se emplea este término en documentos web.

HTML: Lenguaje que compone la estructura de una página web en el navegador.

HTTP: Protocolo de comunicación que permite la transferencia de información en internet.

JavaScript: Lenguaje de programación, principalmente usado en el navegador para otorgar dinamismo a un sitio web.

Metrología: Ciencia que se encarga de estudiar las medidas.

MySQL: Software que se encarga de la gestión de una base de datos.

PHP: Lenguaje de programación, principalmente usado en servidores para otorgar dinamismo a una aplicación web.

Tabla De Contenido

1.	PLANTEAMIENTO DE PROBLEMA-----	1
2.	HUMANISMO-----	2
3.	ANTECEDENTES -----	3
4.	JUSTIFICACIÓN-----	4
5.	OBJETIVO GENERAL -----	5
5.1.	OBJETIVOS ESPECÍFICOS -----	5
6.	MARCO TEÓRICO-----	6
6.1.	METROLOGÍA Y CALIBRACIÓN-----	6
6.2.	APLICACIÓN WEB -----	6
6.2.1.	ARQUITECTURA WEB MVC-----	7
6.3.	TECNOLOGÍAS WEB USADAS -----	8
6.3.1.	APACHE HTTP SERVER -----	8
6.3.2.	PHP -----	8
6.3.3.	MYSQL-----	9
6.3.4.	JAVASCRIPT -----	10
6.3.5.	AJAX-----	10
6.3.6.	CSS -----	12
7.	DISEÑO METODOLÓGICO -----	13
7.1.	POBLACIÓN MUESTRA-----	13
7.2.	LEVANTAMIENTO DE REQUERIMIENTOS-----	15
7.2.1.	ETNOGRAFÍA-----	15
7.2.2.	MODELADO -----	15
7.2.3.	EJECUCIÓN LEVANTAMIENTO DE REQUERIMIENTOS -----	15
7.3.	DISEÑO DE LA PLATAFORMA-----	16
7.3.1.	INTERFACE DE CONEXIÓN CON LA BASE DE DATOS -----	16
7.3.2.	INTERFACE DE ACCESO -----	16
7.3.3.	ENTORNO DE APLICACIÓN-----	17
7.3.4.	SELECCIÓN DE EQUIPO MÉDICO A CALIBRAR -----	17
7.3.4.1.	INTERFACE DE ADQUISICIÓN SEGÚN EL EQUIPO -----	17
7.3.4.2.	VALIDACIÓN DE DATOS -----	17
7.3.4.3.	PROCESAMIENTO DE DATOS -----	18
7.3.4.4.	GENERACIÓN DE CERTIFICADO-----	18
7.3.4.5.	ALMACENAMIENTO BASE DE DATOS-----	18

7.3.5.	INTERFACE DE RESPALDO -----	19
7.3.6.	INTERFACE DE USUARIO -----	19
7.4.	EVALUACIÓN DEL PROYECTO -----	13
7.4.1.	FUNCIONALIDAD -----	13
7.4.2.	FIABILIDAD -----	13
7.4.3.	USABILIDAD -----	13
7.4.4.	EFICIENCIA -----	13
7.4.5.	MANTENIBILIDAD -----	14
7.4.6.	PORTABILIDAD -----	14
8.	EJECUCIÓN DEL PROYECTO -----	15
8.1.	REQUISITOS DE LA APLICACIÓN -----	19
8.1.1.	SERVIDOR -----	19
8.1.2.	DISTRIBUCIÓN XAMPP -----	20
8.1.2.1.	APACHE HTTP SERVER -----	20
8.1.2.2.	MySQL -----	20
8.1.2.3.	PHP -----	21
8.2.	ARQUITECTURA DE LA APLICACIÓN -----	21
8.2.1.	“BOOTSTRAP.PHP” -----	22
8.2.2.	“REQUEST.PHP” -----	22
8.2.3.	“SESSION.PHP” -----	23
8.2.4.	“DATABASE.PHP” -----	23
8.2.5.	“GLOBALES.PHP” -----	24
8.2.6.	“CONTROLLER.PHP” -----	25
8.2.7.	“MODEL.PHP” -----	26
8.2.8.	“VIEW.PHP” -----	27
8.3.	DIAGRAMA DE CASOS DE USO -----	27
8.4.	INTERFAZ DE INSTALACIÓN -----	28
8.4.1.	ESTRUCTURA DE LAS TABLAS -----	30
8.4.1.1.	TABLA “certificados” -----	30
8.4.1.2.	TABLA “certificadosmeta” -----	31
8.4.1.3.	TABLA “usuarios” -----	31
8.4.1.4.	TABLA “usermeta” -----	31
8.4.1.5.	TABLA “terms” -----	31
8.4.1.6.	TABLA “term_relationships” -----	31
8.4.1.7.	TABLA “term_taxonomy” -----	31

8.4.1.8.	TABLA “opciones” -----	31
8.4.2.	NOMBRE DE LA BASE DE DATOS -----	32
8.4.3.	NOMBRE DE USUARIO MYSQL -----	32
8.4.4.	CONTRASEÑA MYSQL -----	32
8.4.5.	SERVIDOR DE LA BASE DE DATOS -----	32
8.4.6.	PREFIJO DE TABLA -----	33
8.4.7.	INTERFAZ PARA LA CREACIÓN DE USUARIO ADMINISTRADOR -----	33
8.5.	INTERFAZ DE ACCESIBILIDAD -----	34
8.6.	ENTORNO DE APLICACIÓN -----	36
8.6.1.	SECCIÓN INICIO -----	37
8.6.2.	SECCIÓN CERTIFICADOS -----	40
8.6.2.1.	INTERFAZ DE ADQUISICION DE DATOS -----	42
8.6.2.2.	PROCESAMIENTO DE DATOS -----	46
8.6.2.3.	GENERACION DE CERTIFICADO Y ALMACENAMIENTO -----	47
8.6.3.	SECCIÓN CLIENTES -----	49
8.6.3.1.	VISTA “NUEVO CLIENTE” -----	50
8.6.3.2.	VISTA “LISTA CLIENTES” -----	51
8.6.4.	SECCIÓN PATRONES -----	51
8.6.4.1.	VISTA “CREAR PATRÓN” -----	52
8.6.4.2.	VISTA “LISTA PATRONES” -----	53
8.6.5.	SECCIÓN ADMINISTRACIÓN -----	54
8.6.5.1.	VISTA “NUEVO USUARIO” -----	55
8.6.5.2.	VISTA “USUARIOS” -----	56
8.6.5.3.	VISTA “USUARIOS ELIMINADOS” -----	57
8.6.6.	SECCIÓN APLICACIÓN -----	58
8.6.7.	SECCIÓN EDITAR MI PERFIL -----	59
9.	CONCLUSIONES, TRABAJO FUTURO Y EVALUACIÓN DEL PROYECTO -----	61
9.1.	CONCLUSIONES -----	61
9.2.	TRABAJO FUTURO -----	62
9.3.	EVALUACIÓN DEL PROYECTO -----	63
9.3.1.	FUNCIONALIDAD -----	63
9.3.2.	FIABILIDAD -----	63
9.3.3.	USABILIDAD -----	63
9.3.4.	EFICIENCIA -----	63
9.3.5.	MANTENIBILIDAD -----	63

9.3.6.	PORTABILIDAD	64
10.	REFERENCIAS	65
11.	ANEXOS	67
11.1.	CERTIFICACIÓN DE LA EMPRESA GMETROLOGIA	67
11.2.	EJEMPLO DE CERTIFICADO GENERADO	68
11.3.	DIAGRAMA DE CLASES	71
11.4.	CLASES PRINCIPALES DE LA APLICACIÓN	72
11.4.1.	"REQUEST.PHP"	72
11.4.2.	"BOOTSRAP.PHP"	73
11.4.3.	"CONTROLLER.PHP"	74
11.4.4.	"MODEL.PHP"	76
11.4.5.	"VIEW.PHP"	98

1. PLANTEAMIENTO DE PROBLEMA

La gestión hospitalaria es importante en la atención oportuna de los usuarios al servicio de salud, para este fin los equipos médicos tienen un papel indispensable y vital para muchos pacientes que necesitan procedimientos quirúrgicos, chequeos y hospitalización. La calibración de equipos médicos hace parte de los mantenimientos preventivos, programados por las entidades hospitalarias con el fin de realizar una gestión eficiente y evitar los mantenimientos correctivos [1]. Por esto, la información de calibración debe cumplir la norma NTC/ISO-IEC-17025 con el fin de establecer procedimientos ágiles para las empresas que realizan la metrología.

La empresa Gmetrologia S.A.S es una empresa dedicada a metrología de equipos médicos. El proceso de generación de certificados de calibración se desarrollaba desde el área técnica, utilizando la herramienta Excel y los instrumentos calibradores. Los técnicos, además de digitar las mediciones del instrumento calibrador, debían crear una hoja de cálculo con los datos arrojados por el instrumento, y procesarlos para determinar si el equipo cumple con los estándares de la norma NTC/ISO-IEC-17025. Por último generaban una plantilla informando el estado del equipo por medio de tablas, gráficos y su correspondiente análisis para entregar al cliente, sin un medio de gestión para futuras consultas y correspondiente seguimiento.

Por todo esto, se analizó que la empresa tiene inconvenientes en los siguientes aspectos: Vulnerabilidad, descentralización, imprecisión e indisponibilidad de la información del proceso de generación de certificados de calibración de equipos médicos. Por lo cual, se hizo un diseño que permita la confidencialidad, integridad y disponibilidad de la información de los certificados de calibración. Con este fin, se propuso hacer una aplicación web que tenga 7 estados: Interface de accesibilidad de usuario, identificación de la clase de equipo, interface de adquisición de datos según el tipo de equipo, validación de datos, procesamiento de datos, generación del certificado y almacenamiento del certificado.

En esta propuesta se trabajó los equipos de clase: I (Básculas), IIA (Monitor, Termo higrómetros), IIB (Electro bisturí), clasificados en el decreto 4725 de 2005 mencionados en el Capítulo 2 - Artículo 5º. Dado lo anterior, surge la siguiente pregunta problema.

¿Cómo se desarrolla una aplicación web que permita generar certificados de calibración para equipos médicos?

2. HUMANISMO

Los servicios de salud en Colombia actualmente se caracterizan por tener un servicio de mala calidad, generando una relación deshumanizante entre las entidades hospitalarias y los pacientes. Parte del problema reside en la Ley 100 de seguridad social, que en el año 1993 reformó el sistema de salud en Colombia, privando a los hospitales públicos de recursos estatales, otorgándoles un subsidio bajo demanda. Además, se crearon entidades intermediarias como las ARS (Administradoras de Régimen Subsidiado) y las EPS (Entidad Promotora de salud), las cuales son costosas, ineficientes y creadoras de corrupción administrativa y política [2]. Adicionalmente existen otros factores de tipo social, institucional y formativo que han contribuido a esta deshumanización.

Basado en la problemática descrita anteriormente, la visión de la Universidad Santo Tomás, la cual hace referencia a la protección y la promoción de la dignidad de la persona humana, motiva a los estudiantes a que participen en proyectos sociales orientados a brindar soluciones de las diversas problemáticas de la actual sociedad colombiana, en la cual estos proyectos hacen uso de los diversos conocimientos adquiridos a lo largo de la vida universitaria, permitiendo que la comunidad se beneficie en temas como la seguridad, salud y bienestar. Por lo tanto, la función del ingeniero electrónico, debiera identificarse con el ejercicio del sentido ético y crítico, propios de la doctrina y filosofía tomista, para que estos respondan a los desafíos de la sociedad actual. Por tal motivo el ingeniero tomasino busca:

- El compromiso social, el cual tenga presente el interés público sobre el individual.
- La manipulación útil de tecnologías e instrumentos utilizables para ejecutar objetivos que busquen el bien de la comunidad.
- Tener presente el respeto y la conservación de los entornos ambientales, económicos y humanos.
- Iniciativa en el desarrollo de acciones que busquen la construcción del tejido social.

Por tanto, este proyecto está dirigido al sector salud en el cual el componente humanístico juega un rol fundamental en salvaguardar y promover la dignidad de la persona humana de una forma integral y con respeto. Además, se pretende que con el desarrollo de este proyecto, se consigne un registro para que ayude a futuros ingenieros que van en búsqueda de respuestas para su crecimiento profesional y también se desea, que sea implementado en un ambiente empresarial ya que reúne conocimientos tecnológicos que fomentan el interés investigativo e intelectual.

Por último, aspiro como ingeniero electrónico enfrentar cada una de las dificultades que surjan en el campo laboral, ser siempre un profesional íntegro y aportar a la sociedad los valores enseñados por el pensamiento de Santo Tomás de Aquino, junto con el conocimiento adquirido para ayudar a la comunidad y al país.

3. ANTECEDENTES

En 2006, en el Departamento de Risaralda existió un riesgo de salud pública. Las entidades prestadoras del servicio de salud, están sin acreditación en norma NTC/ISO-IEC-17025 y las entidades externas que a su vez son contratadas por estas, para calibración de sus equipos, tampoco se encuentran acreditadas. Con base en la información obtenida en equipos sin calibrar, clase: IIA (Monitor, Desfibrilador, Oxímetros, Tensiómetros), IIB (Electrobisturí, Máquinas de anestesia, Ventiladores), los diagnósticos tenían gran potencial para ser errados. También colocaba en desventaja competitiva a las entidades de salud, frente a otras que se encontraban acreditadas. Por ende la Universidad tecnológica de Pereira decide determinar e investigar las características de calidad relacionadas con el seguimiento de patrones de calibración de equipos biomédicos y su seguridad eléctrica, implementadas por las entidades prestadoras del servicio de salud en el Departamento de Risaralda. [3]

Para el investigador tomasino es de interés el poder ayudar en temas que competen con el campo de la salud, es importante destacar que desde los valores de un tomasino sobresale el humanismo y el bien común. En esta investigación se ayuda a un bien común y es el poder certificar equipos médicos, para que no atenten a la salud de las personas y cumplan con las normas establecidas para tal propósito.

4. JUSTIFICACIÓN

Uno de los aportes al realizar este proyecto es dar una solución en los procesos de una empresa, mediante la creación de una aplicación web que permita la generación de certificados de calibración. La empresa requería el uso de Excel para solventar el problema, lo cual implicaba mayor tiempo y no existía un sistema de gestión y seguridad que permitiera hacer seguimiento a los certificados y mantuviera la integridad de los mismos, como lo requiere la norma NTC/ISO-IEC-17025 en la sección 4.3.

A nivel social, es indispensable contar con certificados que garanticen el buen funcionamiento de un equipo médico, rigiéndose por una norma internacional (NTC ISO-IEC 17025.) ya que se tratan de equipos que pueden atentar con la vida, si se encuentran en mal estado. Por eso, es muy importante para un investigador tomasino aportar con sus conocimientos y aptitudes para generar una solución a un problema que involucra el bien común y el humanismo.

Por otro lado, la aplicación optimiza los tiempos y la imprecisión que empleaban para este proceso, ya que se validan los datos ingresados y los cálculos se hacen en tiempo real, esto genera más eficiencia en el proceso de creación del certificado y ayuda al hacer gestión y realizar seguimientos, debido a la centralización de la información. Además, se obtiene mayor seguridad, garantizando que solo el personal elegido tendrá acceso a la aplicación y por ende a los certificados de calibración. También es más difícil que se pierda información debido a su almacenamiento en la base de datos y la empresa tiene mayor accesibilidad ya que puede realizar calibraciones de una manera sencilla y rápida. Todo esto rigiéndose con la norma NTC/ISO-IEC 17025.

5. OBJETIVO GENERAL

Desarrollar una aplicación web para el proceso de generación de certificados de calibración para equipos médicos, de acuerdo con las necesidades establecidas por la empresa G Metrología S.A.S.

5.1. OBJETIVOS ESPECÍFICOS

- Crear entorno gráfico para la instalación de la aplicación y asignación del administrador, junto con el sistema de accesibilidad.
- Identificar las características de los certificados de calibración de los equipos médicos de acuerdo con la norma NTC/ISO-IEC-17025 y la empresa G Metrología S.A.S.
- Construir el entorno web para la identificación y adquisición de datos de acuerdo con la validación de campos y el tipo de equipo médico.
- Elaborar el certificado de calibración de acuerdo con la formula dada por G metrología S.A.S. y los datos adquiridos.
- Generar y almacenar los certificados de calibración, de acuerdo con la norma NTC/ISO-IEC-17025 numeral 4.3 y la empresa G Metrologia S.A.S.
- Permitir que la aplicación pueda hacer gestión de los certificados.
- Garantizar que todos los documentos distribuidos entre el personal del laboratorio como parte del sistema de gestión, deben ser revisados y aprobados para su uso, por el personal autorizado antes de su emisión. Como se indica en el numeral 4.3.2.1 de la norma NTC/ISO-IEC-17025.
- Asegurar que cada certificado cuente con un consecutivo para identificarlo unívocamente. Dicha identificación debe incluir la fecha de emisión y/o una identificación de la revisión, la numeración de las paginas, el numero total de paginas o una marca que indique el final del documento, y la o las personas autorizadas a emitirlos. Tal como indica el numeral 4.3.2.3 de la norma NTC/ISO-IEC-17025.

6. MARCO TEÓRICO

6.1. METROLOGÍA Y CALIBRACIÓN

La metrología es la ciencia que se encarga de las mediciones, y de los equipos utilizados para dicha tarea, así como de su verificación y calibración periódica.

Las mediciones son importantes en los procesos industriales y productivos. Por eso, garantizar una medición confiable es primordial para las empresas, las cuales deben contar con buenos instrumentos para obtener medidas precisas y asegurar buenos resultados. Además, gracias a la metrología las empresas generan: Calidad, Productividad y Competitividad. [4]

Para poder realizar el proceso de calibración y asegurar que los equipos obtengan medidas correctas y verídicas, es necesario regirse por una norma internacional que regule las entidades encargadas de realizar dicha labor a los equipos medidores. Para ello se crea la norma ISO/IEC-17025, la cual es aplicada en los laboratorios que se encarguen de calibración. [5]

El uso de esta Norma Internacional facilitará la cooperación entre los laboratorios ayudando al intercambio de información y experiencia. Para esto, se debe cumplir con la sección 5.10.4 donde se encuentran las reglas que debe tener el certificado de calibración. En dicho certificado es transmitida toda la información de metrología del equipo medidor, donde de manera clara se indicará el estado del equipo de acuerdo al análisis pertinente del laboratorio. [6]

6.2. APLICACIÓN WEB

Una aplicación web es el medio por el cual los usuarios pueden acceder a un servidor web a través de Internet o de una intranet mediante un navegador. Siendo la aplicación un lenguaje soportado por el navegador, quien será el intermediario entre el servidor y la aplicación.

El navegador web tiene la responsabilidad de comunicarse con el servidor por medio del protocolo HTTP y solicitar los recursos que el usuario previamente haya solicitado. De lado del servidor web, este tiene que procesar la solicitud por medio de un programa como Apache Server y por medio de algún lenguaje de servidor (PHP, ASP, Python, Java, etc.) debe entregar el recurso en un formato que el browser o navegador entienda, como son archivos HTML, CSS, JavaScript, Flash, MP4, MP3, JPG, PNG, etc.

Una buena aplicación web, debe tener una base de datos bien estructurada, puesto que es la parte que contendrá la información de la misma y otorga dinamismo al proyecto web. Esta base de datos debe ser instalada en el servidor, cuyo propósito será gestionar la información que se requiera o que se desee crear. Para ello existen múltiples gestores como MySQL, PostgreSQL, Oracle, Microsoft SQL, etc. Los lenguajes de servidor como los mencionados anteriormente, son los encargados de realizar el puente con el gestor de base de datos y poder trabajar mutuamente para poder obtener, crear, editar o eliminar datos según sea la solicitud.

Las aplicaciones web son una solución muy interesante para las empresas hoy en día, ya que en todo momento garantizan accesibilidad, ahorran tiempo a los clientes, se obtienen y analizan datos con mayor facilidad, generan mayor captación de potenciales clientes, están disponibles en todo momento, etc. Todas estas cualidades hacen que las aplicaciones web sean una opción importante a implementar, casi en cualquier modelo de negocio.

6.2.1. ARQUITECTURA WEB MVC

MVC (Model-View-Controller) es un patrón de arquitectura de software, que permite separar la lógica de programación, del diseño gráfico de cualquier aplicación. Su propósito es mantener una mejor estructura en el código, aumentar la escalabilidad de cualquier proyecto y facilitar el mantenimiento de la aplicación [7].

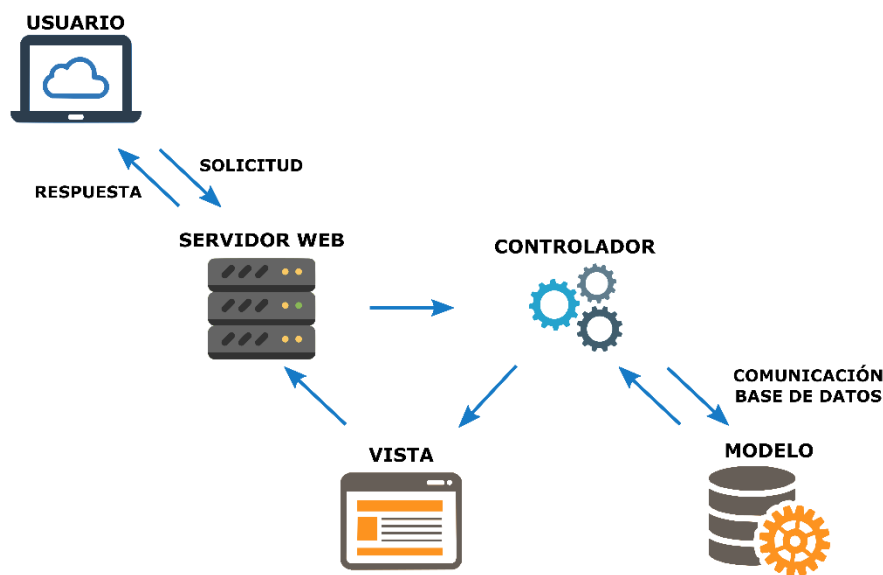


Figura 6-1: Arquitectura MVC. Fuente: [8]

Los principales componentes de MVC (ver **Figura 6-1**) son los siguientes:

- **El Modelo:** es el encargado de comunicarse con la base de datos de la aplicación y llevar a cabo todos los procesos de: obtener, crear, editar y eliminar información.
- **El Controlador:** es el “núcleo” de esta arquitectura, puesto que se encarga de atender las solicitudes que haga el usuario, además debe procesar las solicitudes, comunicándose con “Modelo” o “Vista”, según sea necesario. Por lo general, una solicitud hecha al controlador, siempre debe terminar en una interfaz proporcionada por “Vista” y en dicha vista debe estar contenida la información suministrada por el “Modelo”.
- **La Vista:** Proporciona la interfaz gráfica de la aplicación y su función principal es interactuar con el usuario, representando la información solicitada por el mismo. De tal manera que el usuario pueda volver a realizar solicitudes al “Controlador” y así mantener la funcionalidad del proyecto.

6.3. TECNOLOGÍAS WEB USADAS

6.3.1. APACHE HTTP SERVER

Apache HTTP server es un programa, que funciona en múltiples sistemas operativos, entre ellos: Macintosh, Windows, Unix, etc. Es muy popular, siendo instalado en el 70% [9] de todos los servidores de internet, gracias a que es un sistema seguro, robusto y de código abierto.

En 1999, se formó la Fundación de Software Apache (Apache Software Foundation) para obtener apoyo financiero, organizativo y legal para el servidor.

Apache server tiene varias funciones, entre ellas atender todas las peticiones hechas por los usuarios de una aplicación web, por medio del protocolo HTTP, incluyendo una ejecución multitarea ya que pueden darse peticiones simultáneas. Es importante recalcar que apache debe gestionar los diferentes errores ocurridos en el servidor, informando al navegador que está solicitando el recurso. Además se puede gestionar el uso de logs, es decir almacenar las peticiones recibidas, errores que se han producido y en general toda aquella información que puede ser registrada y analizada posteriormente para obtener las estadísticas de acceso al sitio web.

6.3.2. PHP

PHP (Hypertext Preprocessor), es un lenguaje de programación interpretado que es procesado en un servidor web (Apache, Nginx, etc.). Es principalmente usado para

brindar dinamismo entre una solicitud y una consulta en la base de datos, presentando la información por medio de una aplicación web [10].

PHP es de código abierto y multiplataforma, pudiendo ser utilizado en la mayoría de sistemas operativos y libremente. Es débilmente tipado (no es necesario declarar el tipo de dato a almacenar en una variable) y principalmente está orientado al desarrollo de proyectos web pudiendo embeber código PHP en HTML. Su sintaxis es similar a C++ facilitando la adopción de programadores provenientes de este lenguaje [11].

La comunidad de PHP es una de las más grandes, encontrando mucha información y facilitando el desarrollo de aplicaciones, también es importante mencionar que la documentación es muy buena y siempre esta accesible en la página php.net.

Por último, con PHP se pueden realizar imágenes, ficheros PDF, ficheros de texto y gracias a su amplia comunidad, existen librerías y frameworks que facilitan el desarrollo de proyectos robustos, gracias a la experiencia que aportan otros programadores [12].

6.3.3. MYSQL

MySQL es un gestor de base de datos de código abierto y multiplataforma, pudiendo ser instalado en los principales sistemas operativos como: Linux, Unix, Windows, etc. [13]. Toda la información suministrada a la base de datos es almacenada mediante tablas y atributos, estos atributos también pueden ser relacionados con los atributos de otra tabla, con el fin de evitar duplicidad o redundancia en la información.

MySQL usa SQL (Structured Language Query) para poder gestionar la base de datos mediante consultas, de tal forma que se pueda agregar, actualizar, editar o eliminar información de las tablas.

Principales ventajas de MySQL:

- Velocidad al realizar operaciones, siendo uno de los gestores con mayor rendimiento [14].
- Baja corrupción de los datos.
- Bajo consumo, pudiendo ser instalado en una maquina con bajos recursos técnicos.
- Conectividad y seguridad.

En una aplicación web, es importante almacenar la información, pero también es muy

importante mostrarla. Para ello, se necesita un lenguaje de servidor que interprete las solicitudes del usuario, consulte la base de datos, procese la información y finalmente la devuelva al usuario. Para otorgar este dinamismo, MySQL generalmente es usado con: PHP, Python, ASP.Net, etc.

