



UNIVERSIDAD TÉCNICA DE AMBATO

**FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS**

Tema:

APLICACIÓN WEB DE MONITOREO DEL CENTRO DE PROCESAMIENTO DE DATOS MEDIANTE SENSORES DE SEGURIDAD EN LA COOPERATIVA DE AHORRO Y CRÉDITO INDÍGENA "SAC LTDA." DE LA CIUDAD DE AMBATO.

Trabajo de Grado. Modalidad: TEMI. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

Sublínea de Investigación: Aplicación Web.

AUTOR: Diego Fernando Poaquiza Azogue.

TUTOR: Ing. Mg. Elsa Pilar Urrutia Urrutia.

Ambato - Ecuador

Abril/2015

APROBACIÓN DEL TUTOR

En mi calidad de Tutora del trabajo de investigación sobre el tema: “**Aplicación Web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena SAC Ltda. de la ciudad de Ambato.**”, del señor Diego Fernando Poaquiza Azogue, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el Art. 16 del Capítulo II, del Reglamento de Graduación para obtener el título terminal de tercer nivel de la Universidad Técnica de Ambato.

Ambato Abril 23, 2015

EL TUTOR

Ing. Mg. Elsa Pilar Urrutia Urrutia

AUTORÍA

El presente trabajo de investigación titulado: **“Aplicación Web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena SAC Ltda de la ciudad de Ambato.”**, es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato Abril 23, 2015

Sr. Diego Fernando Poaquiza Azogue

CC: 180449432-4

APROBACIÓN DE LA COMISIÓN CALIFICADORA

La Comisión Calificadora del presente trabajo conformada por los señores docentes Ing. Carlos Núñez e Ing. Patricio Gonzalez, revisó y aprobó el Informe Final del trabajo de graduación titulado “Aplicación Web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena SAC Ltda de la ciudad de Ambato”, presentado por el señor Diego Fernando Poaquiza Azogue de acuerdo al Art. 17 del Reglamento de Graduación para obtener el título Terminal de tercer nivel de la Universidad Técnica de Ambato.

Ing. Vicente Morales L., Mg.

PRESIDENTE DEL TRIBUNAL

Ing. Carlos Núñez, Mg.

DOCENTE CALIFICADOR

Ing. Patricio Gonzalez, Mg.

DOCENTE CALIFICADOR

DEDICATORIA

A ti Dios que me diste la oportunidad de vivir y regalarme una familia maravillosa.

Con mucho cariño y amor a mis padres César Poaquiza Martínez y María Azogue Ayme, porque creyeron en mí, me apoyaron a pesar de tantas circunstancias dándome ejemplos dignos de superación y entrega, en gran parte gracias a ustedes hoy puedo ver alcanzado mi meta ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera y de mi vida personal, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final. Se lo dedico a ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí

A mi hermano Wilson por brindare su respaldo cuando lo necesite para cumplir mi objetivo, y a mi hermana Martha, gracias hermanos fueron pilares fundamentales en mi vida para lograr los éxitos conseguidos durante mi carrera profesional con sus ejemplos y concejos.

AGRADECIMIENTO

A Dios, por las bendiciones que me dio para llegar a culminar con éxito uno de los mayores anhelos de mi vida.

A toda mi familia por los valores que me han inculcado, por haberme dado la oportunidad de tener una excelente educación y sobre todo por ser un excelente ejemplo de vida a seguir, a mis amigos quienes me brindaron su apoyo incondicional en lo que he necesitado y por la confianza que en mi depositaron.

A la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato por haberme recibido en sus aulas y a mis maestros por compartir sus sabios conocimientos que encaminaron al desarrollo exitoso de mi vida estudiantil.

A mi Directora de tesis la Ing. Pilar Urrutia, quien con su experiencia ha sabido brindarme el asesoramiento y consejo oportuno. A los distinguidos miembros del Tribunal de Grado por su apoyo brindado que ha aportado al desarrollo y culminación del presente proyecto.

ÍNDICE DE CONTENIDOS

PÁGINAS PRELIMINARES

Portada.....	i
Aprobación del tutor.....	ii
Autoría	iii
Aprobación de la comisión calificadora	iv
Dedicatoria.....	v
Agradecimiento.....	vi
Índice de contenidos.....	vii
Índice de figuras.....	ix
Índice de tablas	xi
Resumen ejecutivo.....	xii
Summary	xiv
Glosario de términos.....	xv
Introducción.....	xvii

CAPÍTULO I EL PROBLEMA

1.1 Tema.....	1
1.2 Planteamiento del problema.....	1
1.3 Delimitación del problema	3
1.4 Justificación	4
1.5 Objetivos	4
1.5.1 Objetivo general.....	4
1.5.2 Objetivos específicos.....	5

CAPÍTULO II MARCO TEÓRICO

2.1 Antecedentes investigativos	6
2.2 Fundamentación teórica	7
2.3 Propuesta de solución	22

CAPÍTULO III METODOLOGÍA

3.1 Modalidad de la investigación.....	23
--	----

3.2	Población y muestra	23
3.3	Recolección de información	23
3.4	Procesamiento y análisis de datos	24
3.5	Desarrollo del proyecto.....	24

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4	Metodología	27
4.1	Análisis.....	27
4.1.1	Análisis de los procesos de seguridad física utilizados en la cooperativa de ahorro y crédito indígena “sac Ltda.” de la ciudad de ambato.....	27
4.1.2	Análisis y procesamiento de la información recolectada de los procesos de seguridad utilizados en el cpd.....	29
4.1.3	Análisis y requerimientos de la aplicación web	30
4.1.4	Definición de los requerimientos software y hardware para el desarrollo.....	32
4.2	Diseño.....	64
4.2.1	Diseño de la base de datos	65
4.2.2	Diseño de la interfaz	69
4.3	Implementación	85
4.3.1	Implementación hardware arduino uno.....	86
4.3.2	Implementación software arduino uno	88
4.3.3	Implementación de la base de datos, interfaz de usuario y código fuente.....	97
4.4	Pruebas.....	116
4.4.1	Pruebas de caja blanca.....	116
4.4.2	Pruebas de caja negra	119

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1	Conclusiones.....	123
5.2	Recomendaciones	125
	Bibliografía	126
	Anexos.....	129

ÍNDICE DE FIGURAS

Fig. 1: Partes de un Sistema DAQ	8
Fig. 2: Sensor de Temperatura y Humedad DHT11.....	15
Fig. 3: Sensor de Movimiento PIR DYP-ME003	17
Fig. 4: Contacto Magnético modelo 943.....	18
Fig. 5: Cubieboard A20.....	20
Fig. 6: Estructura de la metodología en cascada	21
Fig. 7: Diagrama Casos de Uso Inicio de Sesión Administrador-Operador	38
Fig. 8: Diagrama Casos de Uso Grupos, Usuarios y Zonas - Administrador.....	39
Fig. 9: Diagrama Casos de Uso Dispositivo, Partición, Zonif, y Evento - Administrador	44
Fig. 10: Diagrama Casos de Uso Eventos Alarma Administrador - Operador	50
Fig. 11: Inicio de Sesión	53
Fig. 12: Ingreso de Nuevo Grupo.....	53
Fig. 13: Modificación de Grupo.....	54
Fig. 14: Eliminación de Grupo.....	54
Fig. 15: Ingreso de Nuevo Usuario	55
Fig. 16: Modificación de Usuario	55
Fig. 17: Eliminación de Usuario	56
Fig. 18: Ingreso de Nueva Zona.....	56
Fig. 19: Modificación de Zona.....	57
Fig. 20: Eliminación de Zona.....	57
Fig. 21: Ingreso de nuevo Dispositivo	58
Fig. 22: Modificación de Dispositivo.....	58
Fig. 23: Eliminación de Dispositivo.....	59
Fig. 24: Ingreso de Nueva Partición.....	59
Fig. 25: Modificación de Partición.....	60
Fig. 26: Eliminación de Partición.....	60
Fig. 27: Ingreso de Nueva Zonificación.....	61
Fig. 28: Modificación de Zonificación.....	61
Fig. 29: Eliminación de Zonificación.....	62
Fig. 30: Eliminación de Evento.....	62
Fig. 31 Monitoeo de Zonas	63
Fig. 32: Activar Sistema de Alarma.....	63
Fig. 33: Consulta de eventos – Reportes	64
Fig. 34: Base de Datos Modelo Relacional	65
Fig. 35: Interfaz de la Página Inicio de Sesión.....	70
Fig. 36: Interfaz de la Página Maestra	71
Fig. 37: Interfaz configuración de la cuenta de usuario	72
Fig. 38: Interfaz del Administrador.....	73
Fig. 39: Interfaz del Operador	74
Fig. 40: Interfaz Gestión de Información	75
Fig. 41: Ventana modal ingreso de información	76
Fig. 42: Ventana modal modificación de información.....	77
Fig. 43: Cuadro de dialogo de Eliminación	78

Fig. 44: Interfaz Monitoreo de Zonas	79
Fig. 45: Interfaz de Monitoreo	80
Fig. 46: Interfaz Monitoreo mediante Croquis.....	81
Fig. 47: Interfaz Zonificación	82
Fig. 48: Interfaz Consultar de Eventos.....	83
Fig. 49: Interfaz Reporte de Datos	84
Fig. 50: Arduino Uno	86
Fig. 51: Arduino con Sensores	87
Fig. 52: Entorno de Desarrollo IDE	88
Fig. 53: Información capturada sensores de contacto y movimiento	90
Fig. 54: Información capturada sensor de temperatura	91
Fig. 55: Información temperatura tratada.....	92
Fig. 56: Lectura del puerto serie mediante Python.....	93
Fig. 57: Obtención de la información formato JSON	96
Fig. 58: Implementación Interfaz Inicio de Sesión	97
Fig. 59: Implementación Interfaz Página Administrador.....	99
Fig. 60: Implementación Página Monitoreo de Zonas	100
Fig. 61: Implementación Página Monitoreo de zonas mediante Croquis y Cámara web.....	101
Fig. 62: Implementación ventana emergente cámara web	102
Fig. 63: Implementación ventana emergente cámara web	102
Fig. 64: Implementación Código Modelo Zonas	104
Fig. 65: Vista Index de la Zona.....	105
Fig. 66: Vista Editar registro de la Zona	108
Fig. 67: Vista guardar nuevo registro de la Zona.....	110
Fig. 68: Vista registro seleccionado de la Zona	112
Fig. 69: Grafo de Validación.....	118
Fig. 70: Prueba de Caja Negra información mostrada correctamente.....	119
Fig. 71: Prueba de Caja Negra validación de campos requeridos	120
Fig. 72: Prueba de Caja Negra mensaje de advertencia	120
Fig. 73: Prueba de Caja Negra validación de campos únicos	121
Fig. 74: Prueba de Caja Negra mensaje de advertencia campo único.....	121
Fig. 75: Prueba de Caja Negra validación de campos alfanuméricos	121
Fig. 76: Prueba de Caja Negra mensaje de advertencia campo alfanumérico	122
Fig. 77: Prueba de Caja Negra validación cantidad de caracteres.....	122
Fig. 78: Prueba de Caja Negra mensaje de advertencia cantidad de caracteres	122
Fig. 79: Infraestructura CPD Cooperativa SAC.....	162

ÍNDICE DE TABLAS

Tabla 1: Especificación Técnica Sensor DHT11	16
Tabla 2: Características Sensor PIR DYP-ME003.....	18
Tabla 3: Guía de Observación, Seguridad Física CPD	29
Tabla 4: Cuadro comparativo Frameworks php opensource.....	34
Tabla 5: Cuadro comparativo aspectos generales Frameworks php opensource	34
Tabla 6: Inicio de Sesión.....	38
Tabla 7: Administrar Grupos (Ingresar).....	39
Tabla 8: Administrar Grupos (Editar).....	40
Tabla 9: Administrar Grupos (Eliminar).....	40
Tabla 10: Administrar Usuarios (Ingresar)	41
Tabla 11: Administrar Usuarios (Editar).....	41
Tabla 12: Administrar Usuarios (Eliminar)	42
Tabla 13: Administrar Zonas (Ingresar).....	42
Tabla 14: Administrar Zonas (Editar).....	43
Tabla 15: Administrar Zonas (Eliminar).....	43
Tabla 16: Administrar Dispositivos (Ingresar)	45
Tabla 17: Administrar Dispositivos (Editar).....	45
Tabla 18: Administrar Dispositivos (Eliminar).....	46
Tabla 19: Administrar Particiones (Ingresar).....	47
Tabla 20: Administrar Particiones (Editar).....	47
Tabla 21: Administrar Particiones (Eliminar).....	48
Tabla 22: Administrar Zonificación (Ingresar)	48
Tabla 23: Administrar Zonificación (Editar)	49
Tabla 24: Administrar Zonificación (Eliminar)	49
Tabla 25: Administrar Eventos (Eliminar).....	50
Tabla 26: Activar o Desactivar Sistema de Alarma	51
Tabla 27: Monitorear Zonas.....	51
Tabla 28: Consultar Eventos	52
Tabla 29: Descripción de la Tabla Users	66
Tabla 30: Descripción de la Tabla Fotousuarios.....	66
Tabla 31: Descripción de la Tabla Groups.....	66
Tabla 32: Descripción de la Tabla Dispositivos.....	67
Tabla 33: Descripción de la Tabla Tipos	67
Tabla 34: Descripción de la Tabla Eventos.....	67
Tabla 35: Descripción de la Tabla Zonas.....	68
Tabla 36: Descripción de la Tabla Zonificacions.....	68
Tabla 37: Descripción de la Tabla Particiones.....	68
Tabla 38: Descripción de la Tabla Sistemaestados	69
Tabla 39: Descripción de la Tabla Eventousuarios	69
Tabla 40 Prueba de Caja Blanca	117
Tabla 41 Comprobación camino básico.....	119

RESUMEN EJECUTIVO

La Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato, cuyo objeto social es servir las necesidades financieras de sus socios y de terceros.

Actualmente en el Centro de Procesamiento de Datos cuya área es una de las más importantes ya que ahí se concentran los recursos necesarios para el procesamiento de la información de la organización cuenta con un sistema de alarma con dos sensores de Movimiento y Temperatura, los mismos que están conectados a una Sirena, cumpliendo así solo la función disuasoria activando una sirena, el mismo que no está conectado a una Central de Monitoreo haciendo que dicho sistema no sea eficiente, confiable y seguro, pudiéndose provocar inconvenientes de control de seguridad como robos, daños e incluso una ineficiente gestión de información y bienes en momentos oportunos por personal no autorizado.

Al no disponer de un sistema altamente seguro no se dispone de una aplicación web que permita monitorear dicho lugar mediante los dispositivos instalados en dicho lugar. El presente proyecto de investigación propone el desarrollo de una aplicación web que permita monitorear el Centro de procesamiento mediante los dispositivos instalados, permitiendo así disponer de un control y monitoreo relevante haciendo que el tiempo de ejecución de las acciones a tomar en función al problema presentado sea eficiente y oportuno.

Mediante la investigación de campo se realizó el levantamiento de requerimientos respectivos a través de la observación y entrevista, cuya información ha sido plasmada en una ficha de observación para su respectivo análisis y procesamiento, lo cual ha permitido tener una idea clara de los requerimientos para el desarrollo de la aplicación web además se ha incluido la investigación bibliográfica-documental que fue de gran ayuda. Para el desarrollo del proyecto se ha seguido la Metodología en Cascada la cual permitió tener una mejor administración del proyecto cumpliendo cada fase una a la vez, hasta llegar al objetivo propuesto.

Se ha utilizado el Lenguaje Unificado de Modelado (UML), para capturar información de cómo un sistema debe trabajar, ya que es un estándar para la representación de

procesos o esquemas de software adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

Al realizar la implementación de la aplicación web los resultados obtenidos cumplieron los requerimientos de los beneficiarios obteniendo una aplicación web que permite monitorear el Centro de Procesamiento de Datos de manera real, eficiente y confiable mediante los sensores instalados en dicho lugar.

SUMMARY

The Cooperativa de Ahorro y Crédito Indígena "SAC Ltda." Ambato city, whose purpose is to serve the financial needs of its members and others.

Currently in the Data Processing Center whose area is one of the most important since there the resources needed to process information the organization has an alarm system with two temperature sensors Movement and focus, the same as are connected to a siren, fulfilling only deterrent activating a siren, it is not connected to a central station by that system is not efficient, reliable and safe, being able to cause inconvenience safety control as theft, damage and even inefficient management of information and goods at appropriate times by unauthorized personnel.

In the absence of a highly secure system does not have a web application that can monitor such place by the devices installed therein. This research project proposes the development of a web application that can monitor the processing center by the installed devices, allowing to have a control and relevant monitoring causing the runtime of the actions to take based on the presenting problem is efficient and timely.

Through field research lifting respective requirements was conducted through observation and interviews, whose information has been captured in a tab observation for examination and prosecution, which has allowed to have a clear idea of the requirements for the web application development has also included bibliographic and documentary research was of great help. To develop the project has followed the methodology Cascading which allowed a better project management meeting each stage one at a time, until the proposed objective.

We used the Unified Modeling Language (UML) to capture information about how a system should work, is a standard for representing processes or schemes software has been adopted internationally by numerous organizations and companies to create diagrams, diagrams and documentation relating to software development.

When deploying the web application the results met the requirements of the beneficiaries obtaining a web application that allows you to monitor the processing center real, efficiently and reliably through the sensors installed therein.

GLOSARIO DE TÉRMINOS

CPD: Centro de Procesamiento de Datos.

Raspberry Pi: Es un ordenador de placa reducida desarrollada con el objetivo de estimular la enseñanza de ciencias de la computación y desarrollo de proyectos.

Arduino: Es una plataforma de hardware libre y código abierto, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

DAQ: (Adquisición de datos), consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otras electrónicas (sistema digital).

GPRS: (Servicio de Radio transmisión de Paquetes Generales), es una solución para datos móviles que ofrece eficiencia espectral para nuevos y más veloces servicios de datos.

HTML: (Lenguaje de Marcación de Hipertexto), es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet.