6.3.4. JAVASCRIPT

JavaScript es un lenguaje de programación interpretado, es decir no requiere de compilación ya que es analizado por un navegador web. En pocas palabras JavaScript está diseñado para funcionar en un browser o navegador web. Aunque también es posible usar este lenguaje en un servidor web, mediante el uso de Node.js. [15]

Es un lenguaje orientado a objetos mediante el uso de prototipos (en vez de clases), es débilmente tipado (no es necesario especificar el tipo de dato que se almacena en una variable) y su principal función es añadir interactividad a las aplicaciones web, pudiendo realizar animaciones, verificar formularios, solicitar peticiones al servidor de manera asíncrona, modificar el DOM (Document Object Model) de la página web, etc.

JavaScript también es usado para el control de eventos [16], es decir, poder detectar las pulsaciones de teclado, clicks, movimientos del mouse, webcam, etc. Esto con el fin de aportar más dinamismo a una aplicación web y ayudar a la experiencia de usuario. Para la inserción de este lenguaje es necesario agregar las etiquetas `<script></script>` en un documento HTML o también es posible agregarlo mediante un archivo con extensión .js (`<script src="mijavascript.js"></script>`).

Por último es importante mencionar que es un lenguaje que goza de una comunidad inmensa, ayudando constantemente con librerías y frameworks para facilitar tareas específicas y hacerlo más potente. Hoy en día es usado por la mayoría de sitios web, incluidas empresas tan importantes como: Google, Facebook, Amazon, etc.

6.3.5. AJAX

El término AJAX es un acrónimo de (Asynchronous JavaScript and XML), es una tecnología de desarrollo web que permite crear aplicaciones web de manera más dinámica y asíncrona. [17]

AJAX surgió para agilizar las consultas hechas al servidor, ya que de manera tradicional, realizar una consulta requiere que el servidor web genere una recarga total del sitio web para poder enviar todos los datos solicitados por el usuario, afectando de esta manera el rendimiento y la usabilidad de una aplicación o página web.

Para solucionar este problema, y poder solicitar información de secciones específicas de una aplicación web, sin la necesidad de recargar el sitio, es necesario el uso de una

tecnología asíncrona, que permita realizar esta tarea en segundo plano y sea imperceptible para el usuario. Siendo necesario el uso de Javascript para realizar peticiones al servidor HTTP y recibir la información solicitada. Dicha información es devuelta en formato XML (eXtensible Markup Language) o JSON (JavaScript Object Notation), donde se encuentra “comprimida” para agilizar las respuestas y pueda ser entendida por Javascript para hacer las modificaciones necesarias en el DOM (Document Object Model), que es donde se encuentra la estructura HTML y el diseño CSS (ver **Figura 6-2**).

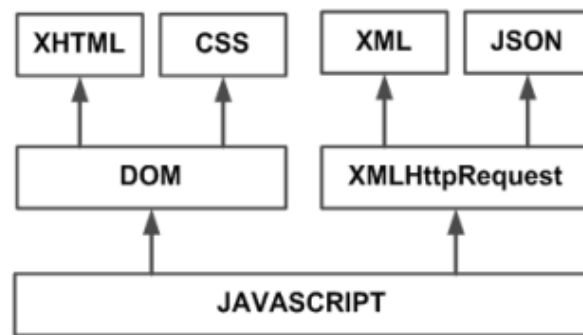


Figura 6-2: Tecnologías Ajax. Fuente: [18].

En la **Figura 6-3** se observa el modelo tradicional frente a AJAX. El modelo tradicional es completamente funcional y a día de hoy es ampliamente usado, pero en proyectos que requieren mayor uso de solicitudes al servidor (Buscadores, chats, juegos, estadísticas en tiempo real, etc.), es altamente recomendable el uso de tecnologías asíncronas [19] como AJAX, Comet, WebSocket, etc.

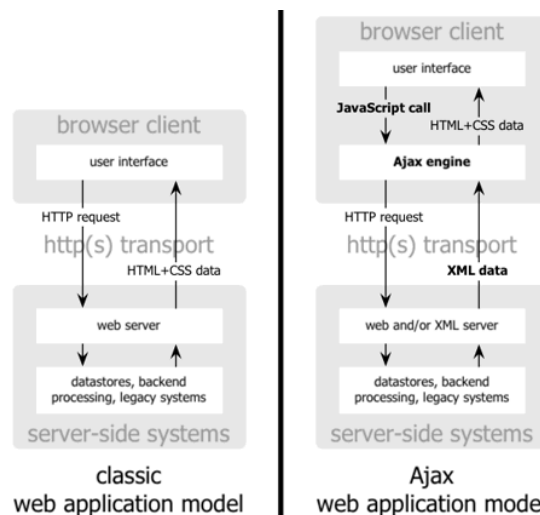


Figura 6-3: Modelo tradicional vs modelo Ajax. Fuente: [18].

6.3.6. CSS

CSS es el acrónimo de (Cascading Style Sheets) es el lenguaje utilizado para la presentación de documentos HTML o XML, otorgando a los diseñadores un control más conciso en el proceso de desarrollo.

Es uno de los lenguajes base del diseño web y que posee un estándar por parte del W3C (World Wide Web Consortium). Se divide en tres niveles: CSS1, CSS2.1 y CSS3. Donde CSS1 es obsoleto, CSS2.1 es recomendado y CSS3, desarrollado más reciente y por ende, en proceso de ser estandarizado [20]. Con CSS es posible el diseño gráfico de una página o aplicación web, permitiendo realizar animaciones, gráficos, tablas, etc. También es usado para la adaptación de sitios web, para lograr que un proyecto se vea de la mejor manera en todas las resoluciones posibles [21].

Por último, CSS otorga varias ventajas en el desarrollo web, siendo la principal, el poder separar el diseño de la maquetación HTML [22], ahorrando esfuerzo, tiempo y ayudando a tener código más limpio y estructurado. Además, se pueden crear archivos CSS específicos para cada sección de la página, ayudando de esta manera, a tener un menor consumo de ancho de banda en el servidor y acelerando la velocidad de carga del sitio web. [23]

7. DISEÑO METODOLÓGICO

7.1. POBLACIÓN MUESTRA

La empresa Gmetrologia SAS ubicada en la CRA 9 #48-40. Oficina 401, está conformada por:

- 1 Gerente General.
- 1 Director de laboratorio.
- 2 Secretarias.
- 1 Metrólogo.
- 2 Técnicos.

De los cuales el proyecto está dirigido al metrólogo, director de laboratorio y técnicos.

7.2. EVALUACIÓN DEL PROYECTO

Se evaluó el proyecto de acuerdo al modelo de calidad del software ISO/IEC 9126, que cuenta con los siguientes 6 factores:

7.2.1. FUNCIONALIDAD

Es la capacidad de la aplicación para proporcionar las funciones previamente declaradas cuando se usa de maneras específicas.

7.2.2. FIABILIDAD

Es la capacidad de la aplicación para mantener prestaciones preestablecidas cuando se usa bajo condiciones especificadas, ejemplo: tolerancia a fallos, capacidad de recuperación, cumplimiento de fiabilidad, etc.

7.2.3. USABILIDAD

La capacidad de la aplicación para ser entendida, usada y ser amigable para el usuario, cuando es usada en condiciones especificadas.

7.2.4. EFICIENCIA

Correcto funcionamiento de la aplicación relativo a la cantidad de recursos usados, ejemplo: Comportamiento temporal (tiempos de proceso, tiempos de respuesta), utilización de recursos (hardware), etc.

7.2.5. MANTENIBILIDAD

La capacidad de la aplicación para ser modificada, podría incluir corrección de errores, mejoras, adaptaciones, cambios de entorno, requisitos y especificaciones.

7.2.6. PORTABILIDAD

La capacidad de la aplicación para ser transferida de un entorno a otro, ejemplo: Adaptabilidad (dispositivos móviles, tabletas, PC, etc.), instalación (capacidad de la aplicación para ser instalada en otro medio).

8. EJECUCIÓN DEL PROYECTO

La ejecución del proyecto se divide en tres ítems principales: Requisitos de la aplicación, arquitectura de la aplicación y diagrama de casos de uso. Estos se incluyen con el fin de explicar cómo se realizó la aplicación y del porqué fueron parte fundamental en el desarrollo de la misma. También se incluyen más ítems explicando la funcionalidad de las secciones de la aplicación, en parte heredadas del diagrama de casos de uso para poder explicar con más detalle la funcionalidad de la aplicación.

8.1. LEVANTAMIENTO DE REQUERIMIENTOS

Se utilizaron 2 técnicas para el levantamiento de requerimientos: Etnografía y Modelado. [24]

8.1.1. ETNOGRAFÍA

La etnografía es una técnica de investigación que mediante la observación y entrevistas el analista o investigador se sumerge, para entender los requerimientos de un sistema, organización o unidad social. En este proceso, el analista observa el trabajo a diario donde anota las tareas en las que el personal se encuentra involucrado y realiza una recopilación de datos, que permite mostrar muchas de las actividades de los usuarios y procesos que generalmente son desconocidos al tratarse de otro entorno.

8.1.2. MODELADO

El modelado como técnica de levantamiento de requerimientos, se basa en realizar diagramas de flujo, UML, modelos entidad relación, etc. Esto con el fin de facilitar el entendimiento entre la población muestra, proveyendo de un medio visual para contener grandes volúmenes de información necesario para expresar y llevar a cabo los requerimientos del proyecto.

8.1.3. EJECUCIÓN LEVANTAMIENTO DE REQUERIMIENTOS

En la primera técnica se observó una semana normal de trabajo del metrólogo y técnicos, de esta manera se hizo un esquema del orden de trabajo que ellos tenían para implementarlo en el proyecto, con el fin de generar un ambiente más familiar de trabajo. Se realizaron preguntas para determinar posibles maneras de optimizar el trabajo y si era posible llevarlo a cabo sin alterar los resultados de la calibración, también se cuestionó sobre los métodos y procesos que realizaban, para ir haciendo un modelo más estructurado.

En la segunda técnica se hizo un modelado de la aplicación por medio de diagramas de caso de uso, esto hace que el personal autorizado se adapte a la aplicación de una manera más rápida y su curva de aprendizaje sea mucho más eficiente, también se pretende que con el modelado, se simplifique la detección de errores y se promueva la solución de los mismos, más eficientemente.

8.2. DISEÑO DE LA PLATAFORMA

Para el diseño de la plataforma se hicieron las siguientes secciones:

8.2.1. INTERFACE DE CONEXIÓN CON LA BASE DE DATOS

Es la parte inicial y primer acceso a la aplicación, en esta parte se determina como la aplicación se comunicará con la base de datos y cuál será el usuario administrador, al que se le otorgara automáticamente todos los permisos de la aplicación, pudiendo de esta manera, crear nuevos usuarios, tener total control de la plataforma y otorgar permisos. El proceso de instalación es bastante intuitivo sin llevar a confusiones, puesto que en su totalidad se hizo en un entorno gráfico.

8.2.2. INTERFACE DE ACCESO

Será la primera interfaz que vea el usuario, en esta parte, el usuario ya debe tener los datos de acceso suministrados por el administrador, quien previamente debió crear y otorgar permisos para su cuenta correspondiente. Una vez el usuario accede a la aplicación se le genera una sesión, la cual tiene acceso a ciertas partes de la plataforma según lo haya establecido el administrador.

8.2.3. ENTORNO DE APLICACIÓN

Es como tal el ambiente de trabajo del usuario, para este proyecto se manejaron 4 tipos de usuarios: Administrador, director de laboratorio, metrólogo y técnico.

Dependiendo del rol que se tenga, la aplicación muestra un entorno con diferentes opciones a excepción del administrador, quien siempre tiene acceso a la totalidad de la plataforma. De esta manera el metrólogo tiene acceso a crear certificados y revisarlos, también puede crear clientes y editarlos. El director de laboratorio puede crear patrones, clientes y certificados, además de poder editarlos. Por último el técnico puede crear certificados y clientes.

8.2.4. SELECCIÓN DE EQUIPO MÉDICO A CALIBRAR

Este proceso es llevado a cabo por un usuario con rol técnico, en este entorno, habrá una interfaz con los equipos que calibra la empresa Gmetrologia S.A.S.

Dependiendo del equipo que el técnico tenga que calibrar, él debe seleccionar en la plataforma dicho equipo, donde se le abrirá una nueva interfaz con los campos que debe llenar para el proceso de calibración de ese equipo en específico.

8.2.4.1. INTERFACE DE ADQUISICIÓN SEGÚN EL EQUIPO

Esta interfaz es única por equipo, debido a que el proceso y los datos de calibración son diferentes para cada tipo de equipo. Pero, siempre comparten algo en común, y es que se debe tener un diseño que permita el ingreso de los datos arrojados por el instrumento calibrador que posteriormente digitará el técnico.

Para ello se desarrolló un diseño amigable al técnico, y que en la mayor medida posible, ayude a que los datos sean íntegros y no sean propensos a alteraciones. Puesto que cada dato es indispensable para el proceso de calibración y la corrupción de cualquiera de ellos puede llegar a afectar en gran medida el resultado del certificado.

8.2.4.2. VALIDACIÓN DE DATOS

Es obligatorio contar con un sistema que valide los campos de ingreso de datos, puesto que la integridad de los datos es un factor muy importante en la elaboración del certificado. Para ello, se implementó un algoritmo que valide el campo de acuerdo al dato que el técnico debe ingresar. Es decir, si el técnico ingresa un dato que el algoritmo identifique como incorrecto se le mostrara un mensaje para que rectifique su error hasta que el algoritmo valide el campo, este proceso se hace cada vez que el usuario digite un dato para que el error sea corregido en tiempo real.

De esta manera se garantiza que el sistema informe de posibles errores que cometa el técnico, y que la corrupción de los datos sea más difícil de llevar a cabo.

8.2.4.3. PROCESAMIENTO DE DATOS

Este proceso, al igual que la validación de campos, ocurre en tiempo real. Primero se ejecuta el sistema de validación para verificar que sea un dato correcto y que se pueda procesar, si el proceso de validación es correcto, el dato es procesado mediante otro algoritmo, este algoritmo es único por equipo. En otras palabras es una ecuación matemática por equipo, que provee la empresa Gmetrologia S.A.S y que emplea para el proceso de calibración.

8.2.4.4. GENERACIÓN DE CERTIFICADO

Es el penúltimo paso para concluir con el certificado y el más importante, pues en este se decidirá el funcionamiento del equipo que se está calibrando. El técnico tendrá dos opciones: ver el certificado y guardar el certificado. En la primera opción el observa que todo esté bien y que el certificado cumpla con la norma NTC/ISO-IEC-17025, si determina que el resultado de la calibración es la correcta y que cumple tanto con los requerimientos de la norma y la empresa, debe elegir la segunda opción.

Para ello, se desarrolló un sistema que compare el resultado del algoritmo de procesamiento de datos con los datos que la empresa provee, con el fin de realizar el dictamen del equipo calibrado. Por lo tanto, el sistema debe entregar todos estos datos de tal manera que un humano los entienda y que en el proceso se determine si el equipo calibrado está apto para seguir funcionando o si por el contrario debe darse de baja.

Se diseñó una plantilla que está incluida en la interfaz de adquisición según el equipo, la cual el técnico no puede modificar ni ver. Puesto que está hecha desde código y otorgara mayor fidelidad al certificado. En la cual, están integrados los datos anteriormente mencionados para que el cliente entienda el estado de su equipo y dar por finalizado la realización del certificado.

8.2.4.5. ALMACENAMIENTO BASE DE DATOS

Es el último paso y el que da inicio al sistema de gestión de los certificados. El certificado es almacenado en la base de datos de la aplicación y tiene los siguientes datos: Nombre del técnico que realizó la calibración, nombre del cliente, serie del equipo, nombre del equipo, marca del equipo, modelo del equipo, fecha de creación del certificado, fecha de la última modificación del certificado, estado de la certificación, ciudad del cliente, lugar de calibración, fecha de calibración del equipo, fecha de recepción del equipo, teléfono del cliente y archivo del certificado.

Estos datos almacenados sirven para una futura gestión y seguimiento de los certificados, también para eventualidades y para llevar un control estricto de las calibraciones que deben ser llevadas a cabo en un futuro.

8.2.5. INTERFACE DE RESPALDO

Esta sección se encarga de realizar los backups de la aplicación. Por tanto, se dispone de dos opciones: importar respaldo y generar respaldo.

Generar respaldo, se encarga de generar un archivo SQL con toda la información de la base de datos que se encuentra en dicho momento de la aplicación. Importar Respaldo, como su nombre lo indica, se encarga de importar un archivo SQL, para que la aplicación vuelva a un estado anterior, dependiendo de la información que contenga el archivo importado.

8.2.6. INTERFACE DE USUARIO

La interfaz de usuario, es aquella que se encarga de mostrar el perfil de usuario que se encuentra en la sesión, en dicho perfil está la información de acceso que también podrá ser modificada, según lo requiera el usuario. La información que puede modificar el usuario es el correo electrónico y la contraseña.

8.3. REQUISITOS DE LA APLICACIÓN

Para el desarrollo del proyecto es importante identificar los requerimientos de funcionamiento del mismo, a continuación se explica cada uno de estos requisitos para que la aplicación pueda funcionar en óptimas condiciones.

8.3.1. SERVIDOR

El servidor es la parte funcional más importante del proyecto, ya que en él es donde estará todo lo necesario para el funcionamiento del programa. A nivel de hardware, el servidor se puede contratar a empresas de cloud computing como Amazon, Google, etc. Pero para este proyecto en específico, se hace uso de un computador portátil con las siguientes especificaciones técnicas:

- **Marca:** Acer.
- **Modelo:** E5-575G-38B5.
- **Memoria RAM:** 4GB DDR4L SDRAM.
- **Procesador:** Intel Core i3 6100U.

- **Velocidad del Procesador:** 2.3 GHz (3M Cache, 2.30 GHz).
- **Núcleos:** Dual Core (2 Núcleos).
- **Disco Duro:** 500GB HDD 5400 RPM.

La empresa Gmetrologia S.A.S decidió no optar por un servidor, ni en contratar un servicio de hosting, debido a sus políticas, y prefiere trabajar el proyecto de manera local, con la opción de que en un futuro se pueda migrar la aplicación a un servidor, si así lo requieren.

8.3.2. DISTRIBUCIÓN XAMPP

XAMPP es un software que permite instalar en cualquier sistema operativo, Apache HTTP Server, MySQL y PHP. Estos programas, permiten la ejecución de la aplicación y como tal del funcionamiento de la misma. El programa es instalado en el sistema operativo Windows 10 y en el servidor mencionado anteriormente.

8.3.2.1. APACHE HTTP SERVER

Este programa sirve para que transmita e interprete todas las peticiones del protocolo HTTP que se llevan a cabo en la aplicación web, fundamental para aplicaciones de este tipo ya que es la manera en que se comunica el navegador web con el servidor y así, procesar las solicitudes que se envían desde el usuario.

Se decide usar apache server ya que es de código abierto y es usado en el 70% [25] de todas las aplicaciones web, indicando que es un programa estable, seguro y optimo, además es multiplataforma, pudiendo instalarse en cualquier sistema operativo que se desee.

8.3.2.2. MySQL

En toda aplicación web es importante contar con un gestor de base de datos, para el almacenamiento de la información. MySQL, es un gestor de base de datos ampliamente usado en proyectos web, debido a que es de código abierto. Además, su robustez permite abarcar proyectos de todo tipo, pues su rendimiento y velocidad es de los mejores para los bajos recursos que consume. Por estas razones, se decide usar este gestor en el proyecto.

MySQL por defecto crea un usuario “root” sin contraseña, dicho usuario es el administrador principal de MySQL, quien podrá manipular la información, crear bases de datos, tablas, etc. Por tanto, es indispensable cambiar el nombre de usuario y otorgarle una contraseña, con el fin de tener asegurado el gestor de base de datos y por ende, la información.

Para poder interactuar con la base de datos y servir paginas dinámicas es necesario el uso de un lenguaje de programación que trabaje del lado del servidor web, para este proyecto se usó PHP, el cual es un lenguaje potente, muy usado y altamente recomendado para desarrollo web. Además, tiene una muy buena documentación junto con una enorme comunidad que constantemente está mejorando dicho lenguaje.

Para el desarrollo de la aplicación se utilizó el patrón de arquitectura de software: modelo-vista-controlador, en el cual se utilizan 5 clases principales las cuales emplean la mayor parte de la funcionalidad del programa (ver **Figura 8-1**).



Se decide implementar esta arquitectura para separar la lógica del diseño, donde se potencie la facilidad de mantenimiento, reutilización de código y la separación de conceptos. De esta manera se consigue mayor orden en el código de programación permitiendo mayor escalabilidad a medida de que el programa se desarrolla. Las clases principales de la arquitectura o capas son “Model.php”, “View.php” y “Controller.php”, además de estas, se hace uso de 5 clases más, llamadas: “Bootstrap.php”, “Database.php”, “Globales.php”, “Request.php” y “Session.php”, las cuales serán explicadas a continuación.

8.4.1. “BOOTSTRAP.PHP”



Figura 8-2: Clase “Bootstrap.php”

La clase “Bootstrap.php” (ver **Figura 8-2**), se encarga de recibir como argumento la instancia de la clase “Request.php”, en donde está contenido el controlador, el método del controlador y argumentos. De esta manera la clase “Bootstrap.php”, principalmente se encarga de revisar si la aplicación se encuentra instalada y de revisar si existe una sesión iniciada, si estas 2 condiciones se cumplen, se instancia el controlador que se mencionó anteriormente y llama el método declarado. Si existen argumentos, serán enviados como parámetros.

La clase “Bootstrap.php”, es llamada o instanciada desde el archivo “index.php”, siendo esté archivo el que siempre se ejecuta cada vez que se accede a una URL de la aplicación.

8.4.2. “REQUEST.PHP”

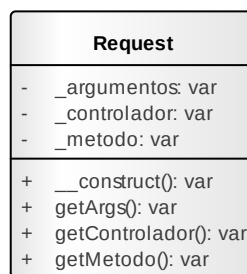


Figura 8-3: Clase “Request.php”

La clase “Request.php” (ver **Figura 8-3**) es la encargada de procesar la URL, cada vez que se hace una solicitud a la aplicación, para ello, la clase divide la URL y la agrega en un array. Este array contiene la información del controlador a usar, el método del controlador y argumentos (si existen), finalmente estos valores son asignados a sus variables correspondientes de la clase (`_argumentos`, `_controlador`, `_metodo`), que posteriormente serán usadas por la clase “Bootstrap.php”.

8.4.3. “SESSION.PHP”



Figura 8-4: Clase “Session.php”

La clase “Session.php” (ver **Figura 8-4**), es la encargada de administrar las sesiones de la aplicación, por medio de esta clase se puede verificar una sesión, crear una sesión, editar una sesión y eliminar una sesión. Además tiene la función de restringir partes de la aplicación por medio de los privilegios del usuario.

Por último, esta clase se encarga de revisar si existe una instalación de la aplicación y retornar “true” si es el caso.

8.4.4. “DATABASE.PHP”

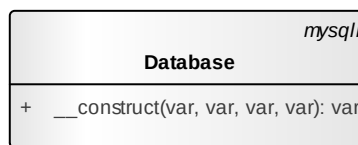


Figura 8-5: Clase “Database.php”

La clase “Database.php” (ver **Figura 8-5**), es la que se encarga de inicializar la conexión con MySQL, esta conexión se realiza por medio de la clase *mysqli* de php, permitiendo enviar los datos necesarios para la conexión, como son el host, user, pass y el nombre de la base de datos. Es decir es el puente entre la base de datos y la aplicación.

Cuando sucede un error de conexión, la clase automáticamente detiene la continuidad del programa y muestra que error es el que ha sucedido. Los errores son suministrados por la clase mysqli en forma de código o descripción según sea requerido.

8.4.5. “GLOBALES.PHP”

Globales
+ <u>checkDB(): var</u>
+ <u>cleanInput(var): var</u>
+ <u>equalhashValue(var, var): var</u>
+ <u>esc_url(var): var</u>
+ <u>get_home_url(): var</u>
+ <u>get_timebog(var): var</u>
+ <u>get_url(): var</u>
+ <u>get_userrole(var): var</u>
+ <u>hashValue(var): var</u>
+ <u>is_ajax(): var</u>
+ <u>is_serialized(var, var): var</u>
+ <u>is_ssl(): var</u>
+ <u>maybe_serialize(var): var</u>
+ <u>maybe_unserialize(var): var</u>
+ <u>mbstring_binary_safe_encoding(var): var</u>
+ <u>moreYear(var): var</u>
+ <u>redirect(var): var</u>
+ <u>remove_accents(var): var</u>
+ <u>sanitize_title(var, var): var</u>
+ <u>sanitize_title_with_dashes(var, var): var</u>
+ <u>seems_utf8(var): var</u>
+ <u>wp_check_password(var, var): var</u>
+ <u>wp_hash_password(var): var</u>

Figura 8-6: Clase “Globales.php”

La clase “Globales.php” (ver **Figura 8-6**), contiene todas las funciones globales de la aplicación, estas funciones pueden ser llamadas desde cualquier otro lugar del programa, son funciones que permiten la reutilización de código y ayudan al correcto funcionamiento de la aplicación. Algunas de las funciones son: encriptar contraseñas, obtener la fecha, obtener el rol de un usuario, sanear variables, etc.