HTTP: (Protocolo de Transferencia de Hipertexto), es el método más común de intercambio de información en la red informática mundial, el método mediante el cual se transfieren las páginas web a un ordenador.

BD: (Base de Datos), es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

SGBD: (Sistema de Gestión de Bases de Datos), es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos.

PHP: (Hypertext Preprocessor), lenguaje de programación adecuado para desarrollo web y que puede ser incrustado en HTML.

MVC: Modelo-Vista-Controlador, es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación.

Framework Web: Es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de aplicaciones web.

C.O.A.C: Cooperativa de Ahorro y Crédito.

Datasheet: Es un documento que resume el funcionamiento y otras características de un componente (por ejemplo, un componente electrónico) o subsistema (por ejemplo, una fuente de alimentación).

TTL: (Transistor-Transistor-Logic), es el tipo de circuito digital en donde la salida se deriva de dos transistores.

ARDUINO: Plataforma de hardware libre y código abierto, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

IDE: (Entorno interactivo de desarrollo), entorno de desarrollo integrado de Arduino, el lenguaje de programación Arduino se basa en C/C++.

Puerto COM: ("Comunicaciones"), es un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez.

Cubieboard: Es un mini ordenador utilizado para la enseñanza de ciencias de la computación y desarrollo de proyectos con una arquitectura de ARM.

Arquitectura ARM: Conjunto Reducido de Instrucciones de 32 bits ampliamente utilizado en ordenadores personales

UML: (Lenguaje Unificado de Modelado), es un lenguaje gráfico para especificar, visualizar, construir y documentar los sistemas de software.

GPL: (Licencia Pública General de GNU), usada en el mundo del software y garantiza a los usuarios finales la libertad de usar, estudiar, compartir y modificar el software.

CSS: (Hoja de Estilo en Cascada), es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

GUI: (Interfaz Gráfica de Usuario), conjunto de formas y métodos que posibilitan la interacción de un sistema informático con los usuarios, utilizando formas gráficas e imágenes.

INTRODUCCIÓN

Al informe final del proyecto nominado “Aplicación web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato” que se presenta a continuación, se le ha dividido en capítulos que pretenden facilitar la comprensión del contenido de este trabajo.

Capítulo I denominado “**PROBLEMA**”, se identifica el problema a investigar, la justificación respectiva de la investigación y el planteamiento de los objetivos a obtener tras la culminación del trabajo de investigación.

Capítulo II denominado “**MARCO TEÓRICO**”, comprende los antecedentes investigativos, fundamentación teórica que sirven de soporte a la investigación para luego establecer la propuesta de solución del problema.

Capítulo III denominado “**METODOLOGÍA**”, se describe las modalidades de investigación, se especifica la forma como se recolectará y procesará la información, por último se especifica las actividades a realizar en base a los objetivos específicos para el desarrollo del proyecto.

Capítulo IV denominado “**DESARROLLO DE LA PROPUESTA**”, en este capítulo se describe todo el desarrollo de la propuesta de solución para el desarrollo de la Aplicación Web.

Capítulo V denominado “**CONCLUSIONES Y RECOMENDACIONES**”, se estableciendo las conclusiones y recomendaciones pertinentes luego del desarrollo del proyecto.

Al final se incluye las referencias consultadas, glosario de términos, y se anexa los correspondientes documentos respecto al trabajo de investigación.

CAPÍTULO I

EL PROBLEMA

1.1 Tema

Aplicación Web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.

1.2 Planteamiento del problema

La inseguridad es un factor que ha crecido mucho en los últimos años en todas las sociedades, por lo que la seguridad de bienes e integridad de la información es fundamental en toda entidad.

A nivel mundial con el avance del tiempo la tecnología progresa, las empresas, instituciones se sienten en la necesidad de adquirir sistemas de alarma integrados a centros de monitoreo que se encargan e intervienen ante cualquier problema en la seguridad del personal, hardware, software e instalaciones en general.

Tomando en cuenta que en el mundo, en países industrializados y desarrollados existen sistemas de alarma muchos más avanzados integrados a servicios de monitoreo por internet quienes envían la información directamente a quienes deben recibirla, usuarios o titulares de los servicios, facilitando así el monitoreo desde cualquier parte del mundo.

A nivel Nacional las empresas, instituciones, hogares y centros comerciales han visto en la necesidad de implementar en sus instalaciones el monitoreo mediante sistemas de

alarma que utilizan dispositivos como sensores y cámaras los mismos que ha dado excelente resultado en cuanto a seguridad se refiere.

En la actualidad resulta necesario y hasta cierto punto fundamental, que toda institución, organización, entidad pública o privada, cuente con sistemas de alarma automatizados que permitan el control y monitoreo, de manera oportuna y eficaz respecto a la seguridad física, es así que las instituciones que carecen de tecnología que permitan disponer de un control y monitoreo relevante en momentos precisos y oportunos, a menudo se enfrenta a problemas de distinta índole como la inseguridad de la información, bienes, recursos humanos entre otros.

A nivel provincial exclusivamente en Tungurahua, la mayoría de entidades han venido haciendo uso de la tecnología implementando sistemas de alarma para cuidar sus recursos humanos, bienes e integridad de su información los mismos que no están controlados y monitoreados en su totalidad, por lo tanto están propensas a afrontar problemas como el robo, daños, mal uso o manipulación incorrecta las cuales son importantes para el crecimiento y supervivencia de dichas organizaciones.

Este es el caso de la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.”, actualmente cuenta con un sistema de alarma con dos dispositivos (sensores) antirrobo (Sensor de Movimiento) y contra incendio (Sensor de Temperatura y humedad) los mismos que están conectados a un solo dispositivo (Sirena), cumpliendo así solo la función disuasoria activando una sirena, el mismo que no está conectado a una Central de Monitoreo haciendo que dicho sistema no sea eficiente, confiable y seguro.

Al no disponer de un sistema altamente seguro no se dispone de una aplicación web que permita monitorear dichos dispositivos que están instalados en áreas precisas no autorizadas e importantes como es el caso del Centro de Procesamiento de Datos (CPD), cuya área es una de las más importantes ya que ahí se concentran los recursos necesarios para el procesamiento de la información de la organización, pudiéndose provocar inconvenientes de control de seguridad como robos, daños e incluso una ineficiente gestión de información y bienes en momentos oportunos.

La inexistencia de una aplicación web de monitoreo que ayude a detectar desastres en el tiempo más corto posible, hace que el tiempo de ejecución de las acciones a tomar en función del problema presentado sean mayores.

Otra situación es la carencia de dispositivos de detección (sensores) en áreas vulnerables en el CPD, la entidad al no contar con estos dispositivos esta propensa a que en el CPD no esté totalmente seguro, limitando así la toma de decisiones en caso de algún desastre o problemas de riesgos detectados llegando a ocasionar inconvenientes de alto índole y de riesgo para la Cooperativa.

La Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” al no disponer de recursos económicos suficientes ha conllevado a que los directivos no pongan énfasis en la adquisición de una aplicación web que monitoree en tiempo real el Centro de Procesamiento de Datos de los dispositivos (sensores) instalados, exponiendo así en riesgo y peligros la seguridad e integridad de una de sus áreas más importantes para el procesamiento de la información de la entidad financiera.

1.3 Delimitación del problema

De Contenidos:

Área académica: Software.

Línea de investigación: Desarrollo de software.

Sublínea: Aplicación Web.

Espacial:

El presente trabajo investigativo se realizará en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.”, ubicada en la Ciudad de Ambato, calle Castillo 7-27 entre Juan B. Vela y 12 de Noviembre.

Temporal:

El presente proyecto de investigación tendrá una duración de 6 meses, a partir de la aprobación por el Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

1.4 Justificación

Actualmente la tecnología en lo que respecta la seguridad está ocupando una amplia área en el sector industrial, empresarial, institucional entre otros, donde la tecnología avanza día a día con una rapidez considerable en lo que respecta a la seguridad.

La realización de este trabajo de investigación es de gran importancia porque se puede vincular la teoría con la práctica, ya que gracias a esto se puede complementar el desarrollo de este proyecto y los resultados serán de gran ayuda en el desarrollo tecnológico logrando que la cooperativa cuente con una buena aplicación web que permita monitorear de manera exacta y oportuna el CPD mediante los distintos dispositivos (sensores), logrando así la seguridad e integridad del personal, datos, hardware, software e instalaciones en general.

Con este proyecto se beneficiará principalmente la entidad, ya que al contar con un sistema de alarma eficiente ayudará a detectar desastres en el tiempo más corto posible velando así la seguridad e integridad del CPD, luego quienes manejan la aplicación web de monitoreo, ya que dispondrán de una información exacta y oportuna en tiempo real, reduciendo el tiempo de ejecución de las acciones a tomar en función del problema presentado.

1.5 Objetivos

1.5.1 Objetivo General

Desarrollar una Aplicación Web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.

1.5.2 Objetivos Específicos

- Analizar los procesos de seguridad utilizados en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.
- Establecer los requerimientos necesarios para el desarrollo de la aplicación web de monitoreo del Centro de Procesamiento de Datos en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.
- Diseñar una aplicación web de monitoreo que mejore la seguridad e integridad del CPD de la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.
- Implementar una aplicación web de monitoreo del Centro de Procesamiento de Datos mediante sensores de seguridad en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes investigativos

Una vez revisado los archivos en la biblioteca de la facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, se ha determinado que existen trabajos con variables afines a este estudio que a continuación se describen.