8.4.6. “CONTROLLER.PHP”

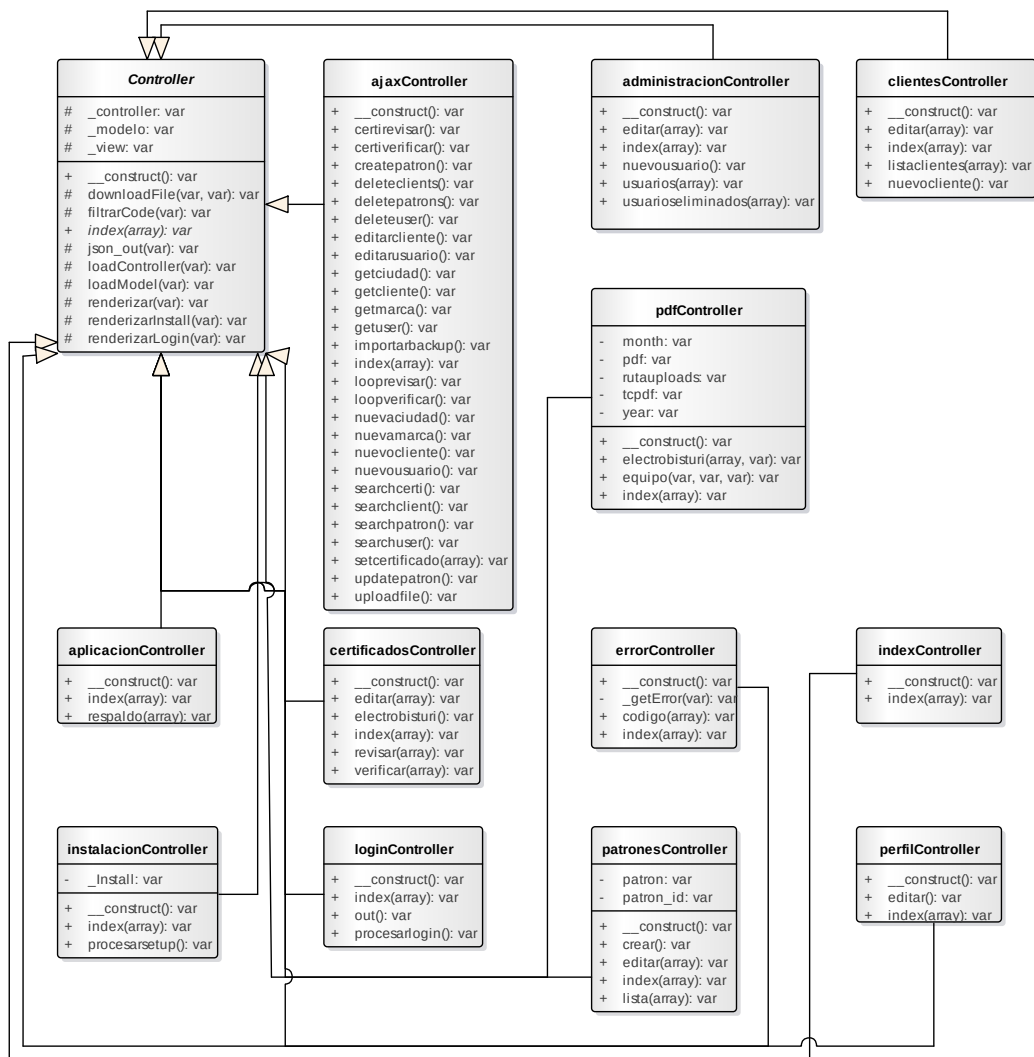


Figura 8-7: Clase “Controller.php”.

La clase “Controller.php” (ver **Figura 8-7**), es una clase principal o clase “padre”, de donde se extienden otras clases “hijas” las cuales hereden sus métodos, éstas clases “hijas” a su vez poseen sus propios métodos o funciones y en ellas radica toda la funcionalidad de la aplicación.

La clase “Controller.php” hace referencia al controlador de la arquitectura de software MVC y por ende es el encargado de realizar peticiones a la clase “Model.php” o “View.php”, funcionando como un intermediario entre estas dos, Siendo así la clase más usada.

8.4.7. “MODEL.PHP”

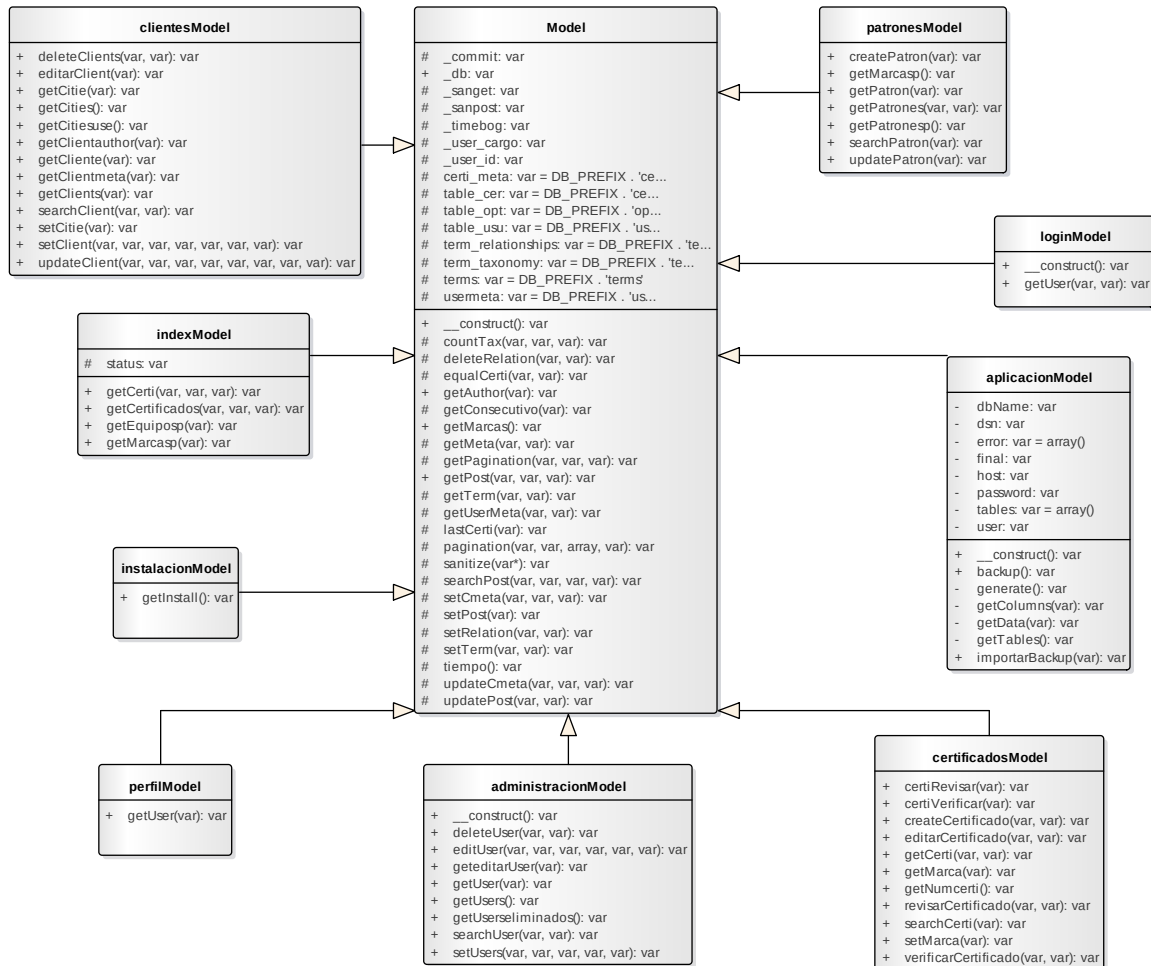


Figura 8-8: Clase “Model.php”

La clase “Model.php” (ver **Figura 8-8**), tiene similar estructura con “Controller.php”, puesto que es una clase “padre” y de ésta heredan múltiples clases “hijas”. La principal funcionalidad de ésta clase es interactuar con todo lo relacionado a la base de datos. Es decir, se encarga de obtener, eliminar, actualizar y crear información según sea necesario.

La clase “Database.php”, es instanciada cada vez que se hace uso de ésta clase.

8.4.8. “VIEW.PHP”



Figura 8-9: Clase “View.php”

La clase “View.php” (ver **Figura 8-9**), se encarga de mostrar la interfaz gráfica que vera el usuario, según el controlador lo requiera. De tal forma que los métodos de esta clase son los encargados de renderizar archivos HTML, CSS y JavaScript, según sea la vista a mostrar. Cada sección o controlador tiene sus propios diseños y esta clase permite la visualización de ellos.

8.5. DIAGRAMA DE CASOS DE USO

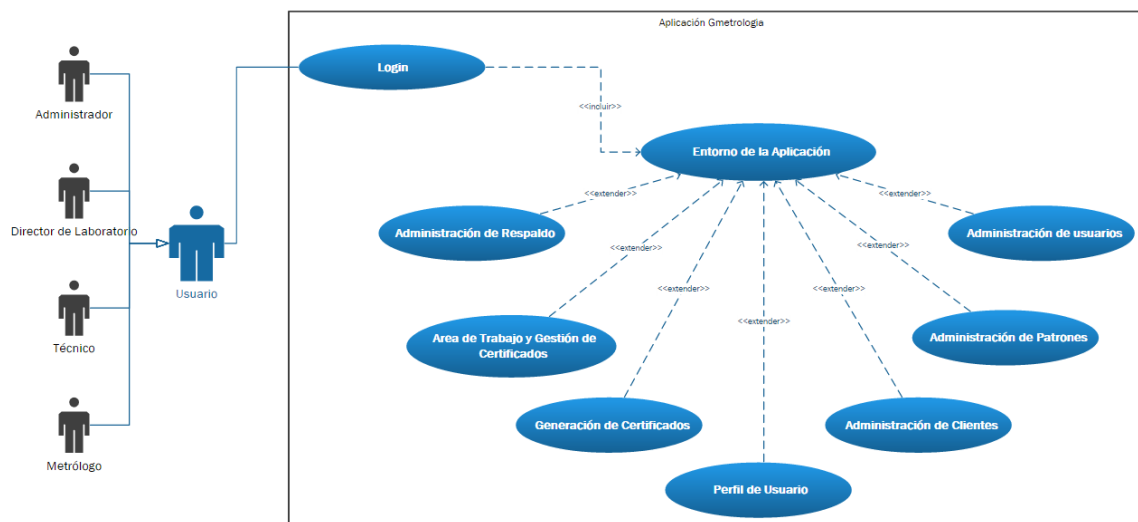


Figura 8-10: Diagrama de casos de uso de la aplicación.

De acuerdo a las técnicas de levantamiento de requerimientos, se estableció el diagrama de la **Figura 8-10**, como resultado de la primera técnica (etnografía). En esta técnica se observó el proceso de trabajo durante una semana y se realizó el primer diagrama de casos de uso para empezar el desarrollo del proyecto, en el cual, se integran de manera general todas las secciones de la aplicación, a continuación, se explicará de manera más detallada dichas secciones.

8.6. INTERFAZ DE INSTALACIÓN

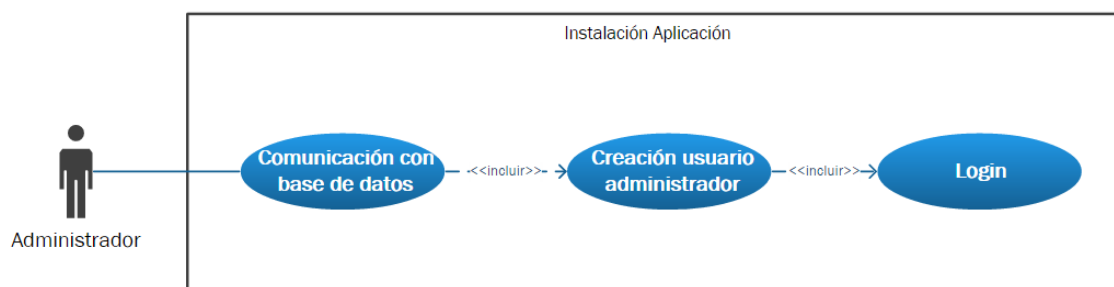


Figura 8-11: Diagrama de caso de uso para la instalación.

En el proceso de instalación de la aplicación, el administrador debe cumplir con el diagrama de casos de uso mostrado en la **Figura 8-11**, al acceder a la aplicación, ésta verifica si se encuentra instalada, para ello, revisa si existe información en la base de datos o si existe un usuario administrador, si alguna de las dos no se cumple, automáticamente se observa la interfaz de instalación, donde se muestra los datos que el administrador debe suministrar para la correcta instalación de la aplicación. Finalizada la instalación, el programa redirige al sistema de login, para dar acceso al software.

Cada vez que se hace un acceso a la aplicación o se realiza una solicitud al programa, inmediatamente se verifica que el programa se encuentre correctamente instalado y si no es el caso, el programa redirige de nuevo a la interfaz de instalación.

G METROLOGIA S.A.S.

Información Necesaria

A continuación deberás introducir los detalles de conexión a tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web.

Nombre de la base de datos	gmetrologia	El nombre de la base de datos en la que quieres ejecutar Gmetrologia.
Nombre de Usuario	root	Tu usuario de MySQL
Contraseña	contraseña	...y tu contraseña de MySQL
Servidor de la base de datos	localhost	Deberías poder acceder desde tu servidor web si localhost no funciona.
Prefijo de tabla	gm_	Si quieres ejecutar varias instalaciones de Gmetrologia en una sola base de datos cambia esto.

Figura 8-12: Interfaz de instalación.

La interfaz de instalación como se observa en la **Figura 8-12** se compone de 5 campos, los cuales cumplen con la función de obtener los datos para conectarse exitosamente a la base de datos MySQL. Estos campos son: “Nombre de la base de datos”, “Nombre de Usuario”, “Contraseña”, “Servidor de la base de datos” y “Prefijo de tablas”. Al final de este proceso se crea un archivo “config.php” que contendrá estos datos para el funcionamiento de la aplicación. Este archivo solo se crea si la aplicación pudo comunicarse con el servidor MySQL, la base de datos y además si pudo crearse el usuario administrador. Es decir si no se efectuó ningún error durante la instalación.

Para suministrar los datos de MySQL y poder ejecutar la aplicación, el administrador previamente debió ejecutar la instalación de un servidor web, junto con la instalación del gestor de base de datos MySQL, esto es explicado en el ítem de **“REQUISITOS DE LA APLICACIÓN”**.

gmetrologia gm_terms term_id : bigint(20) unsigned name : varchar(200) slug : varchar(200) term_group : bigint(10)	gmetrologia gm_certificados ID : bigint(20) unsigned post_author : bigint(20) unsigned post_metrologo : bigint(20) unsigned post_director : bigint(20) unsigned post_date : datetime post_date_gmt : datetime post_content : longtext post_title : text post_excerpt : text post_nit : varchar(30) post_status : varchar(20) post_password : varchar(20) post_name : varchar(255) post_modified : datetime post_modified_gmt : datetime post_parent : bigint(20) unsigned post_type : varchar(20)	gmetrologia gm_usuarios ID : bigint(20) unsigned user_login : varchar(60) user_pass : varchar(255) user_nicename : varchar(30) user_email : varchar(100) user_url : varchar(100) user_registered : datetime user_activation_key : varchar(255) user_status : int(11) user_cargo : int(11) display_name : varchar(255)
gmetrologia gm_term_relationships object_id : bigint(20) unsigned term_taxonomy_id : bigint(20) unsigned term_order : int(11)		
gmetrologia gm_term_taxonomy term_taxonomy_id : bigint(20) unsigned term_id : bigint(20) unsigned taxonomy : varchar(32) description : longtext parent : bigint(20) unsigned count : bigint(20) unsigned		
gmetrologia gm_opciones option_id : bigint(20) unsigned option_name : varchar(191) option_value : longtext autoload : varchar(20)	gmetrologia gm_certificadosmeta meta_id : bigint(20) unsigned post_id : bigint(20) unsigned meta_key : varchar(255) meta_value : longtext	gmetrologia gm_usermeta umeta_id : bigint(20) unsigned user_id : bigint(20) unsigned meta_key : varchar(255) meta_value : longtext

Figura 8-13: Tablas de la aplicación Gmetrologia.

En el proceso de instalación, la aplicación además de crear el archivo “config.php”, debe crear las tablas. Estas componen la estructura de la base de datos y contienen la información que se ingresa, básicamente satisfacen cualquier necesidad de almacenamiento del proyecto.

8.6.1. ESTRUCTURA DE LAS TABLAS

La estructura está conformada por 8 tablas como se muestra en la **Figura 8-13** y se explican a continuación.

8.6.1.1. TABLA “certificados”

Esta tabla contendrá toda la información correspondiente a entradas de tipo “certificados”, “clientes” y “patrones”. Cada vez que es creada una entrada se le asigna automáticamente el autor que la creo, además de otros datos como lo son: Fecha de creación, fecha de modificación, contenido de la entrada, titulo, estatus de la entrada, etc. Estos datos son importantes para soportar la creación de cada certificado, cliente o patrón.

8.6.1.2. TABLA “certificadosmeta”

En esta tabla se inserta la información adicional que no corresponda a la tabla “certificados”. Para el caso de la aplicación, se inserta información de una entrada tipo cliente como correo, dirección, teléfono, etc. Esta información se relaciona al ID de la entrada a la cual se le quiera anexar dicha información.

8.6.1.3. TABLA “usuarios”

Esta tabla contiene las credenciales para el acceso a la aplicación, como es la cedula, correo electrónico, contraseña, nombre, cargo, y estatus.

8.6.1.4. TABLA “usermeta”

De manera similar a “certificadosmeta”, esta tabla contiene información adicional, relacionada al usuario, en dado caso que sea necesario agregar más datos del usuario.

8.6.1.5. TABLA “terms”

La tabla “terms”, contiene todas las categorías de la aplicación, como son: marcas de equipos, equipos y ciudades. Con el fin de categorizar las entradas de tipo “certificados”.

8.6.1.6. TABLA “term_relationships”

Esta tabla contiene todas las relaciones entre las ID de las taxonomías y las entradas creadas de la aplicación, mediante la ID de la entrada. Es decir, es la tabla que relaciona las categorías con las entradas.

8.6.1.7. TABLA “term_taxonomy”

En la tabla “term_taxonomy”, se consigna la clasificación de cada categoría mediante la relación entre la ID de la categoría y el nombre de clasificación, ejemplo: “ciudades”, “equipo”, “marca”. Con el fin de organizar las categorías.

8.6.1.8. TABLA “opciones”

Esta tabla contiene la información de variables generales de la aplicación, como la URL base, título de la aplicación, descripción de la aplicación, etc.

Continuando con la interfaz de instalación, a continuación se explican los campos requeridos para la comunicación con el gestor de base de datos MySQL.

8.6.2. NOMBRE DE LA BASE DE DATOS

Este campo es el encargado de indicarle al programa en que base de datos se desea realizar la instalación de la aplicación, éste nombre, es el de la base de datos creada previamente por el administrador, ya sea por medio de una aplicación externa que permita la administración de MySQL (phpMyAdmin), o por medio de conocimientos más técnicos, como es el uso de comandos por medio de la consola del sistema operativo que se esté usando.

8.6.3. NOMBRE DE USUARIO MYSQL

El nombre de usuario MySQL, es aquel que tiene los privilegios para administrar la base de datos. Es muy importante que este usuario tenga los permisos de crear, modificar y borrar tablas de la base de datos, puesto que la aplicación hará uso de este usuario para realizar la instalación de las tablas correspondientes de la aplicación (ver **Figura 8-13**), como también podrá crear, modificar y eliminar filas dentro de las tablas. El usuario por defecto es “root” y es altamente recomendable cambiar el nombre de este usuario para otorgar mayor seguridad ante un posible hackeo a la base de datos.

8.6.4. CONTRASEÑA MYSQL

Es la contraseña del usuario MySQL. Sin esta credencial es imposible hacer uso de los privilegios del usuario MySQL. Además de que es importante para la seguridad de la base de datos, debido a que si no tiene contraseña, cualquier persona que conozca el host y el usuario puede tener total acceso, pudiendo robar información o alterar la aplicación.

8.6.5. SERVIDOR DE LA BASE DE DATOS

Este campo es usado para indicar el servidor MySQL que hará conexión con la base de datos, generalmente se hace uso de “localhost” o 127.0.0.1, esto depende de quien gestione el servidor ya que también puede contener un puerto específico para la conexión. Cuando son servidores remotos por lo general también se hace uso de “localhost” y si no es el caso. Se debe poner en contacto con el administrador del servidor para que indique la credencial que establece la conexión con el gestor de base de datos MySQL.

8.6.6. PREFIJO DE TABLA

Este campo es necesario si se desea realizar varias instalaciones en la misma base de datos, ya sea para pruebas de mantenimiento o incorporación de nuevas funciones. Debido que al hacer una instalación de la aplicación, esta crea las tablas anteriormente mencionadas en **ESTRUCTURA DE LAS TABLAS** y por ende si se desea realizar una segunda instalación, no se podrá efectuar, ya que existen tablas con el mismo nombre.

Por tal motivo, si se desea realizar más de una instalación es importante definir un prefijo para crear tablas únicas en cada instalación.

8.6.7. INTERFAZ PARA LA CREACIÓN DE USUARIO ADMINISTRADOR



The screenshot shows the 'Instalación' (Installation) screen of the Gmetrologia S.A.S. application. At the top is the logo 'G METROLOGIA S.A.S.'. Below it, a welcome message states: '¡Bienvenido al proceso de instalación de Gmetrologia! Simplemente completa la información siguiente y podrás usar la plataforma'. The section is titled 'Información Necesaria' (Necessary Information). A note says: 'Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.' (Please, provide us with the following data. Don't worry, you can always change these settings later). There are three input fields: 1. 'Nombre de usuario:' (Username) with the value 'root' and a green 'Usuario Valido' (Valid User) button. A note below says: 'Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.' (Usernames can only have alphanumeric characters, spaces, underscores, hyphens, dots, and the @ symbol). 2. 'Password:' with masked characters '****' and a 'Mostrar' (Show) button. A note below says: 'Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.' (Important: You need this password to access. Please, save it in a safe place). Another note below says: 'Un consejo: La contraseña debe tener al menos, ocho caracteres de longitud. Para hacerla más fuerte, utiliza mayúsculas y minúsculas, números y símbolos como ! " ? \$ % ^ & .' (A tip: The password must have at least eight characters in length. To make it stronger, use uppercase and lowercase letters, numbers, and symbols like ! " ? \$ % ^ & .). 3. 'Tu correo electrónico:' (Your email) with an empty text box. A note below says: 'Comprueba bien tu dirección de correo electrónico antes de continuar.' (Check your email address carefully before continuing). At the bottom is a large 'Instalar' (Install) button.

Figura 8-14: Interfaz para la creación del usuario administrador.

Esta interfaz (ver **Figura 8-14**) es el proceso final para instalar la aplicación. En estos campos se requieren los datos del administrador de la aplicación, una vez se indiquen las credenciales del administrador y se oprima el botón instalar, la aplicación procederá a crear la estructura de las tablas ya mencionadas (ver **Figura 8-13**) y a insertar el usuario administrador en la tabla “usuarios”, el cual tendrá todos los privilegios de la aplicación.

8.7. INTERFAZ DE ACCESIBILIDAD

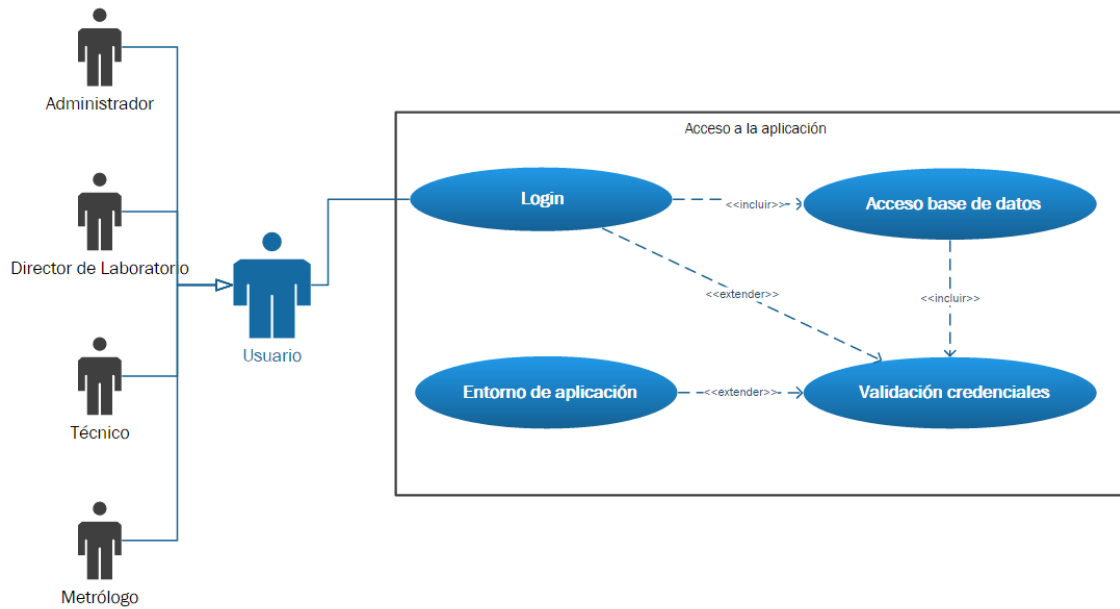


Figura 8-15: Diagrama de casos de uso para el acceso a la aplicación.

En el proceso de acceso a la aplicación, el usuario debe obligatoriamente pasar por la interfaz de “login” o acceso, en ésta interfaz, se obtienen los datos para hacer la validación de las credenciales, comparando esta información con la almacenada en la tabla “usuarios” de la base de datos, si estas credenciales son correctas, el usuario accede al entorno de la aplicación y si por el contrario, las credenciales son incorrectas, el usuario vuelve a la interfaz “login” como se muestra en la **Figura 8-15**.



Figura 8-16: Interfaz de accesibilidad

La interfaz de accesibilidad (ver **Figura 8-16**) es la que permite dar inicio a una sesión en la aplicación, esta sesión depende del usuario que está accediendo, ya que el software comprueba los datos ingresados en la base de datos y dependiendo el rol o cargo que le haya otorgado el administrador a dicho usuario, la sesión tendrá los privilegios del rol asignado.

Para poder acceder al entorno de la aplicación es obligatorio pasar por este proceso, es decir, todos los usuarios están obligados a pasar por esta interfaz.

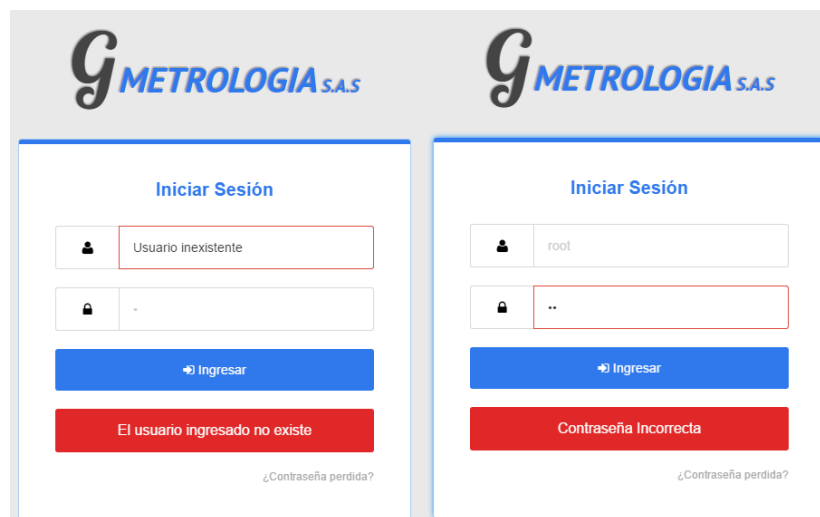


Figura 8-17: Inicio de sesión fallido

Para comprobar el rol e iniciar sesión, el programa verifica las credenciales suministradas por la interfaz, en la tabla “usuarios”. Estas credenciales constan de un password o contraseña y un usuario de acceso que puede ser: cédula, correo electrónico o el nombre de usuario (según lo asigne el administrador).

El programa comprueba que esta información le corresponda a un usuario existente, si existe tal usuario, la aplicación inicia sesión con el rol que esté relacionado para dicha información, si no existe, el programa mostrará un error indicando que el usuario no existe. Y si la información es incorrecta, es decir, el usuario existe pero la contraseña no es válida para ese usuario, se muestra un error indicando que la contraseña es incorrecta, tal cual muestra la **Figura 8-17**.

En dado caso que el usuario no recuerde su contraseña, éste tendrá que comunicarse con el administrador para que le asigne una nueva.

8.8. ENTORNO DE APLICACIÓN

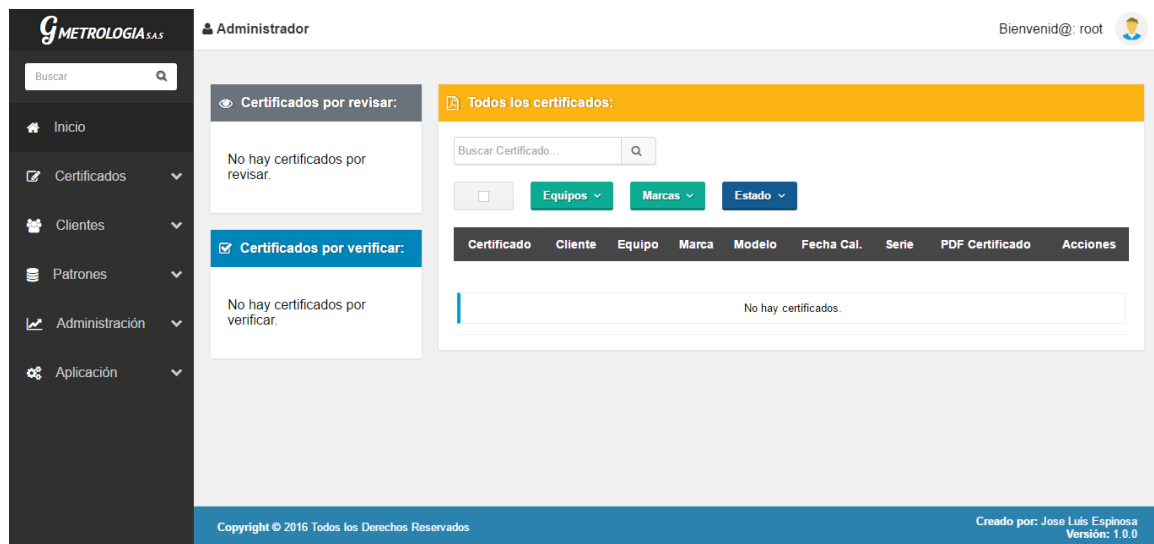


Figura 8-18: Entorno de aplicación.

El entorno de aplicación (ver **Figura 8-18**) es donde se tiene acceso a todo el programa, esta vista solo es accesible después de que cualquier usuario inicia sesión, de lo contrario solo se seguirá viendo la interfaz de accesibilidad (ver **Figura 8-16**).

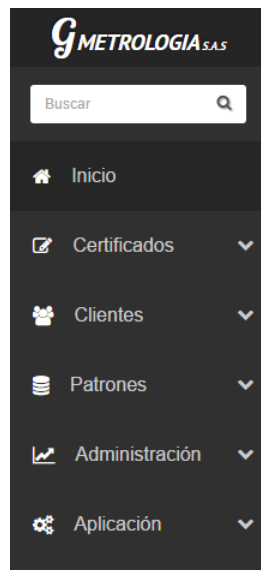


Figura 8-19: Secciones de la aplicación.

En la barra izquierda del programa (ver **Figura 8-19**) se encuentran todas las secciones del software, en estas secciones recarga toda la funcionalidad de la aplicación y se divide en 6 partes o secciones.

8.8.1. SECCIÓN INICIO

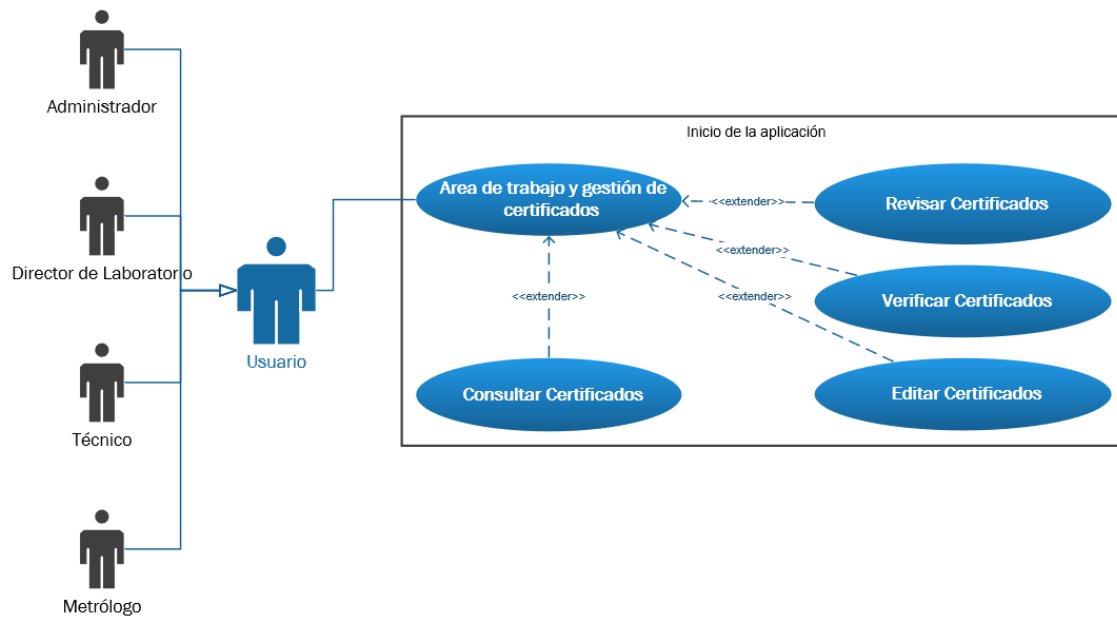


Figura 8-20: Diagrama de casos de uso para el inicio de la aplicación.

En la vista inicio el usuario tiene su área de trabajo y podrá hacer gestión de los certificados, como se muestra en la **Figura 8-20**. Esto depende del rol que posea el usuario que accede a la aplicación ya que no todos los usuarios tienen los mismos privilegios o trabajos por hacer.

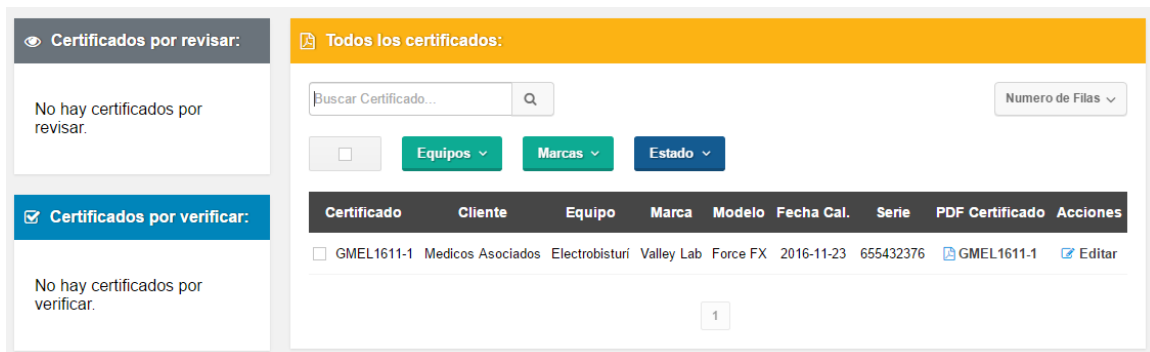


Figura 8-21: Vista Inicio.

También se muestran todos los certificados almacenados en la tabla “certificados” y que el usuario haya realizado, pudiendo filtrar los certificados por: equipos, marcas y el estado en el que se encuentre el certificado (ver **Figura 8-21**).

Si el usuario posee el rol de técnico, éste solo podrá observar los certificados que haya generado en su sesión. Todos los certificados que genere un técnico, tendrán como estado “revisar”, éste estado, es con el fin de que un usuario con rol metrólogo pueda revisar el certificado que haya creado cualquier usuario con rol técnico. A su vez el rol metrólogo, cuando revise un certificado, éste cambiara de estado a “verificar”. Por último todos los certificados con estado “verificar” tendrán que ser vistos por un usuario con rol director de laboratorio, finalizada la verificación del certificado, este cambiará a un estado de “verificado”.



Figura 8-22: Certificados por revisar y verificar.

En el campo certificados por revisar, aparecen todos los certificados con estado revisar, éste campo solo es visible por un usuario con rol metrólogo o superior, del mismo modo

sucede con el campo certificados por verificar, que solo es visible por un usuario con rol director de laboratorio o administrador. Estos campos muestran los certificados que están pendientes de revisar o verificar (ver **Figura 8-22**).

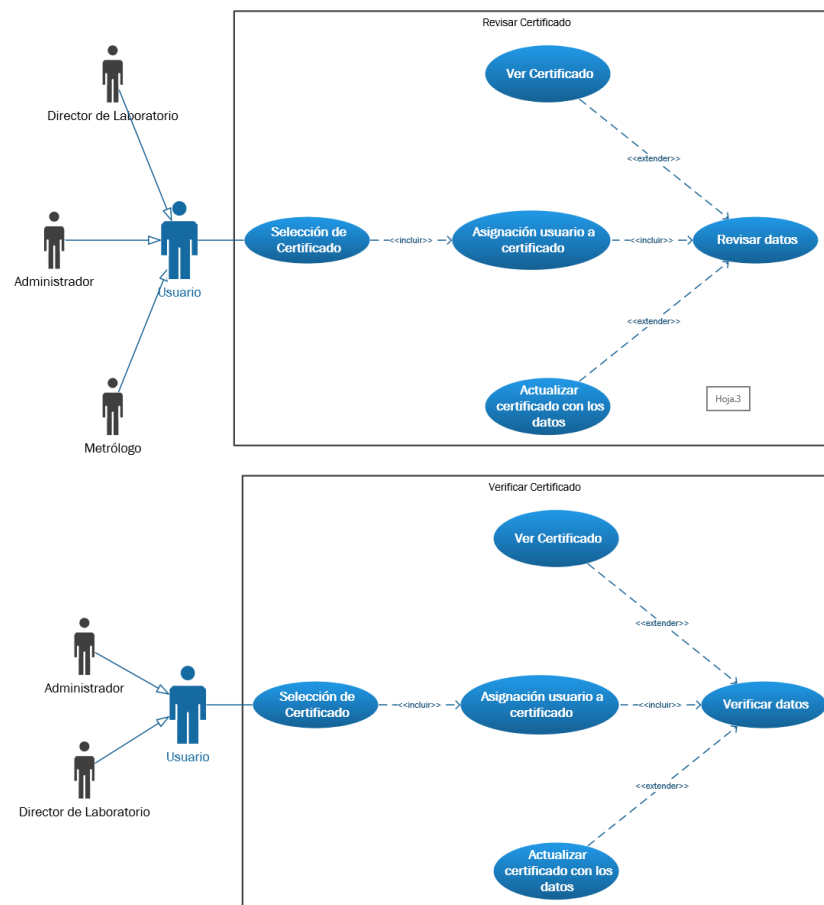


Figura 8-23: Diagrama de casos de uso para revisar y verificar certificados.

Una vez se oprime el botón de revisar o verificar, al certificado se le asigna el usuario que este en esa sesión, de esta manera el certificado pasara a tener hasta 3 autores diferentes (técnico, metrólogo y director de laboratorio), garantizando que solo un usuario pueda trabajar en ese certificado al mismo tiempo (ver **Figura 8-23**). De esta manera un certificado que ya haya pasado por revisión, solo podrá ser editado por el usuario que haya trabajado con el certificado, o si el certificado ya ha sido verificado, éste podrá ser editado por el administrador o por el director de laboratorio que lo verifico.

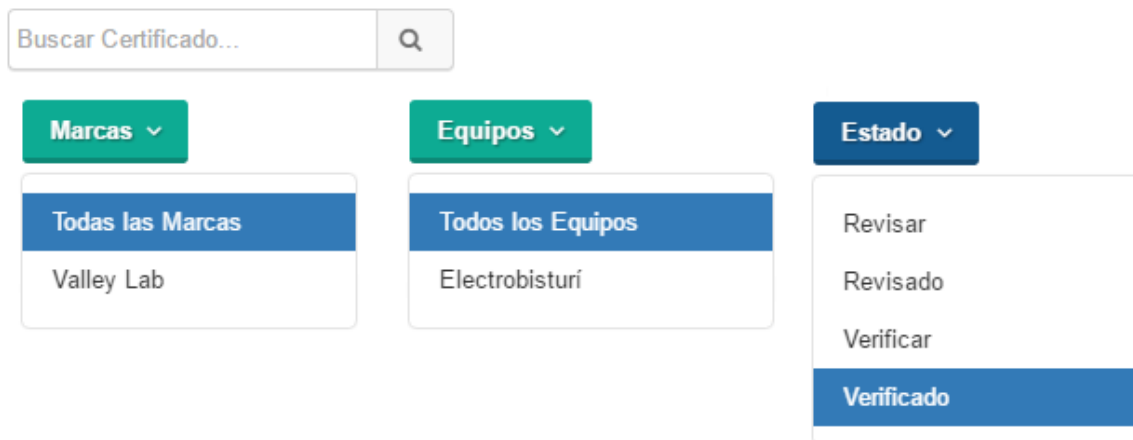


Figura 8-24: Opciones de filtrado.

Por último, en la vista inicio se puede filtrar y buscar certificados con diferentes estados, mediante el uso del buscador o mediante los menús que se muestran en la **Figura 8-24**. Esto con el fin de poder encontrar un certificado con rapidez y eficiencia.

8.8.2. SECCIÓN CERTIFICADOS

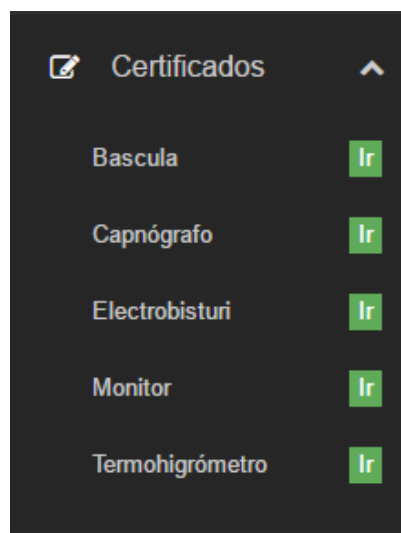


Figura 8-25: Sección certificados.

En esta sección se ingresa a la vista que contiene los formularios para la adquisición de datos que son necesarios para la realización del certificado, según el equipo seleccionado en el menú de la **Figura 8-25**.

A continuación, se muestra el diagrama de casos de uso para la creación de un certificado en la aplicación (ver **Figura 8-26**)

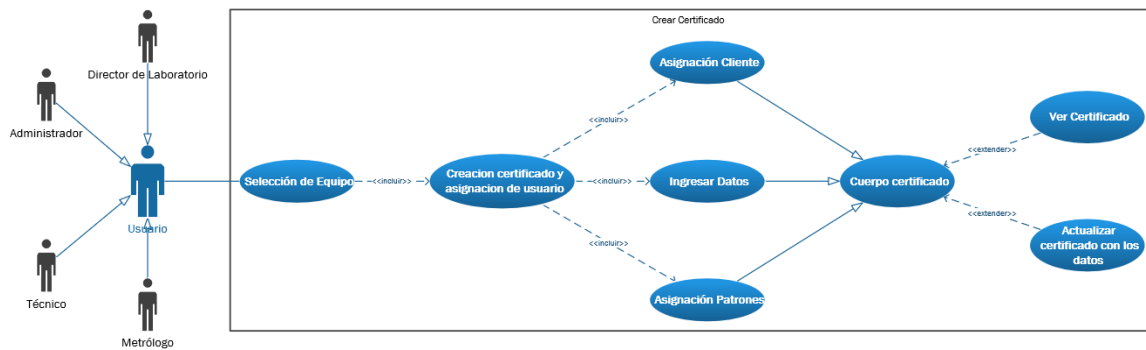


Figura 8-26: Diagrama de casos de uso para la creación de un certificado.

Para la creación de un certificado, principalmente se debe elegir el equipo al cual se le quiere hacer el certificado, inmediatamente después el software dirige la aplicación al entorno encargado para dicho equipo, almacenando en la tabla “certificados” una entrada de tipo “certificados”, estado “borrador” y asignando el usuario encontrado en la sesión. En éste certificado, automáticamente también es asignado un consecutivo que varía según el equipo y la fecha, éste consecutivo es con el fin de no tener redundancia de certificados y cumplir con la norma NTC ISO-IEC 17025 numeral 4.3.2.3.

Generado el consecutivo, el usuario ahora tiene que ingresar los datos del cliente, los datos arrojados por el simulador y también debe ingresar los patrones utilizados. Los patrones, son los certificados de los simuladores usados en el proceso de la calibración. A continuación se explica mejor todo el proceso de creación, mediante imágenes del entorno gráfico de un equipo electrobisturí.

8.8.2.1. INTERFAZ DE ADQUISICION DE DATOS

Figure 8-27 shows the graphical environment for creating a certificate for an electrobisturi equipment. The interface is a web application with a sidebar menu on the left and a main content area. The sidebar menu includes options like Inicio, Certificados, Bascula, Capnógrafo, Electrobisturi, Monitor, Termohigrómetro, Clientes, Patrones, Administración, and Aplicación. The main content area has two panels: 'Buscar Cliente' and 'Buscar Patrones'. The 'Buscar Cliente' panel shows a search for 'medicos' with suggestions like 'Medicos Asociados'. The 'Buscar Patrones' panel shows a search for 'electro' with suggestions for various equipment. The 'Generar Certificado Electrobisturi' panel shows a 6-step process, with the first step being 'Numero de Certificado y Datos del Cliente'. This step includes fields for 'Certificado N° (Obligatorio)', 'Nombre Cliente (Obligatorio)', 'Ciudad (Obligatorio)', 'Dirección (Obligatorio)', and 'Teléfono (Obligatorio)'. The 'Certificado N°' field contains 'GMEL1611-2'. The 'Nombre Cliente' field contains 'Medicos Asociados'. The 'Ciudad' field contains 'Bogotá'. The 'Dirección' field contains 'Cra. 73 #6 Sur-8'. The 'Teléfono' field contains '5658455'. A red button labeled 'Siguiete' is at the bottom right of the form.

Figura 8-27: Entorno grafico para la creación de un certificado, equipo electrobisturí.

Para crear cualquier certificado es necesario obtener los datos del cliente y los patrones requeridos para el equipo seleccionado. Estos datos se adquieren a partir de los buscadores creados en el área de “Buscar Cliente” y “Buscar Patrones”, como se muestra en la **Figura 8-27**. Una vez encontrados el cliente y el patrón, el usuario debe pulsar el botón añadir y el programa automáticamente agregara al formulario los datos específicos para crear el certificado, y de esta manera poder seguir con el siguiente paso del formulario.

Los clientes y patrones deben ser agregados previamente para que el buscador funcione, esto será explicado en la sección de clientes y patrones respectivamente.

Generar Certificado Electrobisturí:

Pasos: 1 2 3 4 5 6

Datos del equipo a calibrar:

Equipo (Obligatorio):

Electrobisturí

Este campo no es editable.

Marca (Obligatorio):

Valley Lab

Sellecciona una marca o agrega una nueva marca sino se encuentra.

Agregar nueva marca

Modelo (Obligatorio):

S4567

Serie (Obligatorio):

Campo vacío

Serie Equipo...

Id. Institución (Obligatorio):

N/A

Temperatura (Obligatorio):

25

Digite solo numeros.

Humedad relativa (Obligatorio):

42

Digite solo numeros.

« Anterior

Siguiente »

Figura 8-28: Paso 2: “Datos del equipo a calibrar”.

En el siguiente paso del formulario de adquisición de datos, el usuario debe ingresar los datos requeridos por el mismo, como son: la marca del equipo a calibrar, modelo, serie, id de la institución de donde se está calibrando (si aplica) y los datos de temperatura y humedad relativa, del sitio donde se está llevando a cabo el proceso de calibración.

Todos estos datos son obligatorios para poder seguir con el siguiente paso, en dado caso que se quiera pasar al siguiente paso y falte un campo obligatorio, el programa indicara el campo que este vacío y no dejara que el usuario pueda acceder al siguiente paso como se muestra en la **Figura 8-28**.


Generar Certificado Electrobisturi:

Pasos:

1
2
3
4
5
6

Datos de lugar, fecha y quien calibra:

Lugar calibración
(Obligatorio):

Lab. Gmetrologia

Fecha de recepción
(Obligatorio):

09/11/2016

Fecha de calibración
(Obligatorio):

09/11/2016

Quien calibra:

Quien calibra...

Cargo:

Metrólogo

Cargo de la persona que calibra (Metrólogo).

Quien verifica:

Quien verifica...

Cargo:

Director de Laboratorio

Cargo de la persona que verifica (Director de Laboratorio).

« Anterior
Siguiente »

Figura 8-29: Paso 3: “Datos de fecha, lugar y quien calibra”.

En el paso 3 del formulario (ver **Figura 8-29**), se encuentran los datos relacionados al lugar, fecha y las personas que están calibrando el equipo. El lugar de calibración variará según se encuentre el equipo a calibrar, las fechas de recepción y calibración también pueden variar en función del día que se realiza el proceso.

Los campos de quien calibra y quien verifica se encuentran bloqueados y vacíos, puesto que se está creando el certificado, estos campos se actualizarán automáticamente en el momento que se revise o verifique el certificado previamente creado, dependiendo de los usuarios que hagan dicha tarea. Estos datos son recolectados con el fin cumplir la sugerencia del numeral 4.3.2.3 de la norma NTC/ISO-IEC-17025 el cual indica que en el certificado se debe agregar la fecha de emisión o calibración y las personas autorizadas para la calibración del certificado y la emisión del mismo.


Generar Certificado Electrohisturi:

Pasos:

1

2

3

4

5

6

Patrones Usados:

Patrón electrocirugía		Patrón temperatura		Patrón seguridad eléctrica	
Patrón electrocirugía:	Analizador de Electrociru	Patrón temperatura:	Temperatura	Patrón Seg. Eléctrica:	Analizador de Seguridad
Marca:	Fluke Biomedical	Marca:	Kex Germany	Marca:	Fluke Biomedical
Modelo:	QA-ES	Modelo:	SH-109	Modelo:	ESA612
Serie:	6628	Serie:	00084	Serie:	2532025
Fecha Cal:	22/09/2016	Fecha Cal:	10/11/2016	Fecha Cal:	09/11/2016
Certificado N°:	S3802-14	Certificado N°:	S5209-14	Certificado N°:	S1611-14
Certificado Url:	AUTOCALAVE MCAC 1501-01.pdf	Certificado Url:	TERMOHIGROMETRO MCTH 1501-01.pdf	Certificado Url:	EQUIPO RX MCRX 1501-01.pdf

« Anterior

Siguiente »

Llegado al paso 4 del formulario (ver **Figura 8-30**), se aprecia la información de los patrones usados para la calibración de un equipo electrobisturí. Estos patrones son agregados por medio del buscador “Buscar Patrones”, mostrado anteriormente en la **Figura 8-27**. En dicho buscador, el programa encuentra el patrón solicitado y al presionar en añadir, se agrega la información al formulario.

Buscar Patrones:

Para éste certificado, es necesario los patrones:

- Analizador de Electrociugia. ✓
- Temperatura. ✓
- Analizador de Seguridad Eléctrica. ✗

Búsquedas Sugeridas:

Analizador de Seguridad Elétrica.

Marca: Fluke Biomedical
Modelo: ESA612
Serie: 2532025
Fecha Cal.: 2016-01-01
Certi. N°: S1611-14

Patrones Usados:

Patrón electrocugia		Patrón temperatura		Patrón seguridad eléctrica	
Patrón electrocugia:	Analizador de Electrociu	Patrón temperatura:	Temperatura	Patrón Seg. Eléctrica:	Patrón seg. eléctrica.
Marca:	Fluke Biomedical	Marca:	Kex Germany	Marca:	Marca...
Modelo:	QA-ES	Modelo:	SH-109	Modelo:	Modelo...
Serie:	6628	Serie:	00084	Serie:	Serie...
Fecha Cal:	22 / 09 / 2016	Fecha Cal:	10 / 11 / 2016	Fecha Cal:	dd / mm / aaaa
Certificado N°:	S3802-14	Certificado N°:	S5209-14	Certificado N°:	Certificado N°...
Certificado Url:	AUTOCALIVE MCAC 1501-01.pdf	Certificado Url:	THERMOHIGROMETRO MCTH 1501-01.pdf	Certificado Url:	

« Anterior

Siguiente »

8.8.2.2. PROCESAMIENTO DE DATOS


Generar Certificado Electrobisturí:

Pasos:

1

2

3

4

5

6

Medición potencias:

Medición Potencia Coagulación:

Potencia Programada	Medición 1 (W)	Medición 2 (W)	Medición 3 (W)	Medición 4 (W)
10	9	8	7	11
30	26	28	30	29
90	89	87	89	90
120	120	121	119	121

Promedios	Error	Desv Estandar	Incert A	Incert B1	Incert B2	Incert Combinada	Uef (Grados de libertad)	K Factor Student	Incert Expandida	Tolerancia
W	W	W	W	W	W	W	W	W	W	W
8.75	-1.25	1.7078	0.8539	0.3989	0.2887	0.9857	5.3271	2	1.9714	5
28.25	-1.75	1.7078	0.8539	0.3989	0.2887	0.9857	5.3271	2	1.9714	9
88.75	-1.25	1.2583	0.6292	0.3989	0.2887	0.7989	7.8013	2	1.5979	27
120.25	0.25	0.9574	0.4787	0.617	0.2887	0.8326	27.4501	2	1.6652	36

Figura 8-32: Adquisición de medidas para la calibración de un electrobisturí.