“Aplicación Web para la distribución de espacios disponibles de parqueo en la Universidad Técnica de Ambato campus Huachi Chico, el proyecto permite al usuario encargado del parqueadero tener toda la información en la aplicación web, sobre los lugares libres donde pueden estacionarse los vehículos que ingresen a la universidad, todo esto gracias a los indicadores electrónicos utilizados para la obtención y presentación de la ubicación de sitios de estacionamientos libres en la aplicación web [1].”

“El presente trabajo investigativo y de implementación se realiza ante la necesidad de suplir las necesidades existentes en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato (FISEI - UTA) en el área de laboratorios, para lo cual se realiza el estudio e implementación de un sistema centralizado empleando una tarjeta ARDUINO, cámaras IP y elementos de control; para accionar las cerraduras de las puertas de los laboratorios abriéndolas según se necesite, teniendo como auxiliar una interfaz visual en la computadora para poder monitorear la actividad que se desarrolla en los laboratorios [2].”

En lo que respecta a proyectos con arduino desde fuentes bibliográficas de Internet se ha encontrado los siguientes trabajos de tesis:

“El proyecto trata de un diseño y construcción de un sistema de control, monitoreo y vigilancia por cámara, vía internet, para ser utilizado ya sea en un domicilio u departamento, funcionan las diferentes ventajas que nos facilita la tecnología del Internet. El sistema tiene instalado un software libre, dispositivos controladores que son electroválvula y motor, una mini computadora llamada Raspberry Pi, una tarjeta de adquisición de datos llamado Arduino Uno, sensores y dos cámaras web [3].”

“Esta tesis está ejecutada en el Jardín Botánico de la Universidad Técnica de Manabí de la ciudad de Portoviejo, con el propósito de realizar un control, registro e investigación de las especies en peligro de extinción mediante un monitoreo por cámaras hacia las jaulas o espacios físicos donde radican o se desplazan dichas especies hospedadas en este jardín, específicamente el área de rescate [4].”

2.2 Fundamentación teórica

Monitoreo

“El monitoreo consiste en la observación del curso de uno o más parámetros para detectar eventuales anomalías [5].”

Para el investigador el monitoreo en el ámbito de la seguridad, puede realizarse efectivamente a través de un monitor mediante cámaras, dispositivos electrónicos como detectores de humo, movimientos, sensores etc., mediante el trabajo de algún vigilante.

Centro de Procesamiento de Datos

“Es el conjunto de recursos físico, lógicos, y humanos necesarios para la organización, realización y control de las actividades informáticas de una empresa [6].”

Adquisición de Datos

“La adquisición de datos o adquisición de señales, consiste en la toma de muestras del mundo real (sistema analógico) para generar datos que puedan ser manipulados por un ordenador u otras electrónicas (sistema digital) [7].”

“La adquisición de datos (DAQ) es el proceso de medir con una PC un fenómeno eléctrico o físico como voltaje, corriente, temperatura, presión o sonido [8].”

Sistema de Adquisición de Datos

Un sistema DAQ consiste de sensores, hardware de medidas DAQ y una PC con software programable [8]

Partes de un Sistema DAQ



Fig. 1: Partes de un Sistema DAQ

Elaborado por: Diego Poaquiza – Investigador

Sensor

“Un sensor, también llamado un transductor, convierte un fenómeno físico en una señal eléctrica que se puede medir. Dependiendo del tipo de sensor, su salida eléctrica puede ser un voltaje, corriente, resistencia u otro atributo eléctrico que varía con el tiempo. Algunos sensores pueden requerir componentes adicionales y circuitos para producir correctamente una señal que puede ser leída con precisión y con toda seguridad por un dispositivo DAQ [8].”

Un sensor es un dispositivo capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Las variables de instrumentación pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc.

Seguridad

La seguridad, consiste el concepto de seguridad en la protección de las personas y de su entorno mediante elementos como circuitos telefónicos vigilados, cámaras para vigilancia de accesos, cerraduras de alta seguridad, cristales y puertas blindadas, emisoras de radio comunicadas con personal de seguridad y otros sistemas [9].

Seguridad Física

La seguridad física requiere que los equipos y la infraestructura sea salvaguardada de forma que minimice el riesgo de que el recurso sea robado, destruido o modificado, por ejemplo limitando el acceso a las áreas en donde se mantiene la infraestructura de red.

“La parte de seguridad física incluye contemplar medidas para controlar y detectar riesgos como incendios, inundaciones o fallas de control de temperatura que pueda afectar los sistemas de cómputo [10].”

Sistema de Alarma

“Un sistema de alarma con monitoreo consiste en la instalación de una serie de equipos electrónicos conectados hacia la Central de Monitoreo en los lugares de su hogar o empresa considerados estratégicos desde el punto de vista de la seguridad. Estos dispositivos pueden ser sensores de movimiento, contactos magnéticos, botón de pánico, entre otros, y éstos envían señales en forma periódica a nuestra Central de Monitoreo [11].”

Un sistema de alarma es un elemento de seguridad pasiva. Esto significa que no evitan una situación anormal, pero sí son capaces de advertir de ella, cumpliendo así, una función disuasoria frente a posibles problemas.

Según el criterio del investigador un sistema de alarma es el conjunto de dispositivos electrónicos ubicados en sitios estratégicos que están conectados a una Central de Monitoreo para su control ante problemas causados por agentes externos reduciendo así la ejecución de acciones a tomar frente a los problemas presentados.

Central de Monitoreo

“Una Central de monitoreo es un centro remoto a donde se reportan todos los detectores instalados en su propiedad, informando mediante señales enviadas por la línea telefónica o por GPRS (Internet inalámbrico), cada uno de los eventos que se den en la propiedad desde alarma, falla en el sistema, activación y desactivación del sistema de alarma por el usuario o su personal, etc. [12].”

Aplicación Web

“En la ingeniería software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web, y en la que se confía la ejecución de la aplicación al navegador [13].”

Servidor Web

“Un servidor web es un programa que sirve datos en forma de Páginas Web, hipertextos o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo Http. Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que solemos conocer como un Navegador Web. A modo de ejemplo: al teclear (<http://www.cnice.mec.es>) en un navegador, éste realizará una petición HTTP al servidor que tiene asociada dicha URL [14].”

Navegador Web

Para establecer conexiones con los servidores Web, y obtener la información y los servicios que estos prestan, el usuario necesita tener instalado en su equipo un programa cliente capaz de comunicarse con ellos. Estos programas son los llamados Navegadores Web. Los Navegadores Web, también llamados Visores de Web o Browsers, son aplicaciones que permiten ver en pantalla texto con formato (con palabras en negrita, y con distintas fuentes tipográficas, tamaños y colores) y presentar imágenes en línea. También permiten visualizar secuencias de vídeo y escuchar ficheros de sonido [14].

Base de Datos

“Una base de datos (BD) es la representación de una colección de datos estructurada que describe las actividades de una organización. Esta representación incluye entidades del mundo real y sus interrelaciones y tiene que permitir diversas utilidades. Un sistema de gestión de bases de datos (SGBD) es un software específicamente diseñado y desarrollado para asistir en la creación, la manipulación y el mantenimiento de las BD [15].”

Lenguaje de Programación

“Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora [16].”

Lenguaje de programación para Aplicaciones Web:

“Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámicos, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos [17].”

Existen diferentes lenguajes de programación para la web.

- Lenguaje HTML.
- Lenguaje Javascript.
- Lenguaje PHP.
- Lenguaje ASP.
- Lenguaje ASP.NET.
- Lenguaje JSP.
- Lenguaje Python.

- Lenguaje Ruby.
- Frameworks Web.

Php

PHP es un lenguaje interpretado del lado del servidor que se caracteriza por su potencia, versatilidad, robustez y modularidad.

Los programas escritos en PHP son embebidos directamente en el código HTML y ejecutados por el servidor web a través de un intérprete antes de transferir al cliente en forma de código HTML.

Al ser uno lenguaje open source, tanto el intérprete como su código fuente son totalmente gratuitos [18].

Mysql

“Es un sistema de administración de base de datos relacionales rápidos, sólidos y flexibles. Es ideal para crear base de datos con acceso desde páginas web dinámicas, para la creación de sistemas de transacciones En línea o para otra solución profesional que implique almacenar datos, teniendo la posibilidad de realizar múltiples y rápidas consultas” [18] .

Jquery

“JQuery es un framework de JavaScript para facilitar, entre otros, el acceso a los elementos del DOM, los efectos, interactuar con los documentos HTML, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web” [19].

JavaScript

“JavaScript es un lenguaje interpretado en el cliente por el navegador al momento de cargarse la página, es multiplataforma, orientado a eventos con manejo de objetos, cuyo código se incluye directamente en el mismo documento HTML” [20].

Ajax

“Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications).

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones” [21] .

Css

“CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla, por impresora, por voz (cuando la información es pronunciada a través de un dispositivo de lectura) o en dispositivos táctiles basados en Braille” [22].