Para realizar el procesamiento de datos de las mediciones de los simuladores utilizados, es necesario que el usuario de la aplicación ingrese estas mediciones, estos valores variarán dependiendo del equipo a calibrar. En este caso, se hace uso de las mediciones para calibrar un equipo electrobisturí que consta de varias mediciones de potencia, dichas mediciones son procesadas en tiempo real a medida que el usuario las vaya digitando como se aprecia en la **Figura 8-32**. Los resultados de los cálculos generados son realizados a través de una fórmula matemática entregada por la empresa Gmetrologia S.A.S de forma confidencial que básicamente calcula el Promedio, la Media, la Desviación Estandar, las Incertidumbres Combinadas y Expandidas de todas las mediciones realizadas. Finalmente de acuerdo a la tolerancia se decide si el equipo pasa la calibración.

Específicamente en este equipo, los pasos 5 y 6 son para adquirir las mediciones generadas por los simuladores.

8.8.2.3. GENERACION DE CERTIFICADO Y ALMACENAMIENTO

Generar Certificado Electrobisturi:

Pasos: 123456

Medición seguridad electrica:

Prueba Realizada	Especifico	Medición	Unidad
Tensión de red L1 a L2	-	119.7	V
Tensión de red L2 a Ea	-	0.5	V
Tensión de red L1 a Ea	-	119.4	V
Resistencia de protección a tierra	≤ 0.2	0.182	Ω
Resistencia de aislamiento en protección a tierra	≥ 2	99999	MΩ
Fuga de corriente tierra en condición normal	≤ 5000	148.8	μA
Fuga de corriente tierra en neutro abierto	≤ 10000	290	μA
Fuga de corriente tierra normal (red inversa)	≤ 5000	145.8	μA
Fuga de corriente neutro abierto (red inversa)	≤ 10000	290	μA
Fuga de corriente carcasa en condición normal	≤ 100	0.2	μA
Fuga de corriente carcasa en L2 abierto	≤ 500	0.2	μA
Fuga de corriente carcasa en Ea abierto	≤ 500	148.9	μA
Fuga de corriente carcasa normal (red inversa)	≤ 100	0.1	μA
Fuga de corriente carcasa L2 abierto (red inversa)	≤ 500	0.1	μA
Fuga de corriente carcasa Ea abierto (red inversa)	≤ 500	144.2	μA

Ver Pdf

Guardar

« Anterior

Figura 8-33: Paso 6: Generación PDF o almacenamiento.

En el paso 6 o último pasó para la calibración de este equipo en específico, se recolectan las últimas mediciones, y en la parte inferior derecha de la **Figura 8-33**, con color azul y verde, están las opciones para ver el PDF generado o para guardar el certificado y el PDF en la base de datos, para su posterior revisión. La vista del PDF se muestra en una ventana como se aprecia en la **Figura 8-34**.

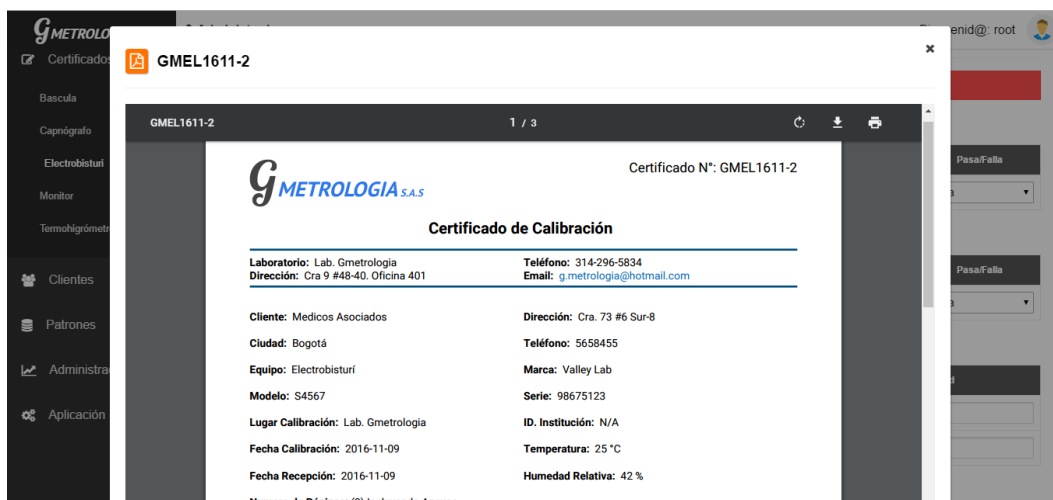


Figura 8-34: Vista certificado generado en PDF para electrobisturí.

Una vez creados y almacenados los certificados, estos quedarán en estado de revisar, así, cuando un usuario con rol metrólogo haga la correspondiente revisión solo tendrá que mirar la misma interfaz pero con los datos ya agregados, de esta manera todos los pasos se repiten y en el paso 6 el metrólogo tendrá dos opciones, la opción de ver el PDF y la opción de guardar como revisado el certificado, en caso de que requiera hacer modificaciones, este las podrá hacer sin ningún problema. Igualmente sucede cuando se verifica un certificado como se muestra en la **Figura 8-35**. Finalmente se cumple con la norma NTC ISO-IEC 17025 numeral 4.3.2.1.

Prueba Realizada	Específico	Medición	Unidad
Tensión de red L1 a L2	-	119,7	V
Tensión de red L2 a Ea	-	0,5	V
Tensión de red L1 a Ea	-	119,4	V
Resistencia de protección a tierra	$\leq 0,2$	0,182	Ω
Resistencia de aislamiento en protección a tierra	≥ 2	99999	M Ω
Fuga de corriente tierra en condición normal	≤ 5000	148,8	μA
Fuga de corriente tierra en neutro abierto	≤ 10000	290	μA
Fuga de corriente tierra normal (red inversa)	≤ 5000	145,8	μA
Fuga de corriente neutro abierto (red inversa)	≤ 10000	290	μA
Fuga de corriente carcasa en condición normal	≤ 100	0,2	μA
Fuga de corriente carcasa en L2 abierto	≤ 500	0,2	μA
Fuga de corriente carcasa en Ea abierto	≤ 500	148,9	μA
Fuga de corriente carcasa normal (red inversa)	≤ 100	0,1	μA
Fuga de corriente carcasa L2 abierto (red inversa)	≤ 500	0,1	μA
Fuga de corriente carcasa Ea abierto (red inversa)	≤ 500	144,2	μA

Ver Pdf Verificado

Figura 8-35: Opciones de verificar certificado.

8.8.3. SECCIÓN CLIENTES

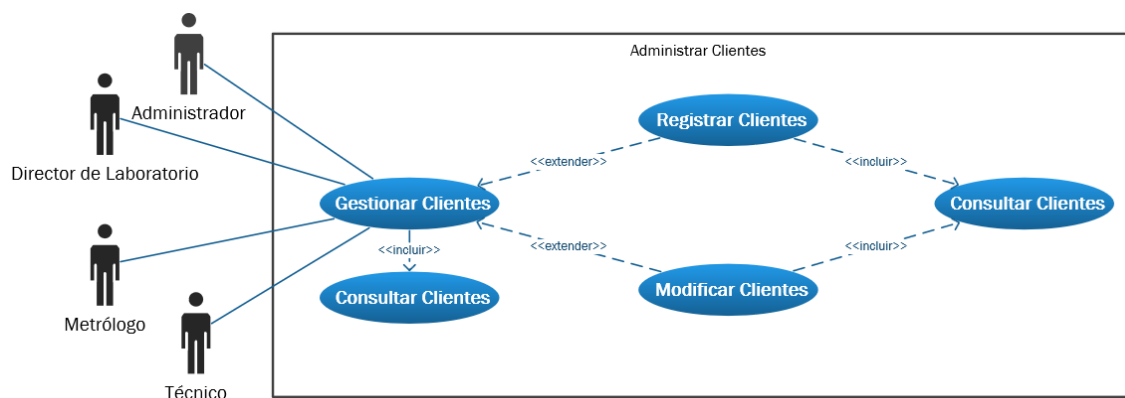


Figura 8-36: Diagrama de caso de uso para administrar clientes.

En esta sección el usuario puede crear clientes y hacer gestión de los mismos, como es buscarlos y modificarlos (ver **Figura 8-36**). Cualquier usuario de la aplicación, independientemente del rol que posea podrá crear clientes, estos clientes pueden ser editados únicamente por el usuario que los creó o el administrador de la aplicación.

Este proceso es fundamental en la realización de un certificado ya que en el mismo es obligatorio indexar los datos del cliente. Además, de esta manera se evitan la inserción de datos duplicados en la base de datos y se podrá hacer un mejor seguimiento de los certificados por cliente.

Todos los clientes son almacenados en la tabla "certificados", y se les agrega el tipo "clientes" con estado "publish".

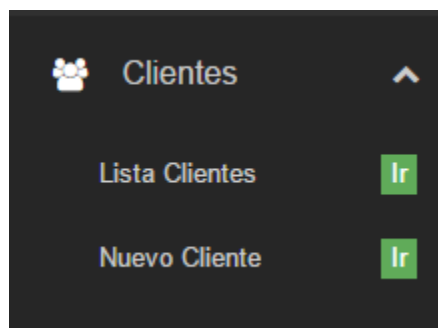


Figura 8-37: Vistas de la sección clientes.

En la aplicación, el área de clientes se divide en dos vistas (ver **Figura 8-37**): "Lista clientes" y "Nuevo Cliente". En estas dos vistas recae toda la funcionalidad de los clientes en la aplicación.

8.8.3.1. VISTA “NUEVO CLIENTE”

The screenshot shows the 'Crear Nuevo Cliente' (Create New Client) form in the G-METROLOGIA S.A.S. application. The interface includes a sidebar with navigation options like 'Inicio', 'Certificados', 'Clientes', 'Patrones', 'Administración', and 'Aplicación'. The main content area is titled 'Crear Nuevo Cliente' and contains several form fields with validation messages:

- Nombre Cliente (Obligatorio):** A text input field with the placeholder 'Nombre Cliente...'. Below it, a message states: 'Los nombres de Clientes pueden tener únicamente caracteres alfanuméricos, espacios, tildes y puntos.'
- NIT o DNI del Cliente:** A text input field with the placeholder 'NIT o DNI...'. Below it, a message states: 'Comprueba bien el NIT o documento de identificación antes de continuar, este debe incluir el dígito de verificación.'
- Teléfono Cliente (obligatorio):** A text input field with the placeholder 'Telefono'. Below it, a message states: 'Telefono del cliente, comprueba que sea correcto.'
- Ciudad o Lugar (Obligatorio):** A dropdown menu with the placeholder '— Elegir Ciudad o Municipio —'. Below it, a message states: 'Selecciona un lugar o agrega una nueva ciudad sino se encuentra.' and a button labeled 'Agregar nuevo lugar'.
- Dirección o Residencia (Obligatorio):** A text input field with the placeholder 'Cra. 9 #51-11...'. Below it, a message states: 'Dirección de Residencia. Caracteres validos: Letras, Numeros, #, *, - y punto.'
- Correo Electrónico:** A text input field with the placeholder 'Email...'. Below it, a message states: 'Comprueba bien la dirección de correo electrónico antes de continuar.'

At the bottom of the form is a button labeled 'Crear Cliente'. The footer of the application shows 'Copyright © 2016 Todos los Derechos Reservados' and 'Creado por: Jose Luis Espinosa Versión: 1.0.0'.

Figura 8-38: Vista “Nuevo Cliente”.

Como se muestra en la **Figura 8-38**, la vista “Nuevo Cliente”, es la interfaz para adquirir los datos de los nuevos clientes que se registrarán en la aplicación, los datos solicitados son: “Nombre del cliente”, “NIT o DNI”, “Teléfono”, “Ciudad”, “Dirección de Residencia” y “Correo electrónico”. El usuario digita los datos solicitados y a continuación oprime el botón “Crear Cliente” guardándolo en la base de datos de la aplicación.

Esta interfaz será la misma en dado caso que se quiera editar un cliente en específico, siempre y cuando el cliente que se vaya a editar sea creado por el mismo usuario que lo piensa modificar.

8.8.3.2. VISTA “LISTA CLIENTES”

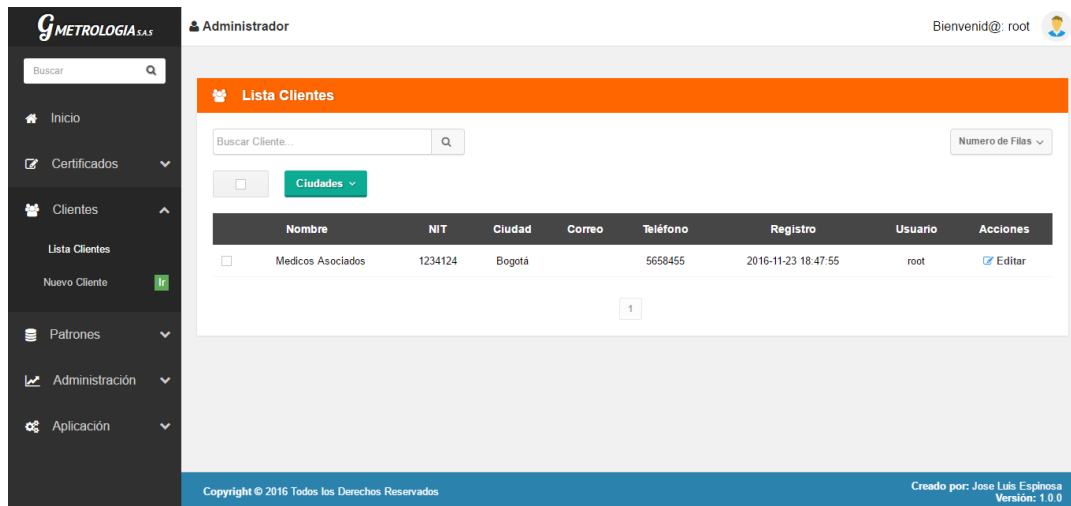


Figura 8-39: Vista “Lista Clientes”.

En la vista “Lista Clientes” (ver **Figura 8-39**) se encuentran todos los clientes creados por el usuario de la sesión iniciada, como también se pueden realizar búsquedas a los clientes requeridos, estos clientes también pueden ser filtrados por ciudades, para poder encontrar fácilmente el usuario necesitado. El administrador podrá ver todos los clientes creados por todos los usuarios de la aplicación.

Además en esta lista se tendrá acceso a la vista de editar usuario, que como se explicaba anteriormente se hace uso de la misma interfaz mostrada en la **Figura 8-38**.

8.8.4. SECCIÓN PATRONES

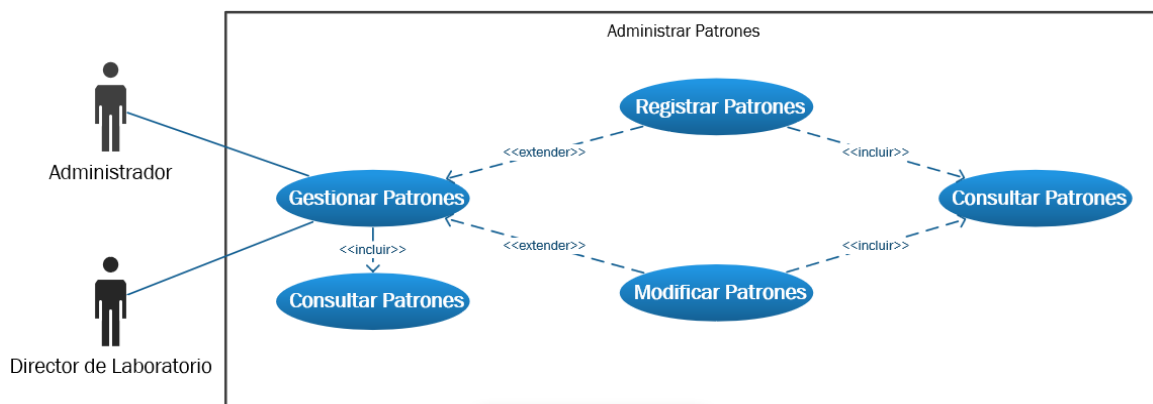


Figura 8-40: Diagramas de caso de uso para administrar patrones.

En esta sección, los usuarios permitidos para administrar patrones son aquellos con rol administrador y director de laboratorio, los patrones son los datos de los certificados que avalan el correcto funcionamiento de los simuladores usados para cada proceso de calibración, estos patrones varían en función del equipo a calibrar y por ende pueden ser múltiples patrones los necesarios para calibrar un equipo en especial. Por ende, los usuarios con rol director de laboratorio y administrador son los indicados para administrar esta parte de la aplicación, pudiendo agregar patrones o editarlos tal como muestra la **Figura 8-40**. Todos los patrones son almacenados en la tabla “certificados”, y se les agrega el tipo “patrones” con estado “publish”.

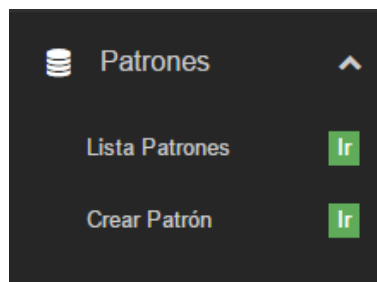


Figura 8-41: Vistas de las secciones patrones.

La administración de patrones se divide en 2 vistas: “Lista Patrones” y “Crear Patrón” como se muestra en la **Figura 8-41**.

8.8.4.1. VISTA “CREAR PATRÓN”

The screenshot shows the 'Crear patrón' form within the 'METROLOGIA S.A.S' application. The left sidebar contains navigation links: Inicio, Certificados, Clientes, Patrones (expanded), Administración, and Aplicación. The 'Patrones' section is active, showing 'Lista Patrones' and 'Crear Patrón'. The main form area has a red header 'Crear patrón:' and a blue header 'Patrón vigente:'. The form includes fields for 'Nombre Patrón (obligatorio)', 'Marca (obligatorio)', 'Modelo (obligatorio)', 'Serie (obligatorio)', 'Fecha calibración (obligatorio)', 'Certificado N° (obligatorio)', and 'Certificado PDF (obligatorio)'. There is a 'Subir Certificado(PDF)' button and a 'Crear Patrón' button at the bottom. The footer shows 'Copyright © 2016 Todos los Derechos Reservados' and 'Creado por: Jose Luis Espinosa Versión: 1.0.0'.

Figura 8-42: Vista crear patrón.

En la **Figura 8-42** se muestra la interfaz para crear un patrón en la aplicación, en esta interfaz el usuario debe elegir seleccionar un patrón de una lista proporcionada por la empresa Gmetrologia S.A.S, en esta lista, se encuentran todos los simuladores que ellos manejan, seguidamente debe elegir una marca de una lista, si la marca no existe, el usuario puede crear una nueva marca, después deberá ingresar los datos del simulador que se va a crear, como son: “Modelo”, “Serie”, “Fecha de Calibración”, “Numero de Certificado o Nombre”, “PDF del Certificado de Calibración”. Todos estos datos son requeridos y obligatorios para la creación de un patrón.

Los patrones son obligatorios para el proceso de crear un certificado, por lo tanto es una parte indispensable del programa.

8.8.4.2. VISTA “LISTA PATRONES”

The screenshot shows the 'Lista Patrones' interface. The sidebar on the left contains the following menu items: Inicio, Certificados, Clientes, Patrones, Lista Patrones, Crear Patrón, Administración, and Aplicación. The main content area has a search bar labeled 'Buscar Patrón...' and a 'Numero de Filas' dropdown. Below the search bar are two buttons: 'Patrones' and 'Marcas'. The table below lists the following data:

Patrón	Marca	Modelo	Serie	Fecha Cal.	Certificado	PDF Certificado	Acciones
☐ Temperatura	Kex Germany	SH-109	00084	2016-11-10	S5209-14	TERMOHIGROMETRO MCTH 1501-01.pdf	Editar
☐ Analizador de Seguridad Eléctrica	Fluke Biomedical	ESA612	2532025	2016-11-09	S1611-14	EQUIPO RX MCRX 1501-01.pdf	Editar
☐ Analizador de Electrocirugía	Fluke Biomedical	QA-ES	6628	2016-09-22	S3802-14	AUTOCLAVE MCAC 1501-01.pdf	Editar

The footer of the interface displays 'Copyright © 2016 Todos los Derechos Reservados' and 'Creado por: Jose Luis Espinosa Versión: 1.0.0'.

Figura 8-43: Vista lista de patrones.

Para realizar la gestión de los patrones creados en la aplicación, es necesario ingresar a la vista de “Lista de Patrones”, como se observa en la **Figura 8-43**, en esta vista se listan todos los patrones creados pudiendo filtrarlos por marcas o por patrones (simuladores), Además se encuentra un buscador el cual sirve para encontrar más fácilmente los patrones pudiendo buscarlos por marca, modelo, serie o número de certificado.

Esta vista es apreciable por el administrador y el director de laboratorio, los cuales pueden modificar un patrón creado por medio de la misma interfaz mostrada en la **Figura 8-42**.

8.8.5. SECCIÓN ADMINISTRACIÓN

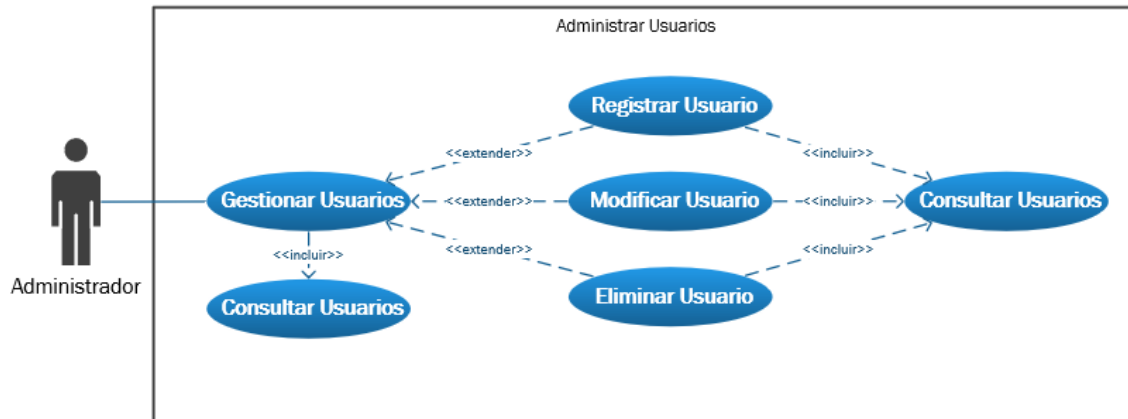


Figura 8-44: Diagrama de casos de usos para la sección de administración.

En la sección de administración el único usuario que puede acceder es el usuario con rol administrador, éste es el responsable de crear todos los usuarios de la aplicación y de hacer gestión a los usuarios creados, por tanto, es el único involucrado en este proceso como se muestra en la **Figura 8-44**. Todos los usuarios creados se almacenan en la tabla “usuarios” de la base de datos con estatus “active”, los usuarios que elimine el administraron posteriormente, se les asigna el estado “delete”.

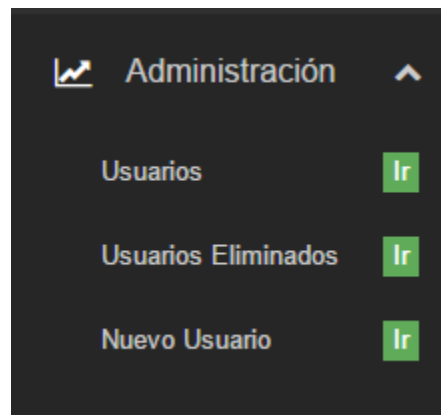


Figura 8-45: Vistas de la sección administración.

La sección “Administración” tiene 3 vistas como se muestra en la **Figura 8-45**, estas vistas son las encargadas de realizar el proceso de administración de los usuarios de la aplicación, y se dividen en 3 vistas: “Usuarios”, “Usuarios Eliminados”, “Nuevo Usuario”.

8.8.5.1. VISTA “NUEVO USUARIO”

METROLOGIA S.A.S. Administrador Bienvenid@: root

Buscar

Inicio

Certificados

Cientes

Patrones

Administración

- Usuarios
- Usuarios Eliminados
- Nuevo Usuario

Aplicación

Crear Nuevo Usuario

Nombre de usuario:

Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Cédula o DNI:

Comprueba bien el numero de Cédula o DNI antes de continuar

Password:

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Un consejo: La contraseña debe tener, al menos, ocho caracteres de longitud. Para hacerla más fuerte, utiliza mayúsculas y minúsculas, números y símbolos como ! " ? \$ % ^ &).

Correo Electrónico:

Comprueba bien la dirección de correo electrónico antes de continuar.

Cargo

Director de Laboratorio: ☐

Metrólogo: ☐

Técnico: ☐

Copyright © 2016 Todos los Derechos Reservados Creado por: Jose Luis Espinosa Versión: 1.0.0

Figura 8-46: Vista “Nuevo Usuario”.

Como es observa en la **Figura 8-46**, la vista “Nuevo Usuario”, tiene una interfaz para adquirir los datos necesarios para registrar un usuario nuevo en la aplicación, los datos requeridos son: “Nombre de usuario”, “Cedula o DNI”, “Password”, “Correo Electrónico” y “Cargo”, los cuales son ingresados por el administrador.

El administrador debe entregar las nuevas credenciales al usuario correspondiente para que este pueda ingresar a la aplicación, ya sea por medio de un correo electrónico o por otro medio.

8.8.5.2. VISTA “USUARIOS”

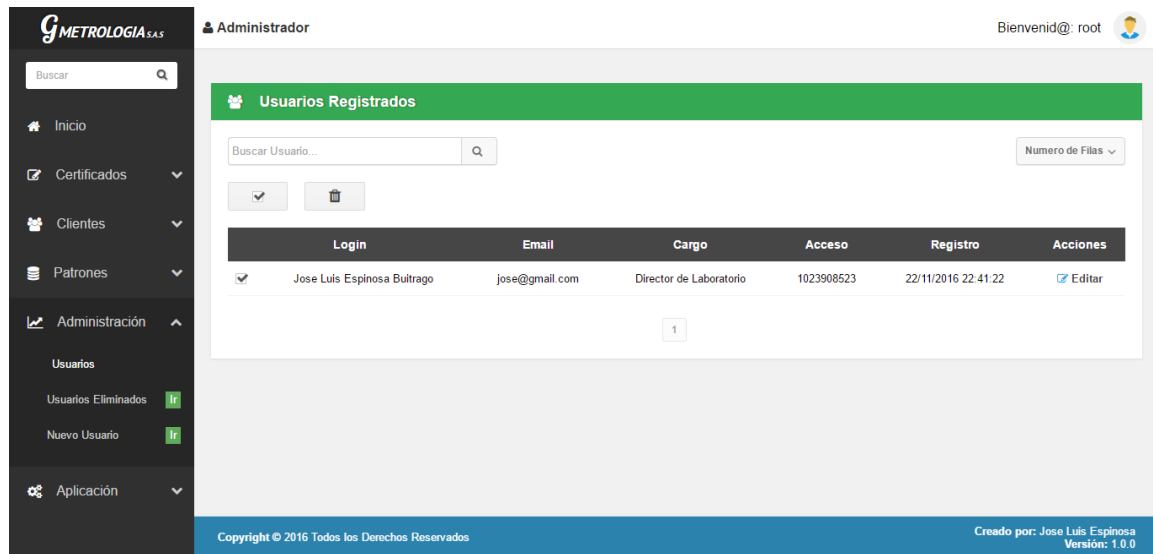


Figura 8-47: Vista “Usuarios”.