Bootstrap

Bootstrap es un framework front-end que simplifica el proceso de creación de diseños web combinando CSS y JavaScript, desarrollado por Twitter y a disposición pública [23].

Para el investigador la mayor ventaja de usar este Framework es que podemos crear interfaces que se adapten a los distintos navegadores (responsive design) apoyándonos en un framework potente con numerosos componentes webs que nos ahorrarán mucho esfuerzo y tiempo.

Font Awesome

Font Awesome es un proyecto open source, es una fuente de iconos gratis que puedes integrar en tus aplicaciones web utilizando Bootstrap (librería Javascript/CSS3 gratis),

la aplicación de esta fuente de iconos es muy sencilla y permite insertar los iconos de forma personalizada en diferentes formatos [24].

La fuente de iconos mediante la que se pueden insertar algunos iconos sociales y símbolos en lugar de imágenes, hace que la página en la que estamos trabajando sea más ligera.

Json

“JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo” [25].

Framework

“Un framework de aplicaciones web es un tipo de framework que permite el desarrollo de sitios web dinámicos, web services (servicios web) y aplicaciones web. El propósito de este tipo de framework es permitir a los desarrolladores construir aplicaciones web y centrarse en los aspectos interesantes, aliviando la típica tarea repetitiva asociada con patrones comunes de desarrollo web” [26].

Los frameworks de aplicación web además promueven la reutilización y conectividad de los componentes, así como la reutilización de código, y la implementación de bibliotecas para el acceso a base de datos.

CakePHP

Es un framework para PHP, libre, de código abierto que facilita el desarrollo de aplicaciones web, utilizando el patrón de diseño MVC (Modelo-Vista-Controlador), está desarrollada en php con programación orientada en objetos [26].

MVC

“El Modelo-Vista-Controlador es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el modulo encargado de gestionar los eventos y las comunicaciones [26]”.

Python

“Es un lenguaje de programación de alto nivel, interpretado y multipropósito, es utilizado en diversas plataformas Windows, Mac Os X y Linux, puede funcionar en smartphones y soporta orientación a objetos [27]”.

Python es utilizado para desarrollar aplicaciones científicas, comunicaciones de red, aplicaciones de escritorio con interfaz gráfica de usuario (GUI), crear juegos para smartphones.

Sensor de Temperatura y Humedad DHT11

Este es el encargado de medir y controlar la temperatura puede medir las condiciones ambientales (temperatura ambiente, humedad relativa).

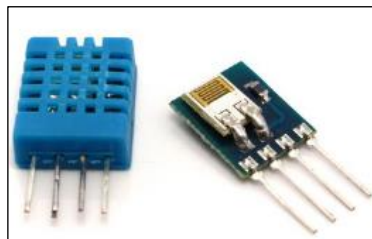


Fig. 2: Sensor de Temperatura y Humedad DHT11

Elaborado por: <http://www.exp-tech.de/service/datasheet/DHT11.PDF>

Características:

- Humedad relativa y temperatura.
- Señal digital calibrada.
- Excepcional estabilidad a largo plazo.
- Componentes adicionales no necesitan.
- Distancia de transmisión larga.
- Bajo consumo de energía.

Descripción:

DHT11

Salida señal digital calibrada. Utiliza la tecnología exclusiva de señal digital de recolección-técnica y con sensor de humedad, asegurando su fiabilidad y estabilidad.

Pequeño tamaño, bajo consumo y larga distancia de transmisión (20m) [28].

Especificación Técnica:

Fuente de alimentación	3-5,5V
La medición de humedad y Temperatura.	20 - 90% de humedad relativa; Temperatura 0-50 Celsius.
Precisión	Humedad + - 4% de humedad relativa (Máx +-5% RH); temperatura + -2.0 Celsius
Señal de Salida	Señal digital a través de un solo bus
Periodo de detección	2 segundos
Dimensiones	tamaño 12 * 15.5 * 5.5mm

Tabla 1: Especificación Técnica Sensor DHT11

Elaborado por: Diego Poaquiza - Investigador

Se puede decir que el dispositivo Dht11 emite señales digitales, en base a la información recolectada y la revisión de las características del datasheet correspondiente.

Señales Digitales:

“Son variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 ó 1, V o F, etc.” [29].

Niveles de voltajes de señales digitales

Los niveles específicos dependen del tipo de dispositivos utilizado. Por ejemplo si se emplean componentes de la familia lógica TTL (transistor-transistor-logic) los niveles son 0 V y 5 V, aunque cualquier valor por debajo de 0,8 V es correctamente interpretado como un 0 y cualquier valor por encima de 2 V es interpretado como un 1, este es el caso de nuestros dispositivos que al ser digitales sus niveles de voltajes oscilaran de 0 a 5 voltios respectivamente.

Sensor de Movimiento PIR DYP-ME003

Este sensor de movimiento es un dispositivo electrónico equipado de sensores que responden un movimiento físico.



Fig. 3: Sensor de Movimiento PIR DYP-ME003

Elaborado por: <http://www.electfreaks.com/store/pir-motion-sensor-moduledypme003-sen005-p-156.html>

Características:

Control fotosensible.	Ajuste del control fotosensible durante el día o la intensidad de la luz.
Compensación de temperatura	Cuando la temperatura ambiente se eleva a 30 ~ 32 ° C
Voltaje de entrada	DC 4.5-20V
Ángulo de monitoreo	<110 grados
Distancia de monitoreo	3 m (predeterminado) - máx. 7 m
Tamaño de la lente	Diámetro: 23 mm (por defecto)
Señal	Digital
Dimensiones	32mm x 24mm

Tabla 2: Características Sensor PIR DYP-ME003

Elaborado por: <http://elecfrakes.com/store/download/datasheet/sensor/DYP-ME003/Specification.pdf>

De igual manera en base a la información recolectada de dicho dispositivo se puede decir que el tipo de señal que emite es digital.

Contacto Magnético modelo 943

Este sensor detecta los campos magnéticos que provocan los imanes metidas en el interior de la cápsula del contacto, puede detectar la proximidad sin contacto directo.

Este tipo de dispositivo es digital, es comúnmente utilizado en los bordes de puertas, ventanas, etc.



Fig. 4: Contacto Magnético modelo 943

Elaborado por: <http://www.security.honeywell.com/es/productos/intrusion/se/cm/ca/82026.html>

Características:

- Contacto magnético montaje en superficie “Mini”
- Potencia nominal de 12 mm cable de 1,5 m incorporado
- Montaje en superficie atornillado o pegado (autoadhesivo)
- Dimensiones: 17,35 x 26,16 x 6,35 mm
- 943WH: color blanco

ARDUINO UNO

Arduino es una plataforma de hardware libre y código abierto, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Arduino se puede utilizar para el desarrollo de objetos interactivos, puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores, sensores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing) que descienden del C/C++ [30].

Al ser open-hardware, tanto su diseño como su distribución es libre es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

Cubieboard A20

Es un mini ordenador con una arquitectura de ARM como los de muchos móviles "inteligentes".

La Cubieboard A20 es un computador de tamaño reducido, que se conecta a un televisor o monitor, a un teclado y a un mouse. Puede ser utilizado para muchas de las tareas que tu PC de escritorio hace, como manejo de hojas de cálculo, edición de textos y juegos. También reproduce vídeo de alta definición.

Cuenta con la posibilidad adicional de conectar un disco duro (SATA 2TB), entre otras conexiones y puede operarse bajo cualquiera de los siguientes OS: Android 4.2 Jelly Bean, Ubuntu 12.04 y otras distribuciones de Linux como Lubuntu, Fedora, Debian, Cubian, Buildroot [31].

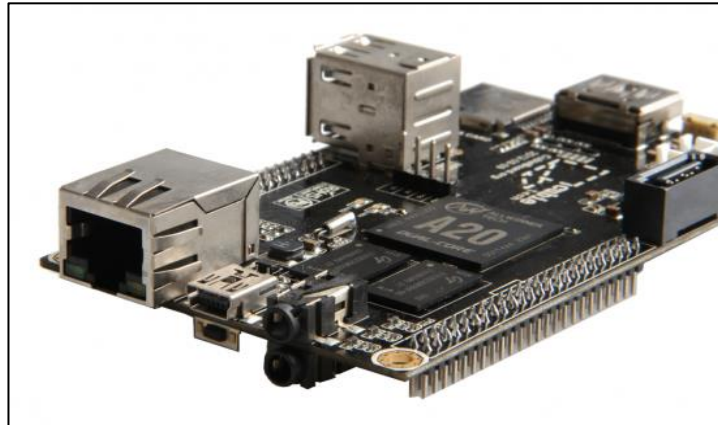


Fig. 5: Cubieboard A20

Elaborado por: <http://docs.cubieboard.org/products/start#a20-cubieboard>

Especificaciones Hardware

- *AllWinnerTech SOC A20, ARM® Cortex™-A7 Dual-Core ARM® Mali400 MP2 Complies with OpenGL ES 2.0/1.1*
- *1GB DDR3 @480M*
- *3.4GB internal NAND flash, up to 32GB on SD slot, up to 2T on 2.5 SATA disk*
- *5VDC input 2A or USB otg input*
- *1x 10/100 ethernet, support usb wifi*
- *2x USB 2.0 HOST, 1x mini USB 2.0 OTG, 1x micro sd*
- *1x HDMI 1080P display output*

- *1x IR, 1x line in, 1x line out*
- *96 extend pin interface, including I2C, SPI, RGB/LVDS, CSI/TS, FM-IN, ADC, CVBS, VGA, SPDIF-OUT, R-TP, and more [31].*

Metodología Modelo en Cascada

La metodología de desarrollo de software denominado Modelo de Cascada o Ciclo de Vida Clásico, consiste en descomponer la actividad global en fases que se van desarrollando en forma lineal, es decir una detrás de la otra de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior; esta metodología permite tener una mejor administración y llegar al objetivo propuesto [32].

Ventajas de la Metodología Modelo de Cascada:

- Este proceso conduce a entregar el proyecto a tiempo.
- Facilita la gestión de proyectos.
- Planificación sencilla.
- Las fases son completadas una a la vez, continúan hasta que los objetivos de la anterior se hayan cumplido.
- Cada fase produce específicos resultados y revisión del proceso.

Estructura de la metodología en cascada

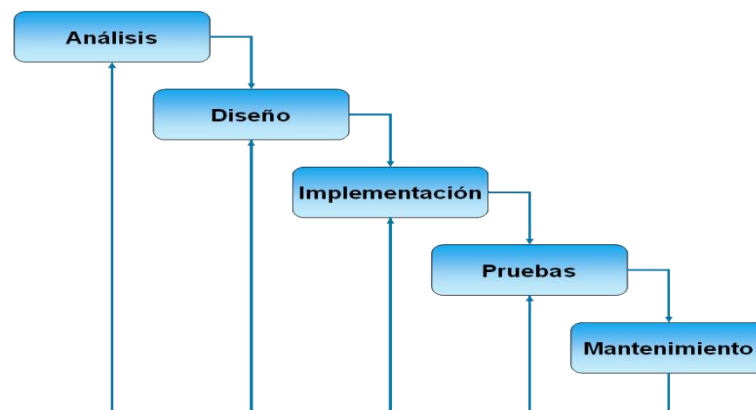


Fig. 6: Estructura de la metodología en cascada

Elaborado por: http://www.ecured.cu/index.php/Metodología_de_cascada

2.3 Propuesta de solución

Desarrollar una aplicación web que permita un monitoreo eficiente del Centro de Procesamiento de datos mediante sensores de seguridad reduciendo así la inseguridad existente.

CAPÍTULO III

METODOLOGÍA

3.1 Modalidad de la investigación

El presente proyecto es de investigación aplicada, ya que permitirá plantear una propuesta de solución que mejore la seguridad e integridad del CPD de la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.

En el desarrollo del proyecto se emplea la modalidad de investigación de campo, ya que se tendrá contacto directo con el escenario institucional en el que se producen los acontecimientos con el fin de recoger información a través de la entrevista y la observación, información que será necesaria para su análisis e interpretación.

Se utilizará la investigación bibliográfica-documental ya que se recurrirá al Internet, tesis, diversos libros y artículos, obteniendo la referencia teórica necesaria para el sustento teórico-científico del proyecto.

3.2 Población y muestra

Por la característica del proyecto de investigación no se requiere la población y muestra.

3.3 Recolección de información

Las técnicas que se emplearán para la recolección de la información en el proceso de investigación serán como fuente primaria la observación que irá acompañada de entrevistas y como secundaria libros, tesis, artículos etc.

La observación así como las entrevistas serán de gran valor en la apreciación de la realidad, circunstancias que permitirán confrontar los hechos de forma directa al problema dando así transparencia e imparcialidad a la investigación, se utilizará como instrumento el registro de datos para la recolección de información.

La investigación además se sustentará en la recolección de información secundaria a través de estudios realizados referentes al tema como tesis, libros, documentos En línea, artículos científicos, que aporten información importante para la investigación y cumplimiento de los objetivos planteados en el proyecto de investigación.

3.4 Procesamiento y análisis de datos

Para la recolección, procesamiento y análisis de la información se aplicará los siguientes procedimientos:

1. Elaboración del instrumento de recolección de información.
2. Ajuste al instrumento de recolección de información.
3. Aplicación del instrumento de recolección de información.
4. Organización de la información.
5. Análisis e interpretación de la información.

Se procederá a la elaboración del instrumento de recolección de información, una vez recolectada la información se procederá al respectivo análisis de los datos obtenidos los mismos que serán procesados.

3.5 Desarrollo del proyecto

Para cumplir el desarrollo de este proyecto de investigación, se realizará en forma secuencial las siguientes actividades en base a la Metodología en Cascada a utilizar:

1. Análisis.

1.1. Realizar una ficha de observación que permita reunir la información sobre los procesos de seguridad utilizados dentro del CPD.

1.2. Analizar y procesar la información sobre los procesos de seguridad utilizados.

1.3. Análisis y requerimientos de la Aplicación Web.

1.4. Captura de los datos que emiten los sensores.

1.5. Definición requerimientos software y hardware.

1.5.1. Seleccionar un Framework Open Source necesario de desarrollo de aplicaciones web basado en la arquitectura (MVC: Modelo-Vista-Controlador).

1.6. Diagramas UML.

1.6.1. Diagramas de Caso de Uso.

1.6.2. Diagrama de Secuencia.

2. Diseño.

2.1. Diseño de la Base de Datos.

2.1.1. Diccionario de Datos.

2.2. Diseño de la interfaz previa de la aplicación web.

3. Implementación.

3.1. Implementación Hardware y Software del sistema de alarma, obtención, análisis y tratamiento de la información emitida por los dispositivos de seguridad.

3.2. Implementación Base de datos, Interfaz gráfica de usuario y código fuente de la Aplicación.

4. Pruebas.

4.1. Pruebas de Caja Blanca.

4.2. Pruebas de Caja Negra.

5. Mantenimiento.

CAPÍTULO IV

DESARROLLO DE LA PROPUESTA

4 Metodología

Para el desarrollo de la propuesta, se tomó los objetivos específicos planteados en el proyecto, detallando a continuación cada actividad y su desarrollo incluido en la Metodología en Cascada a utilizar que nos permitirá llevar una mejor administración y control de cada etapa a desarrollarse, que consisten en el Análisis, Diseño, Implementación y las Pruebas, las cuales son completadas una a la vez de forma secuencial continuado así hasta que los objetivos de la anterior se hayan cumplido hasta alcanzar el final del ciclo de vida de la aplicación cumpliendo así el objetivo principal.

4.1 Análisis

En esta etapa se realiza el proceso de recopilación de los requisitos de todos los elementos del sistema como las necesidades de funcionalidad y servicio con las cuales debe contar así como el análisis de los requisitos de los usuarios finales del software para determinar qué objetivos debe cubrir sin entrar en detalles internos.

4.1.1 Análisis de los procesos de seguridad física utilizados en la Cooperativa de Ahorro y Crédito Indígena “SAC Ltda.” de la ciudad de Ambato.

Para la recolección de la información se utilizó como técnica, la observación y como instrumento una guía de observación además de la entrevista, lo cual permitió reunir la suficiente información sobre los procesos de seguridad utilizados en el

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Centro de Procesamientos de Datos de la “C.O.A.C” “SAC Ltda.” de la ciudad de Ambato.

GUÍA DE OBSERVACIÓN
Objetivo: Recolectar información sobre los procesos de seguridad física utilizados en el Centro de Procesamientos de Datos.
Tipo de Observación: Observación directa.
INDICADORES A OBSERVAR
➤ Control de Acceso Físico.
<ul style="list-style-type: none"> • Guardia de seguridad abre la puerta para la entrada del personal. • No existe control del personal que ingresa.
➤ Dispositivos de Seguridad Física.
<ul style="list-style-type: none"> • Sistema de Alarma (Control de movimiento, temperatura y humedad conectados a una sirena). • Carencia de dispositivos de seguridad. • Ausencia de seguridad perimetral.
➤ Encendido del sistema de Alarma.
<ul style="list-style-type: none"> • De manera manual a través de la conexión de la alarma a la fuente de alimentación de energía. <p>Procesos de encendido:</p> <p>Se enciende el sistema de alarma en los siguientes días y horas:</p> <ul style="list-style-type: none"> • A partir de las 6pm del día Lunes hasta las 8am del día Martes. • Martes desde las 6pm hasta las 8am del día Miércoles. • Miércoles desde las 6pm hasta las 8am del día Jueves. • Jueves desde las 6pm hasta las 8am del día Viernes. • Viernes desde las 6pm hasta las 8am día Sábado. • A partir de la 13:30 pm del día Sábado hasta las 8am del día Lunes. • Se mantiene encendido el sistema de alarma todo el día los feriados.
➤ Apagado del sistema de Alarma.
<ul style="list-style-type: none"> • De manera manual desconectando el sistema de la alarma de la fuente de alimentación de energía. <p>Procesos de apagado:</p>

<p>Se apaga el sistema de alarma en los siguientes días y horas:</p> <ul style="list-style-type: none"> • A partir de las 8am del día Lunes hasta las 6pm del mismo día. • Martes desde las 8am hasta las 6pm del mismo día. • Miércoles desde las 8am hasta las 6pm del mismo día. • Jueves desde las 8am hasta las 6pm del mismo día. • Viernes desde las 8am hasta las 6pm del mismo día. • A partir de las 8am del día Sábado hasta las 13:30 pm del día Lunes.
<p>➤ Activación de la Sirena del Sistema de Alarma.</p>
<p>La sirena emitirá un sonido en las siguientes circunstancias siempre y cuando el sistema de alarma este encendido:</p> <ul style="list-style-type: none"> • Al detectar un movimiento. • Cuando la temperatura sobrepasa los 50° Centígrados y disminuye sobre bajo de los 15°.