La vista “Usuarios” es la encargada de listar todos los usuarios registrados en la aplicación, estos usuarios pueden ser editados o eliminados por el administrador. Para eliminar un usuario se selecciona y se oprime el icono de “basura” ubicado abajo del buscador como se muestra en la **Figura 8-47**, los usuarios eliminados pueden ser rescatados desde la vista “Usuarios Eliminados” la cual se explicara más adelante.

Desde esta vista, el administrador también puede buscar los usuarios por medio de su nombre o cedula y además podrá editar cualquier usuario que el desee. Para poder editar un usuario, se debe pulsar en “editar” y la aplicación cargara la misma interfaz que se muestra en la **Figura 8-46**, pero con las datos guardados de dicho usuario, finalmente el administrador podrá editar cualquier dato y guardar nuevamente la información.

8.8.5.3. VISTA “USUARIOS ELIMINADOS”

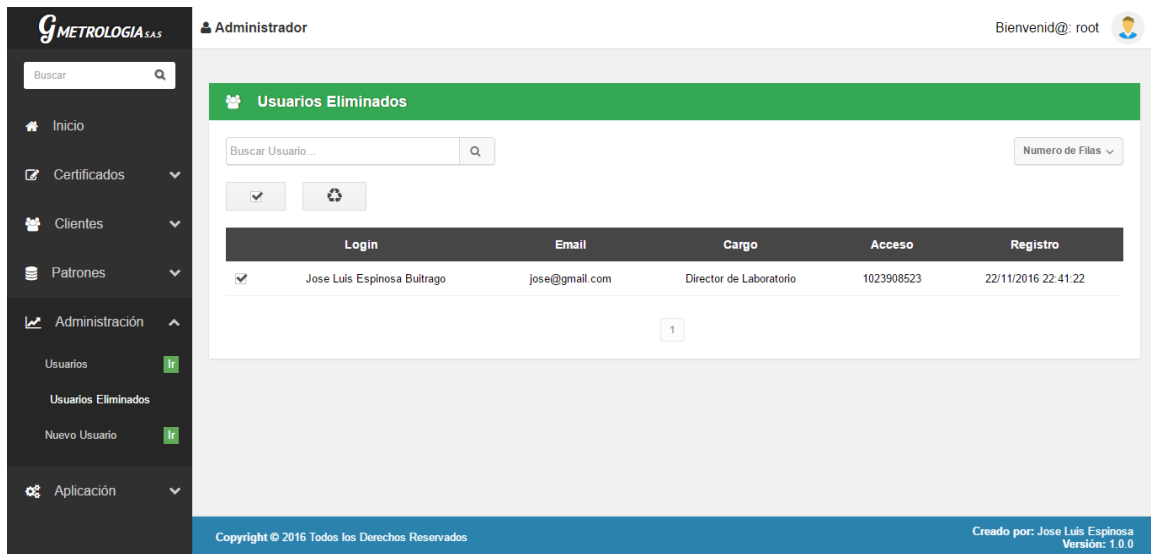


Figura 8-48: Vista “Usuarios Eliminados”.

En la vista de “Usuarios Eliminados” se listan todos los usuarios que ya no tienen acceso a la aplicación o que tienen estado “delete” en la tabla “usuarios”, estos usuarios pueden ser reintegrados nuevamente si el administrador así lo desea, como se muestra en la **Figura 8-48**.

Para reintegrar un usuario eliminado, el administrador debe seleccionar dicho usuario y debe oprimir el icono de reciclaje ubicado abajo del buscador de esta vista. Una vez sea reintegrado, el usuario podrá nuevamente acceder a la aplicación y tener los mismos privilegios que anteriormente haya tenido.

8.8.6. SECCIÓN APLICACIÓN

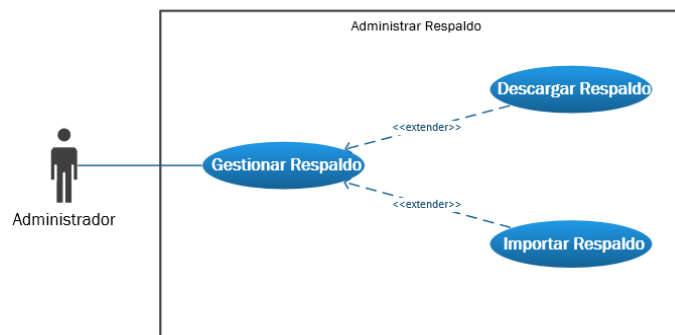


Figura 8-49: Diagrama de casos de uso de la sección aplicación.

En la sección de aplicación se encuentra la vista “Respaldo”, la cual se encarga de gestionar los respaldos de la aplicación, esta vista es únicamente accesible por el administrador como se muestra en la **Figura 8-49**.

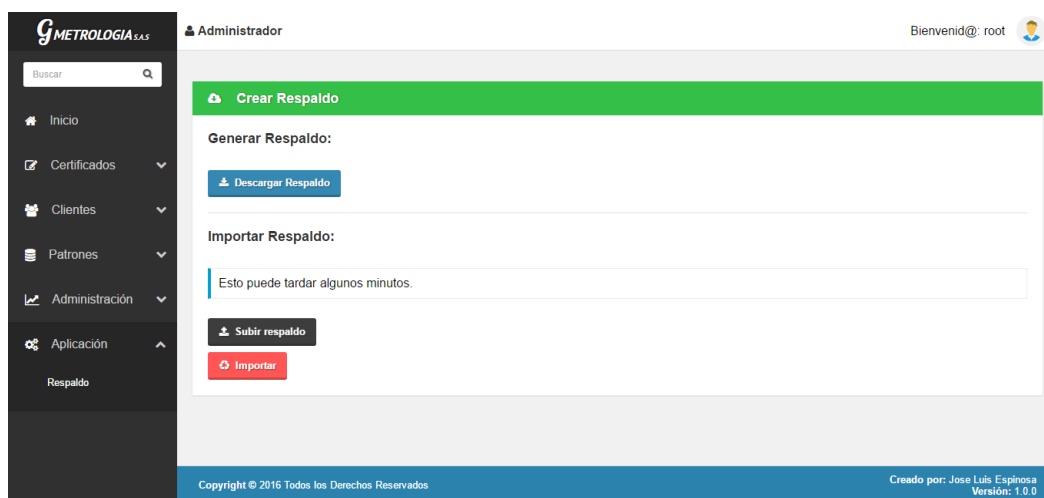


Figura 8-50: Vista “Respaldo”.

El administrador tendrá la responsabilidad de hacer respaldos preventivos, y para poder gestionar dicha tarea, tiene acceso a la interfaz mostrada en la **Figura 8-50**, donde posee 2 opciones. La primera opción es descargar la información de la base de datos que exista en ese momento y la segunda opción, es subir un respaldo previamente descargado el cual restaurara la aplicación dependiendo el contenido que éste contenga.

El proceso de importar respaldo puede ser demorado, esto depende de la cantidad de información que contenga el archivo restaurador.

8.8.7. SECCIÓN EDITAR MI PERFIL

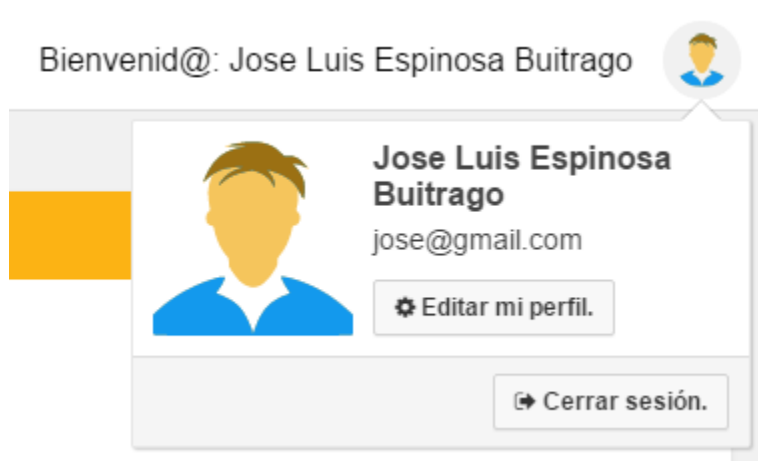


Figura 8-51: Acceso “Editar mi perfil”.

Para poder acceder a la interfaz de “Editar mi perfil”, es necesario dar click sobre el icono de usuario, ubicado en la parte superior derecha de la aplicación como muestra la **Figura 8-51**. En esta ventana se pueden efectuar dos acciones: cerrar sesión o editar mi perfil.

Si se opta por cerrar sesión, la aplicación automáticamente elimina la sesión del usuario y dirige la aplicación a la vista de “Acceso” (ver **Figura 8-16**). Y si el usuario decide editar su perfil, la aplicación muestra la interfaz de la **Figura 8-52**. En esta interfaz el usuario puede cambiar la información de acceso, pudiendo modificar únicamente el correo electrónico y la contraseña.

Para poder efectuar dichos cambios, es necesario que una vez ingresados los nuevos datos, se digite la contraseña actual en el campo “Editar perfil” de la **Figura 8-52**, con el fin de que la aplicación pueda verificar el usuario en la tabla “usuarios” y de esta manera poder efectuar la actualización correspondiente. Es importante mencionar que esta interfaz es accesible para cualquier usuario de la aplicación, independientemente del rol que posea.

Director de Laboratorio

Bienvenid@: Jose Luis Espinosa Buitrago

Buscar

Inicio

Certificados

Cientes

Patrones

Editar mi perfil.

Cambiar

Nombre de usuario:

Jose Luis Espinosa Buitrago

Cédula o DNI:

1023908523

Password:

Contraseña...

Mostrar

Deja este espacio en blanco sino deseas cambiar la contraseña.

Un consejo:

La contraseña debe tener, al menos, ocho caracteres de longitud. Para hacerla más fuerte, utiliza mayúsculas y minúsculas, números y símbolos como ! " ? \$ % ^ &).

Correo Electrónico:

jose@gmail.com

Comprueba bien la dirección de correo electrónico antes de continuar.

Editar Perfil:

Contraseña...

Para efectuar cambios en tu perfil, debes digitar tu contraseña.

Editar mi perfil

Copyright © 2016 Todos los Derechos Reservados

Creado por: Jose Luis Espinosa
Versión: 1.0.0

Figura 8-52: Vista “Editar mi perfil”.

9. CONCLUSIONES, TRABAJO FUTURO Y EVALUACIÓN DEL PROYECTO

9.1. CONCLUSIONES

1. Se creó adecuadamente la interfaz de instalación de la aplicación, junto con la creación del usuario administrador. Facilitando de esta manera la implementación del programa.
2. Se implementó correctamente la interfaz de acceso, otorgando seguridad a la aplicación y asegurando que solo los trabajadores de la empresa Gmetrologia S.A.S tengan acceso a la aplicación, garantizando discreción en el proceso de generación de certificados.
3. La aplicación cumple satisfactoriamente con la norma NTC/ISO-IEC-17025 numeral 4.3, garantizando el control de documentos de la aplicación. Ya que en la generación de cada certificado, este cuenta con un identificador único, el número total de páginas, la fecha de emisión del documento y por último las personas autorizadas para emitir el documento. Además, la aplicación posee un sistema de revisión para los certificados pasando por el técnico, metrólogo y director de laboratorio asegurando que cada certificado deba ser aprobado por el personal pertinente. Cumpliendo de esta manera con los subítems del numeral 4.3 y anexando un certificado final a manera de ejemplo en Anexos 11.2.
4. Al crear la interfaz de adquisición de datos para cada equipo, se integró el algoritmo otorgado por la empresa Gmetrologia S.A.S para el procesamiento de los datos adquiridos, además se realizó un entorno por pasos que garantiza mayor integridad y seguridad en el momento de la creación del certificado junto con la validación de los datos ingresados. Finalmente el certificado debe ser guardado, revisado y verificado satisfaciendo las expectativas de la empresa para poder ser entregado al cliente.
5. El sistema de gestión de la aplicación, garantiza en todo momento la facilidad de encontrar y modificar cualquier certificado existente. Otorgando total privilegio al administrador de la aplicación.
6. Se integró un módulo de clientes y patrones para facilitar la gestión de los mismos, también para facilitar la integración de los clientes y patrones en el momento de la realización de cualquier certificado. Esto garantiza que no habrá tanta redundancia de la información en la base de datos, haciéndola menos pesada y agilizando consultas.
7. Se añadió un módulo de administración de usuarios, para que el administrador tenga control de todos los usuarios creados de la plataforma. También es importante mencionar que en la generación de cada certificado se relaciona con

el usuario que lo está creando. De esta manera facilita la búsqueda de los certificados que ha creado cada usuario.

8. Se agregó un módulo de respaldo, para realizar copias de seguridad preventivas con el fin de no tener toda la información centralizada en dado caso que suceda un error en el servidor. De esta manera, si se corrompe la información, el administrador podrá rescatarla siempre y cuando haya hecho un respaldo de la aplicación. Lo cual mejora el proceso de mantenimientos de la aplicación.
9. El desarrollo de la aplicación mejoró y cumplió con las necesidades de la empresa Gmetrologia S.A.S, frente al modelo que tenían anteriormente en el proceso de generación de certificados. La aplicación genera más seguridad, más efectividad en el proceso de creación de certificados, más control, mejor privacidad y el poder realizar gestión tanto de clientes como de certificados.

9.2. TRABAJO FUTURO

Eventualmente todo trabajo puede y debe ser mejorado conforme pasa el tiempo, de esta manera algunas de las sugerencias para mejorar este proyecto son las siguientes:

1. **Control de errores:** es posible ampliar el control de errores de la aplicación, el programa muestra o controla los principales errores que pueden surgir, pero evidentemente es mejorable el control de errores de la aplicación.
2. **Integrar un sistema de estadísticas:** se sugiere integrar un módulo de estadísticas, para analizar de una mejor manera la productividad de la aplicación, también podrá mejorar las estrategias para cumplir con las metas de la empresa Gmetrologia S.A.S.
3. **Implementación de un módulo de gestión de facturas:** en el contexto de la aplicación, se sugiere la integración de un módulo de facturación, ya que el software obtiene y almacena la información del cliente. Por tanto, sería un buen complemento al programa y a la empresa, ya que facilita un poco la parte administrativa y financiera de la empresa Gmetrologia S.A.S.
4. **Generación de firmas digitales y correo electrónico:** sería un excelente complemento a la aplicación, el integrar un sistema de firmas digitales que garanticen la veracidad del documento, ya que actualmente se tiene que imprimir el documento para entregarlo al cliente. Si se integrara este sistema, los certificados se podrían enviar de manera digital, ya sea por medio de correos electrónicos, USB, CD, etc. Así se contribuiría con el medio ambiente.

9.3. EVALUACIÓN DEL PROYECTO

9.3.1. FUNCIONALIDAD

La aplicación cumple con todos los objetivos propuestos en este proyecto y cuenta con una certificación otorgada por la empresa Gmetrologia S.A.S (ver Anexos 10.1), garantizando el correcto funcionamiento del programa de acuerdo a las necesidades de la empresa.

9.3.2. FIABILIDAD

En el tiempo de uso de la aplicación (2 Meses), no se han presentado errores que comprometan la información de la base de datos ni de los certificados, se han presentado errores de cuestión estética y funcionamiento del lado del cliente. Se agregó un módulo de respaldo el cual ayuda a los mantenimientos de la aplicación y así garantizar la integridad de la información.

9.3.3. USABILIDAD

La aplicación cuenta con un entorno bastante amigable con el usuario y para realizar los trabajos correspondientes, estos son muy intuitivos y entendibles.

9.3.4. EFICIENCIA

Teniendo en cuenta el proceso anterior que empleaba la empresa Gmetrologia S.A.S para realizar la tarea de generación de certificados y comparándolo con la aplicación. Los tiempos de creación, seguridad y gestión, se obtiene una mayor eficiencia en todo el proceso de realización de certificados con la aplicación.

9.3.5. MANTENIBILIDAD

La aplicación se desarrolló con una arquitectura de software MVC, la cual ya fue mencionada en este documento. De esta manera se garantiza una mayor escalabilidad en el código, mejor integración de otras funcionalidades y la facilidad de encontrar errores más fácilmente.

9.3.6. PORTABILIDAD

Al ser una aplicación web esta no requiere instalación y la ventaja es que siempre se va a ver la versión más actualizada. Por otra parte la adaptación a móviles y tabletas debe ser mejorada, ya que actualmente la aplicación se adapta pero tiene algunas fallas estéticas.

10. REFERENCIAS

- [1] Rodríguez, E., M. C. Sánchez, and A. La Miguel, «Praxis en la Gestión de Equipos Médicos,» *Departamento de Bioingeniería, ISPJAE, Cuba*, 2003.
- [2] J. C. Jerez, «La Cátedra de Humanidades una Respuesta a los Servicios de Salud Deshumanizados,» 2014.
- [3] López-Isaza, Giovanni A., and Luis E. Llamosa-Rincón, «Diagnóstico de la calibración del equipo biomédico en entidades de salud del departamento de Risaralda,» *Revista de Salud Pública*, vol. 10, nº 3, pp. 462-469, 2008.
- [4] Celsius Ltda., «Preguntas Frecuentes: ¿Qué es la metrología?,» [En línea]. Available: http://www.celsiusmetrologia.com/index.php?option=com_content&view=article&id=36:ique-es-la-metrologia&catid=13:noticias&Itemid=40. [Último acceso: 06 05 2015].
- [5] Castro, Jorge de, «Certificados de calibración; que son, para que se utilizan y diferentes tipos.,» 19 Noviembre 2012. [En línea]. Available: <http://www.measurecontrol.com/certificados-de-calibracion-que-son-para-que-se-utilizan-y-diferentes-tipos/>. [Último acceso: 14 05 2015].
- [6] Pullés, Marlen Robert, Rogelio Mayarí Navarro, and Victoria Martínez Morales, «Criterios para la acreditación de laboratorios que realizan ensayos microbiológicos según NC-ISO/IEC 17025: 00.,» *Revista CENIC Ciencias Biológicas*, vol. 37, nº 1, 2006.
- [7] A. a. R. J. T. Leff, «Web-application development using the model/view/controller design pattern,» 2001.
- [8] Y. D. a. R. Y. F. González, «Patrón Modelo-Vista-Controlador,» *Revista Telem@tica*, 2012.
- [9] R. T. a. K. G. Fielding, «The Apache HTTP server project,» 1997.
- [10] wikipedia.org, «PHP - Wikipedia, la enciclopedia libre,» [En línea]. Available: <https://es.wikipedia.org/wiki/PHP>. [Último acceso: 11 11 2016].
- [11] A. d. C. San Félix, *Webs dinámicos con PHP*, vol. 2, Prensa Técnica SL, 2000, pp. 57-60.
- [12] php.net, «PHP: ¿Qué puede hacer PHP? - Manual,» [En línea]. Available: <http://php.net/manual/es/intro-whatcando.php>. [Último acceso: 9 11 2016].
- [13] techtarget.com, «MySQL: Definición,» 01 2015. [En línea]. Available: <http://searchdatacenter.techtarget.com/es/definicion/MySQL>. [Último acceso: 14 05 2015].
- [14] M. a. U. M. M. a. A. M. S. a. H. S. Ahmed, *MySQL performance analysis on a limited resource server: Fedora vs. Ubuntu Linux*, 2010, p. 99.
- [15] wikipedia.org, «JavaScript - Wikipedia, la enciclopedia libre,» [En línea]. Available: <https://es.wikipedia.org/wiki/JavaScript>. [Último acceso: 11 11 2016].
- [16] S. a. S. S. a. M. A. a. P. K. Alimadadi, *Understanding JavaScript event-based interactions*, ACM, 2014, pp. 367--377.

- [17] Jesús Matías Almaraz Hernández - Pablo Campos Cantero - Tamara Castelo Delgado, Desarrollo de una aplicación Web para la gestión de Entornos Virtuales, 2011, p. 23.
- [18] S. a. A. B. a. H. A. E. a. Z. Y. McIntosh, «Using indexed sequence diagrams to recover the behaviour of AJAX applications,» 2011.
- [19] C.-z. a. H. H.-k. KE, Principle and application of Ajax technology [J], 2007, p. 11.
- [20] developer.mozilla.org, «CSS | MDN,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/CSS>. [Último acceso: 12 11 2016].
- [21] B. S. Gardner, Responsive web design: Enriching the user experience, vol. 11, 2011, pp. 13-19.
- [22] H. W. a. S. J. Lie, Multipurpose Web publishing using HTML, XML, and CSS, vol. 42, ACM, 1999, pp. 95-101.
- [23] librosweb.es, «Capítulo 1. Introducción (Introducción a CSS),» [En línea]. Available: http://librosweb.es/libro/css/capitulo_1.html. [Último acceso: 12 11 2016].
- [24] Nicolás Aristizábal Mejía – Miguel Eduardo Torres Moreno, «Técnicas de Levantamiento de Requerimientos con Innovación,» 2009.
- [25] J. D. R. F. Jimmy Fernando Jaramillo Benítez, Desarrollo de un Sistema Web para la gestión de presupuestos de Proyectos de investigación-desarrollo-innovación de la Universidad Nacional de Loja utilizando el entorno de programación PHP, Loja, p. 56.
- [26] wikipedia.org, «Aplicación web - Wikipedia, la enciclopedia libre,» [En línea]. Available: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web. [Último acceso: 11 11 2016].
- [27] php.net, «PHP: ¿Qué es PHP? - Manual,» [En línea]. Available: <http://php.net/manual/es/intro-what-is.php>. [Último acceso: 11 11 2016].
- [28] Jesús Matías Almaraz Hernández, Pablo Campos Cantero Tamara, Castelo Delgado, Desarrollo de una aplicación Web para la gestión de Entornos Virtuales, 2011, pp. 30-31.
- [29] librosweb.es, «Capítulo 1. Introducción a AJAX (Introducción a AJAX),» 12 11 2016. [En línea]. Available: https://librosweb.es/libro/ajax/capitulo_1.html.
- [30] librosweb.es, «4.1. La arquitectura MVC (El tutorial Jobeet),» [En línea]. Available: http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html. [Último acceso: 9 11 2016].

11. ANEXOS

11.1. CERTIFICACIÓN DE LA EMPRESA GMETROLOGIA



NIT. 900.521.661-9
CEL. 321 3713099 – Tel 7597463
CR. 9 No. 48-40 oficina 401
BOGOTÁ-COLOMBIA
g.metrologia@hotmail.com

Bogotá, 30 de Noviembre de 2016

Señores:
UNIVERSIDAD SANTO TOMAS
FACULTAD DE INGENIERÍA ELECTRÓNICA
Ciudad.

Cordial saludo,

Certificamos que la aplicación “**Aplicación web para generar certificados de calibración para equipos médicos.**”, realizada por el estudiante Jose Luis Espinosa Buitrago identificado con la cedula de ciudadanía 1.023.908.523 de la ciudad de Bogotá. Cumple satisfactoriamente con nuestro proceso de calibración en el área de certificación de equipos médicos.

La aplicación ha sido probada durante 2 meses, teniendo una buena acogida por nuestro equipo de trabajo y agilizando la producción de certificados de calibración.

Agradecemos la atención prestada.

Atentamente,



Mauricio Alejandro Gómez Figueroa
CC. 80.802.407 de Bogotá
Gerente General
CEL. 321 3713099

11.2. EJEMPLO DE CERTIFICADO GENERADO



Certificado N°: GMEL1611-2

Certificado de Calibración

Laboratorio: Lab. Gmetrologia	Teléfono: 314-296-5834
Dirección: Cra 9 #48-40. Oficina 401	Email: g.metrologia@hotmail.com

Cliente: Medicos Asociados	Dirección: Cra. 73 #6 Sur-8
Ciudad: Bogotá	Teléfono: 5658455
Equipo: Electrobisturí	Marca: Valley Lab
Modelo: S4567	Serie: 98675123
Lugar Calibración: Lab. Gmetrologia	ID. Institución: N/A
Fecha Calibración: 2016-11-09	Temperatura: 25 °C
Fecha Recepción: 2016-11-09	Humedad Relativa: 42 %

Numero de Páginas: (3) Incluyendo Anexos

Este certificado no podrá ser reproducido total o parcialmente excepto cuando se haya obtenido permiso por escrito de GMETROLOGIA S.A.S

Los resultados contenidos en el presente certificado se refieren al momento y condiciones en que se realizaron las mediciones. El laboratorio que lo emite no se responsabiliza de los perjuicios que puedan derivarse del uso inadecuado de los instrumentos calibrados.

El usuario es responsable de la recalibración de sus instrumentos a intervalos apropiados.

FIRMAS AUTORIZADAS:

Metrólogo.

Director de Laboratorio.

Trabajo Realizado:
Calibración del Equipo.

Método Utilizado:
Comparación con el patrón de referencia, procedimiento de calibración GMETROLOGIA S.A.S La incertidumbre de calibración ha sido estimada con un nivel de confianza del 95% utilizando un factor de cobertura $k=2$.

Información de Trazabilidad:
Nuestros instrumentos son trazables a estándares internacionales.