Tabla 3: Guía de Observación, Seguridad Física CPD
Elaborado por: Diego Poaquiza – Investigador

4.1.2 Análisis y Procesamiento de la información recolectada de los procesos de seguridad utilizados en el CPD

De los resultados obtenidos mediante la observación y entrevistas que se aplicó al Jefe de Sistemas encargado del CPD de la Cooperativa “SAC Ltda.”, se determinó que se debe tener un control más eficiente del acceso del personal que ingresa.

En lo que se refiere a la seguridad física implementado en el CPD, se ha podido constatar un sistema de alarma con dispositivos conectados a una sirena los mismos que no están conectados a una Central de Monitoreo para su respectivo control y monitoreo, además se pudo constatar la carencia de dispositivos de detección y alarma.

Otro detalle que se observó es la ausencia de seguridad perimetral o seguridad perimetral insuficiente, en este punto se observó puertas y ventanas abiertas, determinando a las mismas como áreas vulnerables del CPD, llegando a ser un punto de inseguridad de alto riesgo de seguridad.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Por otra parte se realizó un análisis del sistema de alarma determinando los procesos en lo que se respecta al encendido, apagado del sistema, así como también la activación de la alarma, cuya información se encuentra en la ficha de observación descrita anteriormente.

Luego de reuniones mantenidas entre las autoridades superiores se ha llegado a la conclusión de que se requiere una aplicación web que permita un monitoreo del CPD mediante los dispositivos instalados en el lugar, además para un mayor control se decidió adicionar al sistema de alarma un sensor llamado **Contacto Magnético modelo 943**, cuyo sensor se instalará en la ventana y puerta de acceso al CDP, ampliando así la seguridad cuyo objetivo es salvaguardar los equipos informáticos o de comunicaciones implicadas, así como servidores de bases de datos que puedan contener información crítica.

4.1.3 Análisis y requerimientos de la Aplicación web

En esta etapa de análisis se realizó reuniones entre el Gerente y el Jefe de Sistemas de la Cooperativa para establecer los requerimientos de la aplicación web y evaluar alternativas de solución.

Entre los requerimientos solicitados por los beneficiarios esta la seguridad de la información, restableciendo restricciones de usuarios y delegando permisos a cada uno de ellos.

Otro punto importante a tener en cuenta es la parte de los eventos suscitados en base a la información emitida por los sensores, cuya información deberá quedar registrada en tablas lo cual permitirá verificar los eventos suscitados con su fecha respectiva, además de almacenar la información personal de los usuarios que ingresan al sistema con el objetivo de tener un mayor control y así poder llevar un registro ante cualquier suceso.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Además la aplicación deberá contar con un control el cual permita el encendido y apagado del sistema de alarma teniendo en cuenta que dicho encendido o pagado estará basado dependiendo del caso, ya sea en la visualización o no de la información en tiempo real arrojada por los sensores.

La Aplicación web a desarrollarse se deberá realizar en Software Libre ya que dicha cooperativa no está actualmente en posibilidades de adquirir licencia de la algún software de desarrollo propietario por el momento, por lo cual se ha escogido como mejor opción el lenguaje PHP para la parte de programación y para la parte de almacenamiento de datos un Sistema Gestor de Bases de Datos denominado MySQL.

Luego de haber analizado los requerimientos que solicita la entidad se ha determinado que el sistema deberá contar con:

Un módulo para el ingreso, modificación, eliminación y control de: usuarios, dispositivos, tipos de dispositivos, ubicaciones y zonificación.

Un módulo de seguridad que permita el acceso a información importante solo a personas autorizadas.

Un módulo de monitoreo donde muestre de manera comprensible al usuario del sistema, sobre la información en tiempo real emitida por los sensores ante posibles acontecimientos presentados el CPD, además de un módulo que permita el encendido o pagado del sistema de alarma.

Generación de reportes de eventos suscitados sobre la información emitida por los sensores con la fecha además de los eventos de los usuarios como es el ingreso al sistema en la que ingresó con su respectiva fecha, mostrando dicha información de manera clara y entendible por el usuario.

Además, el sistema deberá contar con interfaces sencillas y amigables para su manipulación.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Fácil manejo, de tal forma que el usuario pueda navegar entre las diferentes páginas de una manera confiable.

El proceso y visualización de información deberá ser rápida, confiable y segura.

La presentación de mensajes de error deberán ser lo más específicas posible para una mejor interpretación.

Mantener una combinación de colores en la interfaz de acuerdo a la imagen de la Cooperativa teniendo en cuenta además colores que ayuden a una mejor visualización al usuario.

4.1.4 Definición de los requerimientos software y hardware para el desarrollo de la aplicación web.

Requerimientos Software

Para el desarrollo de la aplicación web se decidió utilizar las siguientes herramientas.

Lenguaje de Programación Web

Se decidió utilizar PHP tomando en cuenta, el tipo de licencia open source, multiplataforma, rápido, eficiente, seguro y la fácil adaptación en cualquier sitio web.

Selección de un Framework php Open Source basado en la arquitectura (MVC: Modelo-Vista-Controlador).

Arquitectura de Programación

La arquitectura de programación a utilizar será MVC (Modelo-Vista-Controlador) por lo que se decidió usar un Framework Open Source como plataforma de desarrollo de la aplicación web.

Al existir un sin número de frameworks php open source es difícil elegir cuál es el mejor de todos, habría que conocer profundamente cada una de ellas haciendo uso y

pruebas de la misma, por lo que a la hora de decidir cuál utilizar debe influenciar el conocimiento profundos de las características que proporciona cada una, funcionalidad, necesidades así como la experiencia en su uso.

Para la selección del Framework php se decidió realizar un cuadro de comparación sobre algunos frameworks php open source en base a características similares, requerimientos hardware y software que la Cooperativa SAC cuenta actualmente para la realización de la aplicación así como el conocimiento y experiencia sobre alguno de ellos.

Entre las características de comparación se mencionan las siguientes.

MVC: Indica si el Framework viene con soporte implícito para configurar el Modelo-Vista-Controlador.

Multiple DB's: Indica si el framework soporte múltiples bases de datos sin tener que cambiar nada.

OMR: Indica si el Framework soporta el mapeo de registros-objetos, habitualmente es una implementación del ActiveRecord.

DB Objects: Indica si el Framework incluye objetos de base de datos como TableGateway.

Template: Indica si el Framework incluye un sistema de cacheo de objetos o parecido.

Validation: Indica si el Framework viene construido con validación o filtro de componentes.

Ajax: Indica si el Framework viene construido con soporte a Ajax.

Auth Module: Indica si el Framework tiene incluido un módulo para el manejo de la autenticación de usuarios.

Modules: Indica si el Framework tiene otros módulos, como lector de RSS, módulo PDF u otros de utilidad.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Para la realización del cuadro comparativo se escogió algunos frameworks php open source.

Cuadro comparativo frameworks php opensource

Características Frameworks Php Open Source												
PHP Framework	PHP4	PHP5	MVC	Multiple DB's	OMR	DB Object	Templates	Caching	Validation	Ajax	Auth Module	Modules
Symfony	-	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓
CodeIgniter	✓	✓	✓	✓	-	✓	✓	✓	✓	-	-	-
CakePHP	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓
PHPDevShell	-	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓
DIY	-	✓	✓	-	✓	✓	✓	✓	-	✓	-	-
QPHP	✓	✓	✓	✓	-	✓	✓	-	✓	✓	✓	✓
WASP	-	✓	✓	-	-	✓	✓	-	✓	✓	✓	✓

Tabla 4: Cuadro comparativo Frameworks php opensource
Elaborado por: Diego Poaquiza – Investigador

Además se realizó un cuadro comparativo sobre aspectos que no están especificados dentro del contexto de las características de los Frameworks especificados anteriormente, cuyos puntos serán importantes para una selección adecuada, y concisa para el desarrollo de la aplicación.

Aspectos generales					
PHP Framework	Conocimiento	Experiencia de uso	Comunidad Activa	Actualizaciones periódicas	Manual actualizado
Symfony	-	✓	✓	✓	✓
CodeIgniter	✓	-	✓	✓	✓
CakePHP	✓	✓	✓	✓	✓
PHPDevShell	-	-	-	-	-
DIY	-	-	✓	✓	-
QPHP	-	-	-	-	-
WASP	-	-	-	-	-

Tabla 5: Cuadro comparativo aspectos generales Frameworks php opensource
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

De acuerdo a las diferentes características y aspectos analizados en los cuadros comparativos, se escogió CakePHP como framework para el desarrollo de la aplicación por sus mejores características sobre los otros frameworks, además se puso énfasis en los aspectos generales tomando en cuenta la experiencia, conocimiento y uso del framework CakePHP.