Patrón	Marca	Modelo	Serie	Fecha Cal	Certificado
Analizador de Electrocirugía	Fluke Biomedical	QA-ES	6628	2016-09-22	S3802-14
Temperatura	Kex Germany	SH-109	00084	2016-11-10	S5209-14
Analizador de Seguridad Eléctrica	Fluke Biomedical	ESA612	2532025	2016-11-09	S1611-14

Resultados de Calibración:

POTENCIA COAGULACIÓN						
SELECCIÓN	MEDICIÓN	ERROR	INCERTID.	UNID	TOLERANCIA	PASA/FALLA
10	8.75	1.25	1.9714	W	5	Pasa
30	28.25	1.75	1.9714	W	9	Pasa
90	88.75	1.25	1.5979	W	27	Pasa
120	120.25	0.25	1.6652	W	36	Pasa

POTENCIA MONOPOLAR						
SELECCIÓN	MEDICIÓN	ERROR	INCERTID.	UNID	TOLERANCIA	PASA/FALLA
30	29	1	2.3473	W	9	Pasa
90	89.75	0.25	2.2568	W	27	Pasa
120	119.25	0.75	2.7857	W	36	Pasa
300	301.5	1.5	2.5515	W	90	Pasa

POTENCIA BLEND1						
SELECCIÓN	MEDICIÓN	ERROR	INCERTID.	UNID	TOLERANCIA	PASA/FALLA
30	28.75	1.25	1.8044	W	9	Pasa
90	90.75	0.75	1.3865	W	27	Pasa
120	120.75	0.75	0.7676	W	36	Pasa
250	250	0	1.0029	W	75	Pasa

Voltaje de Línea				
SELECCIÓN	ESPECIFIC.	MEDICIÓN	UNID.	PASA/FALLA
VOLTAJE	≥ 117	120.3	V	Pasa

Resistencia de Protección a Tierra				
SELECCIÓN	ESPECIFIC.	MEDICIÓN	UNID.	PASA/FALLA
RESISTENCIA	≤ 0.2	0.126	Ω	Pasa

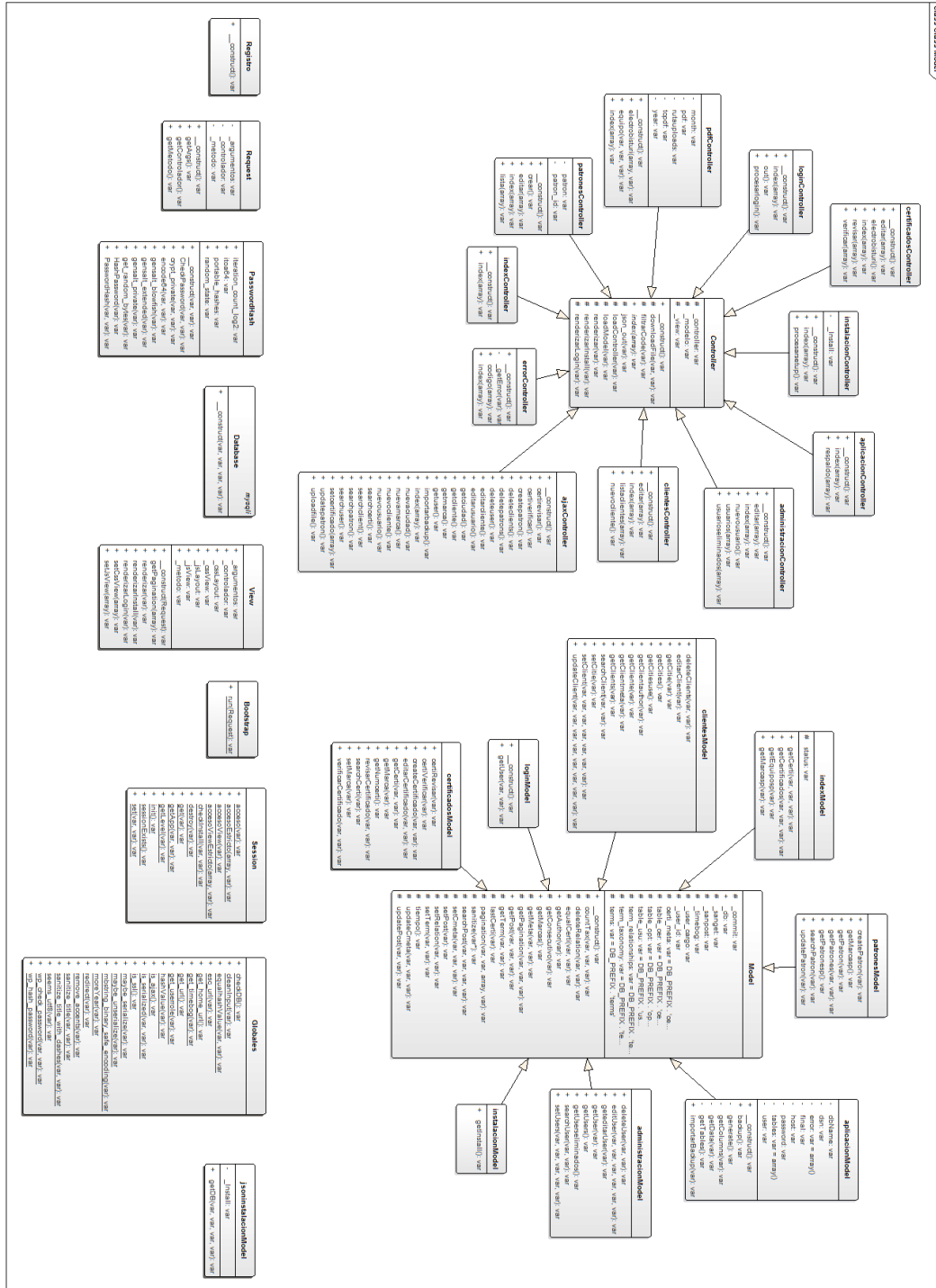
Condiciones de primer defecto (sfc) con neutral abierto		
Abrir neutro-adelante	300	μA
Abrir neutro-reverse	302	μA
Condición normal (NC) con neutro cerrados		
Neutro cerrado-adelante	150	μA
Neutro cerrado-reverse	151	μA

PRUEBA DE SEGURIDAD ELECTRICA IEC 60601-1:

PRUEBA REALIZADA	ESPECIFIC.	MEDICIÓN	UNID.
Tensión de red L1 a L2	-	119.7	V
Tensión de red L2 a Ea	-	0.5	V
Tensión de red L1 a Ea	-	119.4	V
Resistencia de Protección a Tierra	≤ 0.2	0.182	Ω
Resistencia de Aislamiento en Protección a tierra	≥ 2	99999	M Ω
Fuga de Corriente Tierra en Condición Normal	≤ 5000	148.8	μA
Fuga de Corriente Tierra en Neutro Abierto	≤ 10000	290	μA
Fuga de Corriente Tierra Normal (Red Inversa)	≤ 5000	145.8	μA
Fuga de Corriente Neutro Abierto (Red Inversa)	≤ 10000	290	μA
Fuga de Corriente Carcasa en Condición Normal	≤ 100	0.2	μA
Fuga de Corriente Carcasa en L2 Abierto	≤ 500	0.2	μA
Fuga de Corriente Carcasa en Ea Abierta	≤ 500	148.9	μA
Fuga de Corriente Carcasa Normal (Red Inversa)	≤ 100	0.1	μA
Fuga de Corriente Carcasa L2 Abierto (Red Inversa)	≤ 500	0.1	μA
Fuga de Corriente Carcasa Ea Abierta (Red Inversa)	≤ 500	144.2	μA

Lineas de alimentación: L1 = Fase; L2 = Neutro; Ea = Tierra

11.3. DIAGRAMA DE CLASES



11.4. CLASES PRINCIPALES DE LA APLICACIÓN

11.4.1. “REQUEST.PHP”

```
<?php

class Request
{
    private $_controlador;
    private $_metodo;
    private $_argumentos;

    public function __construct() {

        if(isset($_GET['url'])){

            $url = filter_input(INPUT_GET, 'url', FILTER_SANITIZE_URL);
            $url = explode('/', $url);
            $url = array_filter($url);

            $this->_controlador = strtolower(array_shift($url));
            $this->_metodo = strtolower(array_shift($url));
            $this->_argumentos = array_map('strtolower', $url);
        }

        if(!$this->_controlador){
            $this->_controlador = DEFAULT_CONTROLLER;
        }

        if(!$this->_metodo){
            $this->_metodo = 'index';
        }

        if(!isset($this->_argumentos)){
            $this->_argumentos = array();
        }
    }

    public function getControlador() {

        return $this->_controlador;
    }

    public function getMetodo() {

        return $this->_metodo;
    }

    public function getArgs() {

        return $this->_argumentos;
    }
}

?>
```

11.4.2. “BOOTSRAP.PHP”

```
<?php

class Bootstrap{

    public static function run(Request $peticion) {

        $controller = $peticion->getControlador();
        $controlador = $controller . 'Controller';
        $rutaControlador = ABSPATH . 'controllers' . DS . $controlador . '.php';
        $metodo = $peticion->getMetodo();
        $args = $peticion->getArgs();

        if(is_readable($rutaControlador)){

            if(Session::checkInstall($controller, $metodo)){

                Session::init();

                if(Session::getApp($controller, $metodo)){

                    require_once $rutaControlador;

                    if(method_exists($controlador, $metodo)){

                        $controlador = new $controlador();

                        if(isset($args)){

                            call_user_func_array(array($controlador, $metodo),
array($args));
                        }else{

                            call_user_func(array($controlador, $metodo));

                        }

                    }else{

                        throw new Exception('Metodo No Encontrado');

                    }

                }else{

                    throw new Exception('Problema de getApp');

                }

            }else{

                throw new Exception('Problema de instalación');

            }

        }else {

            throw new Exception('Controlador No Encontrado');

        }

    }

}

?>
```

11.4.3. “CONTROLLER.PHP”

```
<?php

abstract class Controller
{

    protected $_view;
    protected $_modelo;
    protected $_controller;

    public function __construct(){

        $this->_view = new View(new Request());

    }

    abstract public function index(array $args);

    protected function loadModel($modelo){

        $modelo = $modelo . 'Model';
        $rutaModelo = ABSPATH . 'models' . DS . $modelo . '.php';

        if(is_readable($rutaModelo)){

            require_once($rutaModelo);
            $modelo = new $modelo;
            return $modelo;

        } else {

            throw new Exception('Modelo no encontrado.');
```

```

        header('Content-type: application/octetstream');
        header('Pragma: no-cache');
        header('Expires: 0');
        echo ($compress ? gzencode($archivo) : $archivo);
        exit;
    }

    protected function json_out($json){

        if(!empty($this->_modelo->_db->thread_id)){
            $this->_modelo->_db->close();
        }

        header('Content-type: application/json; charset=utf-8', true);
        die(json_encode($json, JSON_FORCE_OBJECT));

    }

    protected function renderizar($vista){

        if(!empty($this->_modelo->_db->thread_id)){
            $this->_modelo->_db->close();
        }

        $this-> view->renderizar($vista);
    }

    protected function renderizarLogin($vista){

        if(!empty($this->_modelo->_db->thread_id)){
            $this-> _modelo-> _db->close();
        }

        $this->_view->renderizarLogin($vista);
    }

    protected function renderizarInstall($vista){

        if(!empty($this->_modelo->_db->thread_id)){
            $this->_modelo->_db->close();
        }

        $this->_view->renderizarInstall($vista);
    }

    protected function filtrarCode($codigo){

        if (preg_match('/^[0-9]+$/', $codigo)){

            return $codigo;

        }else{

            return false;

        }

    }

}

?>

```

11.4.4. “MODEL.PHP”

```
<?php
```

```
class Model
{
    public $_db;
    protected $_sanpost;
    protected $_sanget;
    protected $_user_cargo;
    protected $_user_id;
    protected $_commit;
    protected $_timebog;

    protected $table_usu = DB_PREFIX . 'usuarios';
    protected $usermeta = DB_PREFIX . 'usermeta';
    protected $table_cer = DB_PREFIX . 'certificados';
    protected $table_opt = DB_PREFIX . 'opciones';
    protected $certi_meta = DB_PREFIX . 'certificadosmeta';
    protected $terms = DB_PREFIX . 'terms';
    protected $term_taxonomy = DB_PREFIX . 'term_taxonomy';
    protected $term_relationships = DB_PREFIX . 'term_relationships';

    public function __construct() {

        $this->_db = new Database();
        $this->_user_cargo = Session::get('user_cargo');
        $this->_user_id = Session::get('id_usu');
        $this->_timebog = Globales::get_timebog();

        if(isset($_POST) and !empty($_POST)) {

            $this->_sanpost = &$_POST;

            $this->sanitize($this->_sanpost);

        }

        if(isset($_GET) and !empty($_GET)) {

            $this->_sanget = &$_GET;

            $this->sanitize($this->_sanget);

        }

    }

    protected function sanitize(&$item){

        if (is_array($item)) {

            foreach ($item as $key => $value) {

                if(is_array($value)){

                    self::sanitize($value);

                }else{


```

```

        $item[$key] = $this->db->real escape string(Globales::cleanInput($value));
    }

}

}else{

    $item = $this->db->real escape string(Globales::cleanInput($item));

}

}

protected function getPagination($page, $totales, $limite){

    $paginacion = array();
    $totalpagi = ceil($totales / $limite);
    $contador = 0; $maxpaginas = 7; $minpaginas = $maxpaginas - 1;
    $paginacion['actual'] = $page;
    $paginacion['limite'] = $limite;

    if($page > 1){
        $paginacion['anterior'] = $page - 1;
    }

    if($page < $totalpagi){
        $paginacion['siguiente'] = $page + 1;
    }

    if($totalpagi <= $maxpaginas){

        $paginacion['estado1'] = true;

        for($contador=1; $contador <= $totalpagi; $contador++){
            $paginacion['pages'][] = $contador;
        }

    }elseif($totalpagi > $maxpaginas){

        if($page <= $minpaginas){

            $paginacion['estado2'] = true;
            $paginacion['ultimo'] = $totalpagi;

            for($contador = 1; $contador <= $maxpaginas; $contador++){

                $paginacion['pages'][] = $contador;
            }

        }elseif($totalpagi - ($minpaginas - 1) > $page && $page > ($minpaginas - 1)){

            $paginacion['estado3'] = true;
            $paginacion['primero'] = 1;

            for($contador = $page - 3; $contador <= $page + 3; $contador++){

                $paginacion['pages'][] = $contador;
            }

            $paginacion['ultimo'] = $totalpagi;

        }else{

            $paginacion['estado4'] = true;

```



```

        $paginacion['primero'] = 1;

        for($contador = $totalpagi - $minpaginas; $contador <= $totalpagi;
$contador++){

            $paginacion['pages'][] = $contador;

        }

    }

}

return $paginacion;

}

protected function pagination($table, $query, array $args, $querycount){

    $page = NULL;
    $limite = NULL;

    if(isset($_GET['page']) and !empty($_GET['page'])){

        $page = abs(intval($_GET['page']));

    }

    if(isset($_GET['limite']) and !empty($_GET['limite'])){

        $limite = abs(intval($_GET['limite']));

    }

    $returnG = array();

    $totales = 0;
    $returnG['pagination'] = false;
    $returnG['registros'] = false;

    if($count = $this->_db->query("$querycount")){

        $totales = $count->fetch row()[0];
        $count->close();

    }

    if($limite == 25){
        $limite = 25;
    }
    else if($limite == 50){
        $limite = 50;
    }
    else if($limite == 100){
        $limite = 100;
    }
    else if($limite == 200){
        $limite = 200;
    }
    else{
        $limite = 25;
    }

    if($page){

```

```

        $inicio = ($page - 1) * $limite;

        if($inicio >= $totales){
            $inicio = 0;
        }

    }else{
        $inicio = 0;
        $page = 1;
    }

    if($totales > 0){

        $filasA = array();
        $params = array();
        $len = count($args);

        if($len){

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

                if($i==0){
                    $params[] = $args[$i].'.ii';
                }
                if($i>0 and $i <= $len){
                    $params[] = &$args[$i];
                }
                if($i == $len){

                    $params[] = &$inicio;
                    $params[] = &$limite;

                }
            }

        }else{

            $params[] = 'ii';
            $params[] = &$inicio;
            $params[] = &$limite;

        }

        $filas = $this->_db->prepare("

            SELECT *
            FROM {$table} AS t1
            INNER JOIN (

                {$query}

            LIMIT ?, ?

            ) AS tt3

            ON (t1.ID = tt3.ID)

            ORDER BY t1.ID DESC

        ");

        call_user_func_array(array($filas, "bind_param"), $params);

        if($filas->execute()){

```

```

        $resultado = $filas->get_result();

        while ($fila = $resultado->fetch_object()) {

            $filasA[] = $fila;

        }

    }

    $filas->close();
    $returnG['registros'] = $filasA;

    $returnG['pagination'] = $this->getPagination($page, $totales, $limite);

}

return $returnG;

}

protected function tiempo(){

    return date("Y-m-d H:i:s");

}

protected function setCmeta($post_id, $meta_key, $meta_value){

    $codigo = false;

    $setCmeta = $this->_db->prepare("

        INSERT INTO {$this->certi_meta} (post_id, meta_key, meta_value)
        VALUES (?, ?, ?)

    ");

    $setCmeta->bind_param('sss', $post_id, $meta_key, $meta_value);

    $setCmeta->execute();

    if($last_Id = $setCmeta->insert_id){

        $setCmeta->close();
        $codigo = true;

    }

    return $codigo;

}

protected function updateCmeta($post_id, $meta_key, $meta_value){

    $codigo = false;

    $updateCmeta = $this->_db->prepare("

        UPDATE {$this->certi_meta}
        SET meta_value = ?
        WHERE post_id = ? AND meta_key = ?

    ");

```

```

        ");

        $updateCmeta->bind_param('sis', $meta_value, $post_id, $meta_key);

        $updateCmeta->execute();

        if($updateCmeta->affected_rows){

            $updateCmeta->close();
            $codigo = true;

        }

        return $codigo;

    }

    protected function getTerm($slug, $taxonomy = false){

        $codigo = false;
        $tax = '';

        if(!empty($taxonomy)){
            $tax = "AND t1.taxonomy = '$taxonomy'";
        }

        if(is_array($slug) and !empty($slug)){

            $len = count($slug);
            $i = 0;
            $slugP = '';
            foreach ($slug as $value) {

                if(++$i == $len) {
                    $slugP .= "".$value."";
                }else{
                    $slugP .= "".$value."', ";
                }
            }
            $slug = $slugP;

        }else{

            $slug = "".$slug."";

        }

        $query = "
        SELECT DISTINCT t1.term_taxonomy_id, gterms.term_id, gterms.slug
        FROM $this->term_taxonomy AS t1
        LEFT JOIN {$this->terms} gterms ON (t1.term_id = gterms.term_id)
        WHERE gterms.slug IN ($slug)
        {$tax}
        ";

        if($term = $this->_db->query($query)){

            if($term->num_rows == 1) {

                $fila = $term->fetch_object();
            }
        }
    }

```

```

        $codigo['term_id'] = $fila->term_id;
        $codigo['tax_id'] = $fila->term taxonomy id;

    }else if($term->num_rows > 1){

        while ($fila = $term->fetch_assoc()) {

            $codigo['tax_id'][] = $fila['term taxonomy id'];
            $codigo['term_id'][] = $fila['term_id'];

        }

    }

    $term->close();

    return $codigo;

}

public function getMarcas(){

    $codigo = false;
    $marcas = $this->db->query("

        SELECT DISTINCT t3.term_taxonomy_id, t1.term_id, t1.slug, t1.name
        FROM {$this->terms} AS t1
        INNER JOIN (

            SELECT t2.term taxonomy id, t2.term id FROM {$this->term taxonomy} AS t2
            WHERE t2.taxonomy = 'marcas'

        ) AS t3

        ON (t1.term_id = t3.term_id)

        ORDER BY t1.name ASC

    ");

    if($marcas){

        if($marcas->num_rows > 0){

            while ($fila = $marcas->fetch_array()) {

                $filasA[] = $fila;
            }
            $codigo = $filasA;
            $marcas->close();
        }

    }

    return $codigo;

}

public function getPost($post = false, $type = 'certificados', $status =
"publish"){

    $codigo = false;

    if($post){

        $post_name = Globales::sanitize_title($post);

```

```

        $id = $post;

        $post = $this->db->prepare("
            SELECT t1.ID, t1.post_content, t1.post_title, t1.post_author,
            t1.post_metrologo, t1.post_director
            FROM $this->table_cer AS t1
            WHERE (t1.post_name = ? OR t1.ID = ? OR t1.post_nit = ?)
            AND t1.post_type = ? AND t1.post_status IN ($status)
            LIMIT 1
            ");

        $post->bind_param('ssss', $post_name, $id, $post_name, $type);

        if($post->execute()){

            $result = $post->get_result();

            if($result->num_rows == 1) {

                $fila = $result->fetch_object();
                $post->close();
                $codigo = ['post_id' => $fila->ID,
                'post_content' => $fila->post_content,
                'post_title' => $fila->post_title,
                'post_author' => $fila->post_author,
                'post_metrologo' => $fila->post_metrologo,
                'post_director' => $fila->post_director
                ];
            }
        }

        return $codigo;
    }

    protected function setPost($args){

        $codigo = false;

        if(is_array($args)){

            $author = isset($args['author']) ? $args['author'] : Session::get('id_usu');
            $post_content = isset($args['post_content']) ? $args['post_content'] : '';
            $post_title = isset($args['post_title']) ? $args['post_title'] : '';
            $post_excerpt = isset($args['post_excerpt']) ? $args['post_excerpt'] : '';
            $post_nit = isset($args['post_nit']) ? $args['post_nit'] : '';
            $post_status = isset($args['post_status']) ? $args['post_status'] :
            'borrador';

            $post_type = isset($args['post_type']) ? $args['post_type'] : 'certificados';
            $post_parent = isset($args['post_parent']) ? $args['post_parent'] : '';

            $date = $this->tiempo();
            $date_gmt = $this->timebog;
            $post_name = Globales::sanitize_title($args['post_title']);

            $post = $this->db->prepare("
                INSERT INTO $this->table_cer (post_author, post_date, post_date_gmt,
                post_content, post_title, post_excerpt, post_nit, post_status, post_name, post_modified,
                post_modified_gmt, post_parent, post_type)
                VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
                ");

```

```

        $post->bind_param('ssssssssssss', $author, $date, $date_gmt, $post_content,
$post title, $post excerpt, $post nit, $post status, $post name, $date, $date_gmt,
$post_parent, $post_type);

        $post->execute();

        if($last_Id = $post->insert_id){

            $codigo = $last_Id;

        }

        return $codigo;

    }

}

protected function updatePost($args, $id){

    $codigo = false;
    $id = abs(intval($id));
    $set = array();
    $params = array();

    if(is_array($args) and !empty($id) and !empty($args)){

        if(!empty($args['post_content'])){
            $params[] = $args['post_content'];
            $set[] = 'post_content = ?';
        }

        if(!empty($args['post_title'])){
            $params[] = $args['post_title'];
            $set[] = 'post_title = ?';
        }

        if(!empty($args['post_name'])){
            $params[] = $args['post_name'];
            $set[] = 'post_name = ?';
        }

        if(!empty($args['post_metrologo'])){
            $params[] = $args['post metrologo'];
            $set[] = 'post_metrologo = ?';
        }

        if(!empty($args['post_director'])){
            $params[] = $args['post_director'];
            $set[] = 'post director = ?';
        }

        if(!empty($args['post_excerpt'])){
            $params[] = $args['post excerpt'];
            $set[] = 'post_excerpt = ?';
        }

        if(!empty($args['post_nit'])){
            $params[] = $args['post_nit'];
            $set[] = 'post nit = ?';
        }

        if(!empty($args['post_status'])){
            $params[] = $args['post_status'];
            $set[] = 'post_status = ?';
        }
    }
}

```

```

    }

    $params[] = $this->tiempo();
    $set[] = 'post_date = ?';

    $params[] = $this->timebog;
    $set[] = 'post_date_gmt = ?';

    if(!empty($args['post_parent'])){
        $params[] = $args['post_parent'];
        $set[] = 'post_parent = ?';
    }

    $params[] = $id;

    $len = count($params) - 1;
    if($len){
        $type = '';
        for($i=0; $i<=$len; $i++){
            $type .= 's';
        }

        $type = array($type);
        $params = array_merge($type, $params);

        $refs = array();

        foreach($params as $key => $value){
            $refs[$key] = &$params[$key];
        }

        $set = implode(' ', $set);

        $updatePost = $this->db->prepare("

            UPDATE {$this->table_cer}
            SET $set
            WHERE ID = ?

        ");

        call_user_func_array(array($updatePost, "bind_param"), $refs);

        $updatePost->execute();

        if($updatePost->affected_rows){
            $codigo = true;
        }

        return $codigo;
    }
}

protected function setTerm($name, $taxonomy){

    $codigo = false;

    $slug = Globales::sanitize_title($name);

```



```

$getTerm = $this->getTerm($slug, $taxonomy);

if(!$this->_commit){
    $this->_db->autocommit(FALSE);
}

if($getTerm['term_id']){
    return $codigo;
}

$term = $this->_db->prepare("
    INSERT INTO $this->terms (name, slug)
    VALUES (?, ?)
");

$term->bind_param('ss', $name, $slug);

$term->execute();

if($last_Id = $term->insert_id){
    $term->close();

    $taxonomy term = $this->_db->prepare("
        INSERT INTO {$this->term_taxonomy} (term_id, taxonomy)
        VALUES (?, ?)
    ");

    $taxonomy term->bind param('ss', $last_Id, $taxonomy);

    if($taxonomy_term->execute()){

        if(!$this->_commit){
            $this->_db->commit();
        }

        $codigo = $taxonomy_term->insert_id;
        $taxonomy_term->close();

    }

}

if(!$this->_commit){
    if(!$codigo){
        $this->_db->rollback();
    }
}

return $codigo;
}

protected function setRelation($object_id, $term_taxonomy_id){

    $codigo = false;

    if(!$this->_commit){
        $this->_db->autocommit(FALSE);
    }

    $relation = $this->_db->prepare("

```

```

        INSERT INTO $this->term_relationships (object_id, term_taxonomy_id)
        VALUES (?, ?)
    );

    $relation->bind_param('ii', $object_id, $term_taxonomy_id);

    if($relation->execute()){

        if($relation->affected_rows){

            $relation->close();

            $count = $this->_db->prepare("
                UPDATE $this->term_taxonomy
                SET count = count + 1
                WHERE term_taxonomy_id = ?
            ");

            $count->bind_param('i', $term_taxonomy_id);

            if($count->execute()){

                if($count->affected_rows){

                    $count->close();

                    if(!$this->_commit){
                        $this->_db->commit();
                    }

                    $codigo = true;
                }
            }
        }

        if(!$this->_commit){
            if(!$codigo){
                $this->_db->rollback();
            }
        }

        return $codigo;

    }

    protected function deleteRelation($object_id, $term_taxonomy_id){

        $codigo = false;

        if(!$this->_commit){
            $this->_db->autocommit(FALSE);
        }

        $deleteRelation = $this->_db->prepare("

            DELETE FROM {$this->term_relationships}
            WHERE object_id = ? AND term_taxonomy_id = ?
        ");

        $deleteRelation->bind_param('ii', $object_id, $term_taxonomy_id);

        if($deleteRelation->execute()){

```

```

        if($deleteRelation->affected_rows){

            $deleteRelation->close();

            $count = $this->_db->prepare("
                UPDATE $this->term_taxonomy
                SET count = count - 1
                WHERE term_taxonomy id = ?
            ");

            $count->bind_param('i', $term_taxonomy_id);

            if($count->execute()){

                if($count->affected_rows){

                    $count->close();

                    if(!$this->_commit){
                        $this->_db->commit();
                    }

                    $codigo = true;
                }
            }
        }

        if(!$this->_commit){
            if(!$codigo){
                $this->_db->rollback();
            }
        }

        return $codigo;
    }

    protected function countTax($object_id, $taxID = false, $action = 'sum'){

        $codigo = false;

        if(empty($object_id)){
            return $codigo;
        }

        $object_id = abs(intval($object_id));

        if(!$taxID){

            if($action === 'sum'){
                $query = "
                    UPDATE $this->term_taxonomy AS t1
                    LEFT JOIN {$this->term_relationships} grelations ON (t1.term_taxonomy_id
= grelations.term_taxonomy id)
                    SET t1.count = count + 1
                    WHERE grelations.object id = ?
                ";
            }

            if($action === 'rest'){

                $query = "
                    UPDATE $this->term_taxonomy AS t1
                    LEFT JOIN {$this->term_relationships} grelations ON (t1.term_taxonomy_id
= grelations.term_taxonomy_id)

```

```

        SET t1.count = count - 1
        WHERE grelations.object id = ?
        ";
    }

    }else{

        if($action === 'sum'){
            $query = "
            UPDATE $this->term_taxonomy AS t1
            SET t1.count = count + 1
            WHERE t1.term_taxonomy_id = ?
            ";
        }

        if($action === 'rest'){

            $query = "
            UPDATE $this->term_taxonomy AS t1
            SET t1.count = count - 1
            WHERE t1.term_taxonomy_id = ?
            ";
        }

    }

    $count = $this->_db->prepare("
    {$query}
    ");

    $count->bind param('i', $object id);

    if($count->execute()){

        if($count->affected_rows){
            $count->close();
            $codigo = true;
        }

    }

    return $codigo;
}

protected function getMeta($id = false, $meta_key = false){

    $value = false;
    $ids = '';
    if(is_array($id)){

        $ids = implode(' ', $id);

    }else{

        if($id){
            $ids = abs(intval($id));
        }else{
            return false;
        }

    }

    if($meta_key){

```

```

        $meta_key = "AND t2.meta_key = '$meta_key'";
    }else{
        $meta_key = '';
    }

    $query = "
    SELECT t2.post_id, t2.meta_value
    FROM {$this->certi meta} AS t2

    WHERE t2.post_id IN ($ids) {$meta_key}
    ";

    if($meta = $this->_db->query("

        {$query}

    ")){

        if($meta->num_rows >= 1){

            while ($fila = $meta->fetch_assoc()) {

                $value[$fila['post_id']] = $fila['meta_value'];

            }

            $meta->close();

        }

    }

    return $value;

}

protected function getUserMeta($id = false, $meta_key = false){

    $value = false;
    $ids = '';
    if(is_array($id)){

        $ids = implode(', ', $id);

    }else{

        if($id){

            $ids = abs(intval($id));

        }else{

            return $value;

        }

    }

    if($meta_key){

        $meta_key = "AND t2.meta_key = '$meta_key'";
    }else{

        $meta_key = '';
    }

    $query = "
    SELECT t2.user_id, t2.meta_value
    FROM {$this->usermeta} AS t2

    WHERE t2.user_id IN ($ids) {$meta_key}

```

```

";

    if($meta = $this->_db->query("

        {$query}

    ")) {

        if($meta->num_rows >= 1) {

            while ($fila = $meta->fetch_assoc()) {

                $value[$fila['user_id']] = $fila['meta_value'];

            }

            $meta->close();

        }

    }

    return $value;
}

public function getAuthor($id = false) {

    $value = false;
    $ids = '';
    if(is_array($id)) {

        $len = count($id) - 1;

        for($i = 0; $i <= $len; $i++) {

            if($i == $len) {

                $ids .= $id[$i];

            } else {

                $ids .= $id[$i] . ',';

            }

        }

    } else {

        if($id) {

            $ids = abs(intval($id));

        } else {

            return false;

        }

    }

    if($author = $this->_db->query("

        SELECT t2.ID, t2.display_name
        FROM {$this->table usu} AS t2
        WHERE t2.ID IN ($ids)

    ")) {

        if($author->num_rows >= 1) {

            while ($fila = $author->fetch_assoc()) {

```

```

        $value[$fila['ID']] = $fila['display name'];
    }

    $author->close();

}

}

return $value;

}

protected function lastCerti($equipo){
    $return = false;

    $query = "
        SELECT DISTINCT t2.ID, t2.post_title, t2.post_name, gtaxonomy.term_taxonomy_id
        FROM {$this->table_cer} AS t2
        LEFT JOIN {$this->term_relationships} grelations ON (t2.ID =
greations.object_id)
        LEFT JOIN {$this->term_taxonomy} gtaxonomy ON (greations.term_taxonomy_id =
gtaxonomy.term_taxonomy_id)
        LEFT JOIN {$this->terms} gterms ON (gtaxonomy.term_id = gterms.term_id)

        WHERE t2.post_status IN ('borrador', 'revisar', 'revisando', 'verificar',
'verificando', 'verificado')
        AND t2.post_type = 'certificados'
        AND gtaxonomy.taxonomy = 'equipos'
        AND gterms.slug = '$equipo'

        ORDER BY t2.post_name DESC
        LIMIT 1
    ";

    if($borrador = $this->_db->query($query)){

        if($borrador->num_rows == 1){

            $fila = $borrador->fetch_assoc();
            $borrador->close();
            $return = $fila;
        }
    }

    return $return;

}

protected function equalCerti($equipo, $title){

    $return = false;
    $search = Globales::sanitize_title($title);
    $time = explode('-', $this->_timebog);
    $time = substr($time[0], -2) . $time[1];

    $query = "
        SELECT DISTINCT t2.ID, t2.post_title, t2.post_name
        FROM {$this->table_cer} AS t2
        LEFT JOIN {$this->term_relationships} grelations ON (t2.ID =
greations.object_id)

```

```

LEFT JOIN {$this->term_taxonomy} gtaxonomy ON (grelations.term_taxonomy_id =
gtaxonomy.term_taxonomy_id)
LEFT JOIN {$this->terms} gterms ON (gtaxonomy.term_id = gterms.term_id)

WHERE
(t2.post_author != '{$this->user_id}' AND t2.post_name = '{$search}'
AND t2.post_status IN ('borrador', 'revisar', 'revisando', 'verificar',
'verificando', 'verificado')
AND t2.post_type = 'certificados'
AND gtaxonomy.taxonomy = 'equipos'
AND gterms.slug = '{$equipo}')
OR
(t2.post_author = '{$this->user_id}' AND t2.post_name = '{$search}'
AND t2.post_status IN ('revisar', 'revisando', 'verificar', 'verificando',
'verificado')
AND t2.post_type = 'certificados'
AND gtaxonomy.taxonomy = 'equipos'
AND gterms.slug = '{$equipo}')

LIMIT 1
";

if($author = $this->_db->query($query)){
    if($author->num_rows == 1){
        $author->close();

        if($fila = $this->lastCerti($equipo)){
            $consecutivo = explode('-', $fila['post_title']);
            $incremento = intval($consecutivo[1]) + 1;
            $title = $consecutivo[0] . '-' . $incremento;
            $return = $title;
        }
    }
}

return $return;
}

protected function getConsecutivo($equipo = 'electrobisturi'){
    $return = false;
    $time = explode('-', $this->timebog);
    $time = substr($time[0], -2) . $time[1];

    if($equipo == 'electrobisturi'){
        $name = 'Electrobisturí';
        $title = 'GMEL' . $time . '-1';
    }

    $query = "
SELECT DISTINCT t2.ID, t2.post_title, t2.post_name
FROM {$this->table cer} AS t2
LEFT JOIN {$this->term_relationships} grelations ON (t2.ID =
grelations.object_id)
LEFT JOIN {$this->term_taxonomy} gtaxonomy ON (grelations.term_taxonomy_id =
gtaxonomy.term_taxonomy_id)
LEFT JOIN {$this->terms} gterms ON (gtaxonomy.term_id = gterms.term_id)

WHERE t2.post_author = '{$this->user_id}' AND t2.post_status = 'borrador'
AND t2.post_type = 'certificados'
AND gtaxonomy.taxonomy = 'equipos'

```



```

        AND gterms.slug = '$equipo'

        ORDER BY t2.post_name DESC
        LIMIT 1
        FOR UPDATE
        ";

        $this->_commit = $this->_db->autocommit(FALSE);

        if($borrador = $this->_db->query($query)){

            if($borrador->num_rows == 1){

                $fila = $borrador->fetch_assoc();
                $borrador->close();
                $searchA = $fila['post_name'];
                $consecutivo = explode('-', $fila['post_title']);
                $ID = $fila['ID'];

                if(intval(substr($consecutivo[0], -4)) < intval($time)){

                    $titleLast = $title;

                    $args = array(
                        'post title' => $titleLast,
                        'post_name' => Globales::sanitize_title($titleLast)
                    );

                    if($this->updatePost($args, $ID)){

                        $this-> db->commit();

                    }

                }else{

                    $titleLast = $fila['post_title'];

                }

                if($equalCerti = $this->equalCerti($equipo, $titleLast)){

                    $args = array(
                        'post title' => $equalCerti,
                        'post_name' => Globales::sanitize_title($equalCerti)
                    );

                    if($this->updatePost($args, $ID)){

                        $this-> db->commit();
                        $return = array('post_id' => $ID, 'post_title' => $equalCerti);

                    }

                }else{

                    $return = array('post_id' => $ID, 'post_title' => $titleLast);

                }

            }else{

                if($fila = $this->lastCerti($equipo)){

                    $consecutivo = explode('-', $fila['post_title']);

                    if(intval(substr($consecutivo[0], -4)) < intval($time)){

                        $titleLast = $title;

                    }else{

```

```

        $incremento = intval($consecutivo[1]) + 1;
        $titleLast = $consecutivo[0] . '-' . $incremento;
    }

    if($equalCerti = $this->equalCerti($equipo, $titleLast)){

        $args = array(
            'post_title' => $equalCerti
        );

        if($id_post = $this->setPost($args)){

            if($id_relation = $this->setRelation($id_post,
$fila['term_taxonomy_id'])){

                if($this->countTax($fila['term_taxonomy_id'], true,
'rest')){

                    $this-> db->commit();
                    $return = array('post_id' => $id_post, 'post_title'
=> $equalCerti);
                }
            }
        }else{

            $args = array(
                'post_title' => $titleLast
            );

            if($id_post = $this->setPost($args)){

                if($id_relation = $this->setRelation($id_post,
$fila['term_taxonomy_id'])){

                    if($this->countTax($fila['term_taxonomy_id'], true,
'rest')){

                        $this->_db->commit();
                        $return = array('post_id' => $id_post, 'post_title'
=> $titleLast);
                    }
                }
            }
        }else{

            $args = array(
                'post title' => $title
            );

            if($id_post = $this->setPost($args)){

                if($id_termtax = $this->setTerm($name, 'equipos')){

                    if($id_relation = $this->setRelation($id_post, $id_termtax)){

                        if($this->countTax($id_termtax, true, 'rest')){

                            $this-> db->commit();
                            $return = array('post_id' => $id_post, 'post_title'
=> $title);
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
    }
    }
    }

    if(!$return){
        $this-> db->rollback();
    }

    return $return;

}

protected function searchPost($search, $postType='certificados',
$postStatus='publish', $limit=10){

    $codigo = false;

    if(empty($search)){

        return $codigo;

    }

    $limit = intval($limit);

    $this->sanitize($search);
    $searchC = "";

    $porciones = explode(" ", $search);
    $Plength = count($porciones) - 1;

    if($Plength){
        for($i=0; $i<=$Plength; $i++){

            if($i == $Plength){

                $part = preg_match('/[+\-><\(\)\~*\"@]+/', $porciones[$i]) ?
                ''.$porciones[$i].'' : $porciones[$i].'*';
                $searchC .= '+' . $part;
            }else{

                $part = preg_match('/[+\-><\(\)\~*\"@]+/', $porciones[$i]) ?
                ''.$porciones[$i].'' : $porciones[$i].'*';
                $searchC .= '+' . $part;
            }
        }
    }else{

        $part = preg_match('/[+\-><\(\)\~*\"@]+/', $search) ? ''.$search.''' :
        $search.*';
        $searchC = '+' . $part;
    }

    $searchclients = $this->_db->prepare("

    SELECT *
    FROM {$this->table_cer} AS t1
    INNER JOIN (

    SELECT t2.ID,
    MATCH (t2.post_excerpt) AGAINST (? IN BOOLEAN MODE) AS Relevancel,

```

```

        MATCH (t2.post_title) AGAINST (? IN BOOLEAN MODE) AS Relevance2,
        MATCH (t2.post_nit) AGAINST (? IN BOOLEAN MODE) AS Relevance3

    FROM {$this->table_cer} AS t2
    WHERE (MATCH (t2.post_excerpt) AGAINST (? IN BOOLEAN MODE)
    OR MATCH (t2.post_title) AGAINST (? IN BOOLEAN MODE)
    OR MATCH (t2.post_nit) AGAINST (? IN BOOLEAN MODE))
    AND t2.post_status IN (?) AND t2.post_type IN (?)

    ORDER BY Relevance1 * 3 + Relevance2 * 6 + Relevance3 * 9 DESC

    LIMIT $limit

    ) AS t3

    ON (t1.ID = t3.ID)

    ");

    $searchclients->bind_param('ssssssss', $searchC, $searchC, $searchC, $searchC,
    $searchC, $searchC, $postStatus, $postType);

    if($searchclients->execute()){

        $result = $searchclients->get_result();

        if($result->num_rows > 0) {

            while ($fila = $result->fetch_assoc()) {

                $unserialize = Globales::maybe_unserialize($fila['post_content']);

                $filasA['ID'] = $fila['ID'];
                $filasA['ID_signature'] = Globales::hashValue($fila['ID']);
                $filasA['post_title'] = $fila['post_title'];
                $filasA['post_title signature'] =
Globales::hashValue($fila['post_title']);
                $filasA['post_name'] = $fila['post_name'];
                $filasA['post_nit'] = $fila['post_nit'];
                $filasA['post_date_gmt'] = $fila['post_date_gmt'];
                $filasA['post_author'] = $this->getAuthor($fila['post_author']);
                $filasA['post_content'] = $fila['post_content'];
                $filasA['post_excerpt'] = $fila['post_excerpt'];
                $filasA['post_parent'] = $fila['post_parent'];
                $filasA['post_content_unser'] = $unserialize;

                $codigoA[] = $filasA;

            }

            $searchclients->close();
            $codigo = $codigoA;

        }

    }

    return $codigo;
}

}
?>

```

11.4.5. “VIEW.PHP”

```
<?php

class View
{

    public $_controlador;
    public $_metodo;
    public $argumentos;
    public $_jsView;
    public $_cssView;
    public $_jsLayout;
    public $_cssLayout;

    public function __construct(Request $peticion){

        $this-> controlador = $peticion->getControlador();
        $this->_metodo = $peticion->getMetodo();
        $this-> argumentos = $peticion->getArgs();

    }

    public function renderizarInstall($vista){

        $urllayout = Globales::get home url() . '/views/layout/' . DEFAULT_LAYOUT . '/';
        $rutaVista = ABSPATH . 'views' . DS . $this->_controlador . DS . $vista .
        '.phtml';

        if(is_readable($rutaVista)){

            include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
            'instalacion' . DS . 'header.php';
            include_once $rutaVista;
            include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
            'instalacion' . DS . 'footer.php';
            exit;

        }else{

            throw new Exception('Error de vista login');

        }

    }

    public function renderizarLogin($vista){

        $urllayout = BASE_URL . '/views/layout/' . DEFAULT_LAYOUT . '/';
        $rutaVista = ABSPATH . 'views' . DS . $this->_controlador . DS . $vista .
        '.phtml';

        if(is_readable($rutaVista)){

            include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
            'login' . DS . 'header.php';
            include_once $rutaVista;
            include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
            'login' . DS . 'footer.php';
            exit;

        }else{

        }
```

```

        throw new Exception('Error de vista login');
    }
}

public function renderizar($vista){
    $sidebar_left = array();

    $sidebar_left[] = array(
        'id' => 'index',
        'titulo' => 'Inicio',
        'url' => BASE_URL,
        'icon' => '<i class="fa fa-home" aria-hidden="true"></i>',
        'dropdown' => false
    );

    $sidebar_left[] = array(
        'id' => 'certificados',
        'titulo' => 'Certificados',
        'url' => false,
        'icon' => '<i class="fa fa-edit"></i>',
        'dropdown' => array(
            array(
                'id' => 'bascula',
                'titulo' => 'Bascula',
                'url' => BASE_URL.'/certificados/bascula'
            ),
            array(
                'id' => 'capnografo',
                'titulo' => 'Capnógrafo',
                'url' => BASE_URL.'/certificados/capnografo'
            ),
            array(
                'id' => 'electrobisturi',
                'titulo' => 'Electrobisturi',
                'url' => BASE_URL.'/certificados/electrobisturi'
            ),
            array(
                'id' => 'monitor',
                'titulo' => 'Monitor',
                'url' => BASE_URL.'/certificados/monitor'
            ),
            array(
                'id' => 'termohigrometro',
                'titulo' => 'Termohigrómetro',
                'url' => BASE_URL.'/certificados/termohigrometro'
            )
        )
    );

    $sidebar_left[] = array(
        'id' => 'clientes',
        'titulo' => 'Clientes',
        'url' => false,
        'icon' => '<i class="fa fa-users" aria-hidden="true"></i>',
        'dropdown' => array(
            array(
                'id' => 'listaclientes',
                'titulo' => 'Lista Clientes',
                'url' => BASE_URL.'/clientes/listaclientes'
            ),
            array(

```

```

        'id' => 'nuevocliente',
        'titulo' => 'Nuevo Cliente',
        'url' => BASE_URL.'/clientes/nuevocliente'
    )

    )
);

if(Session::accesoView(1)){

    $sidebar_left[] = array(
        'id' => 'patrones',
        'titulo' => 'Patrones',
        'url' => false,
        'icon' => '<i class="fa fa-database" aria-hidden="true"></i>',
        'dropdown' => array(
            array(
                'id' => 'lista',
                'titulo' => 'Lista Patrones',
                'url' => BASE_URL.'/patrones/lista'
            ),
            array(
                'id' => 'crear',
                'titulo' => 'Crear Patrón',
                'url' => BASE_URL.'/patrones/crear'
            )
        )
    );
}

if(Session::accesoViewEstricto(array(0))){

    $sidebar_left[] = array(
        'id' => 'administracion',
        'titulo' => 'Administración',
        'url' => false,
        'icon' => '<i class="fa fa-line-chart" aria-hidden="true"></i>',
        'dropdown' => array(
            array(
                'id' => 'usuarios',
                'titulo' => 'Usuarios',
                'url' => BASE_URL.'/administracion/usuarios'
            ),
            array(
                'id' => 'usuarioseliminados',
                'titulo' => 'Usuarios Eliminados',
                'url' => BASE_URL.'/administracion/usuarioseliminados'
            ),
            array(
                'id' => 'nuevousuario',
                'titulo' => 'Nuevo Usuario',
                'url' => BASE_URL.'/administracion/nuevousuario'
            )
        )
    );

    $sidebar_left[] = array(
        'id' => 'aplicacion',
        'titulo' => 'Aplicación',
        'url' => false,
        'icon' => '<i class="fa fa-cogs" aria-hidden="true"></i>',
        'dropdown' => array(
            array(
                'id' => 'respaldo',
                'titulo' => 'Respaldo',

```

```

        'url' => BASE_URL.'/aplicacion/respaldo'
    )
    );
}

    $urllayout = BASE_URL . '/views/layout/' . DEFAULT_LAYOUT . '/';
    $rutaVista = ABSPATH . 'views' . DS . $this->_controlador . DS . $vista .
    '.phtml';

    if(is_readable($rutaVista)){

        include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
        'header.php';
        include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
        'cabecera.php';
        include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
        'sidebar-left.php';
        include_once $rutaVista;
        include_once ABSPATH . 'views' . DS . 'layout' . DS . DEFAULT_LAYOUT . DS .
        'footer.php';
        exit;

    }else{

        throw new Exception('Error de vista');

    }

}

    public function setJsView(array $js){

        if(is_array($js) && count($js)){

            foreach($js as $jsscript){

                $this->_jsView[] = BASE_URL . '/views/' . $this->_controlador . '/js/' .
                $jsscript . '.js';
            }
        }else{

            throw new Exception('Error de js View');

        }

    }

    public function setCssView(array $css){

        if(is_array($css) && count($css)){

            foreach($css as $acss){

                $this->_cssView[] = BASE_URL . '/views/' . $this->_controlador . '/css/'
                . $acss . '.css';
            }
        }else{

            throw new Exception('Error de css View');

        }

    }
}

```



```

    public function getPagination(array $pages) {

        $args = $this->_argumentos;
        $argsUrl = '';
        $i = 1;
        $len = count($args);

        if($len){
            foreach ($args as $arg) {

                if($i == $len){
                    $argsUrl .= '/' . $arg;
                }else{
                    $argsUrl .= '/' . $arg . '/';
                }

                $i++;
            }
        }

        $url = BASE_URL . '/' . $this->_controlador . '/' . $this->_metodo . $argsUrl;

        $salida = array();

        if(is_array($pages) && count($pages)){

            $salida['paginas'] = '<div id="paginacion" class="visible">';

            if(isset($pages['anterior'])){

                $salida['paginas'] .= '<a class="anterior" href="'. $url . '?page=' .
                $pages['anterior'] . '&limite=' . $pages['limite'] . '"><< Anterior</a>';

            }

            if(isset($pages['estado1'])){

                foreach($pages['pages'] as $page) {

                    if($pages['actual'] == $page){

                        $salida['paginas'] .= '<span class="actual">' . $page . '</span>';

                    }else{

                        $salida['paginas'] .= '<a class="pages" href="'. $url . '?page=' .
                        $page . '&limite=' . $pages['limite'] . '">' . $page . '</a>';

                    }
                }

            }elseif(isset($pages['estado2'])){

                foreach($pages['pages'] as $page) {

                    if($pages['actual'] == $page){

                        $salida['paginas'] .= '<span class="actual">' . $page . '</span>';

                    }else{

                        $salida['paginas'] .= '<a class="pages" href="'. $url . '?page=' .
                        $page . '&limite=' . $pages['limite'] . '">' . $page . '</a>';

                    }
                }

            }

        }

    }

```

```

    }

    }

    $salida['paginas'] .= '<span>...</span>';
    $salida['paginas'] .= '<a class="ultimo" href="'. $url . '?page=' .
$pages['ultimo'] . '&limite=' . $pages['limite'] . '>'. $pages['ultimo'] . '</a>';

    }
    elseif(isset($pages['estado3'])) {

        $salida['paginas'] .= '<a class="ultimo" href="'. $url . '?page=' .
$pages['primero'] . '&limite=' . $pages['limite'] . '>'. $pages['primero'] . '</a>';
        $salida['paginas'] .= '<span>...</span>';

        foreach($pages['pages'] as $page) {

            if($pages['actual'] == $page) {

                $salida['paginas'] .= '<span class="actual">'. $page . '</span>';

            } else {

                $salida['paginas'] .= '<a class="pages" href="'. $url . '?page='
. $page . '&limite=' . $pages['limite'] . '>'. $page . '</a>';

            }

        }

        $salida['paginas'] .= '<span>...</span>';
        $salida['paginas'] .= '<a class="ultimo" href="'. $url . '?page=' .
$pages['ultimo'] . '&limite=' . $pages['limite'] . '>'. $pages['ultimo'] . '</a>';

    }
    elseif(isset($pages['estado4'])) {

        $salida['paginas'] .= '<a class="ultimo" href="'. $url . '?page=' .
$pages['primero'] . '&limite=' . $pages['limite'] . '>'. $pages['primero'] . '</a>';
        $salida['paginas'] .= '<span>...</span>';

        foreach($pages['pages'] as $page) {

            if($pages['actual'] == $page) {

                $salida['paginas'] .= '<span class="actual">'. $page . '</span>';

            } else {

                $salida['paginas'] .= '<a class="pages" href="'. $url . '?page='
. $page . '&limite=' . $pages['limite'] . '>'. $page . '</a>';

            }

        }

    }

    if(isset($pages['siguiente'])) {

        $salida['paginas'] .= '<a class="siguiente" href="'. $url . '?page=' .
$pages['siguiente'] . '&limite=' . $pages['limite'] . '>Siguiente »</a>';

    }

```

```

        $salida['paginas'] .= '</div>';

        $class25 = '';
        $class50 = '';
        $class100 = '';
        $class200 = '';

        switch ($pages['limite']) {

            case 50:
                $class50 = ' select';
                break;

            case 100:
                $class100 = ' select';
                break;

            case 200:
                $class200 = ' select';
                break;

            default:
                $class25 = ' select';

        }

        $salida['limite'] = '<div id="box-limite">';
        $salida['limite'] .= '<button type="button">Numero de Filas <i class="fa fa-angle-down" aria-hidden="true"></i></button>';
        $salida['limite'] .= '<ul class="limites">';
        $salida['limite'] .= '<li class="limite'. $class25 .' ">';
        $salida['limite'] .= '<a href="'. $url .' ">25</a>';
        $salida['limite'] .= '</li>';
        $salida['limite'] .= '<li class="limite'. $class50 .' ">';
        $salida['limite'] .= '<a href="'. $url .' "?page=1&limite=50' .' ">50</a>';
        $salida['limite'] .= '</li>';
        $salida['limite'] .= '<li class="limite'. $class100 .' ">';
        $salida['limite'] .= '<a href="'. $url .' "?page=1&limite=100' .' ">100</a>';
        $salida['limite'] .= '</li>';
        $salida['limite'] .= '<li class="limite'. $class200 .' ">';
        $salida['limite'] .= '<a href="'. $url .' "?page=1&limite=200' .' ">200</a>';
        $salida['limite'] .= '</li>';
        $salida['limite'] .= '</ul>';
        $salida['limite'] .= '</div>';

        return $salida;

    }else{

        throw new Exception('Error de getPagination View');

    }

}

?>

```