Motor de bases de datos

Se utilizó Mysql ya que es un software de código abierto de licencia GPL, veloz robusto de gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos, Flexible sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.

Complementos para el desarrollo de la aplicación web.

- **Jquery.**
- **JavaScript.**
- **Ajax.**
- **Css.**
- **Json.**
- **Bootstrap.**
- **Font Awesome.**

Para el buen funcionamiento de nuestra aplicación web tanto en el lado del servidor como en el lado del cliente se utilizó los siguientes requisitos, considerando que se cuenta con los siguientes requisitos.

Equipo Servidor

Sistema Operativo:

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

- Ubuntu 13.04 (Mini Ordenador Cubieboard A20).

Servicios:

- Apache 2.4.7.
- Mysql 5.5.40.
- Php 5.3.8.

Equipo Cliente

Sistema Operativo:

- Windows Xp, Windows 7 o superior.
- Linux (kernel 2.6 o superior).

Navegadores:

- Microsoft Internet Explorer 8 o Superior.
- Mozilla Firefox 15.0 y Chrome 15.0 (o Superior).

Estos navegadores se sugiere tenerlos actualizados para un buen funcionamiento de la aplicación web.

Requerimientos Hardware

Los requerimientos recomendados de hardware, son las características más apropiadas para poder ejecutar la aplicación correctamente, por lo que se ha tomado énfasis los requisitos proporcionados por la empresa para la descripción de las características mínimas descritas a continuación.

Servidor, requerimientos mínimos

- Procesador Intel Core Quad 2,33 GHz.
- Mínimo 1 GB de RAM.
- Mínimo 100 GB de espacio disponible en disco duro.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Clientes, requerimientos mínimos

- Procesador Intel Pentium 4.
- 512 MB de memoria RAM.
- 50 GB de disco duro.

Se cuenta con una mini computadora llamada cubieboard A20 que será usada como servidor la cual tiene características suficientes para que la aplicación web funcione correctamente, cabe recalcar que si se desea implementar el sistema en toda la cooperativa será necesario mudar la aplicación a un servidor con mejores características.

Servidor, requerimientos mínimos

- Procesador de 86 o 64 bits de doble núcleo de 2,9 gigahercios (GHz) o más.
- 2 GB de RAM o más.
- Mínimo 700 GB de espacio disponible en disco duro o más.

Diagramas UML

Se utilizó los diagramas UML (Lenguaje Unificado de Modelado), ya que es el lenguaje de modelado de sistemas de software utilizado por numerosos organismos y empresas para capturar información de cómo un sistema o negocio trabajará, además de la definición de aspectos conceptuales tales como procesos de negocio y funciones del sistema.

Diagrama de Casos de Uso

Los diagramas de uso se suelen utilizar en el modelado del sistema desde el punto de vista de los usuarios para representar las acciones que realiza cada tipo de usuario, por lo tanto los casos de uso determinan los requisitos funcionales del sistema, es decir, representan las funciones que un sistema puede ejecutar.

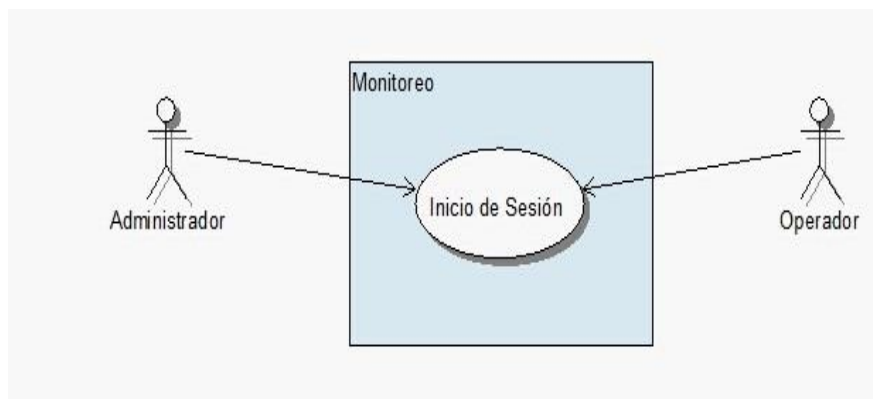


Fig. 7: Diagrama Casos de Uso Inicio de Sesión Administrador-Operador
Elaborado por: Diego Poaquiza – Investigador

Especificación Caso de Uso Inicio de Sesión

Caso de Uso: Inicio de Sesión
<p>Descripción: Permite ingresar la aplicación web y visualizar la página de inicio según el tipo de usuario.</p>
<p>Actores: Usuario Administrador. Usuario Operador.</p>
<p>Precondiciones: El usuario debe estar registrado en el sistema.</p>
<p>Flujo Normal:</p> <ul style="list-style-type: none"> • El actor digita su nombre de usuario y contraseña valida. • Pulsar el botón de Inicio de sesión.
<p>Flujo Alternativo:</p> <ul style="list-style-type: none"> ✓ El sistema comprueba la validez de los datos del usuario. • Si los datos son correctos ingresa el usuario al sistema. • Si los datos son incorrectos envía un mensaje de error al usuario.
<p>Post Condiciones:</p> <ul style="list-style-type: none"> • El usuario ingresa al sistema mostrándole la página principal según su rol.

Tabla 6: Inicio de Sesión
Elaborado por: Diego Poaquiza – Investigador

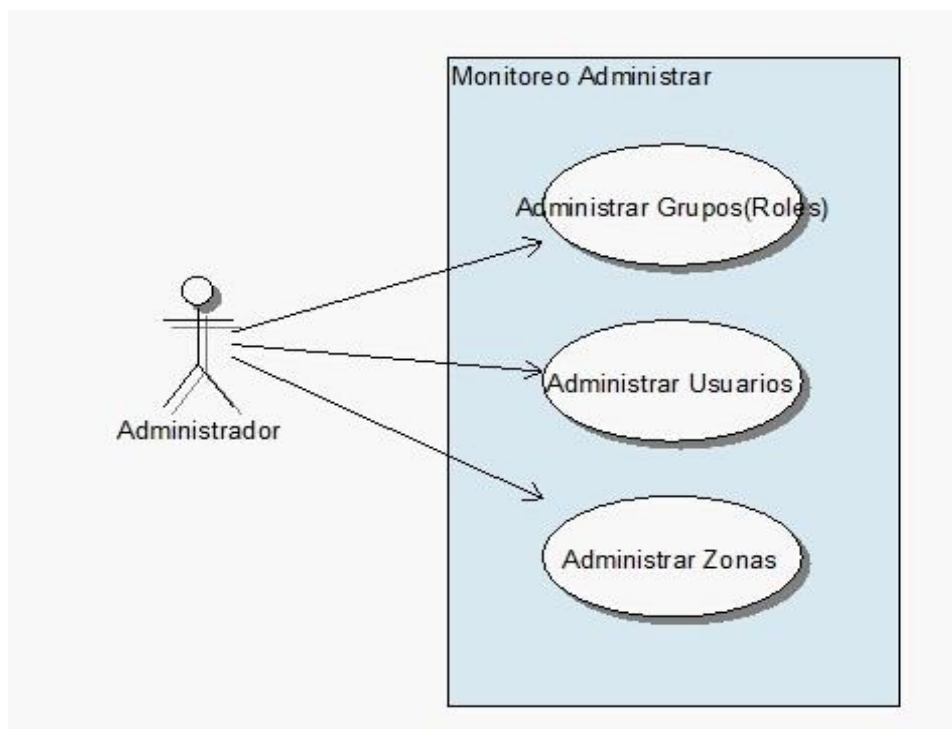


Fig. 8: Diagrama Casos de Uso Grupos, Usuarios y Zonas - Administrador
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Grupos (Ingresar)
Descripción: Permite ingresar un nuevo grupo (rol) al sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Usuarios y Grupos. • Seleccionar el botón grupos. • Presionar el botón nuevo. • Ingresar los datos. • Presionar el botón de Guardar.
Flujo Alternativo: <ul style="list-style-type: none"> • El sistema valida los datos ingresados. • Si los datos son correctos se graba con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se crea un nuevo Grupo (Rol) al sistema.

Tabla 7: Administrar Grupos (Ingresar)
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Administrar Grupos (Editar)
Descripción: Permite editar un grupo (rol) del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Usuarios y Grupos. • Seleccionar el botón grupos. • Seleccionar el grupo a modificar y presionar el botón Editar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Editar. <ul style="list-style-type: none"> • Modificar los datos de Grupos (Roles) que desee. • Presionar el botón de Guardar. • El sistema valida los datos. • Si los datos son correctos actualiza con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se actualiza un Grupo (Role) del sistema.

Tabla 8: Administrar Grupos (Editar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Grupos (Eliminar)
Descripción: Permite eliminar un grupo (rol) del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Usuarios y Grupos. • Seleccionar el botón grupos. • Seleccionar el grupo a eliminar y presionar el botón Eliminar. • Aparece un mensaje de confirmación con los botones Aceptar, Cancelar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones: <ul style="list-style-type: none"> • Se elimina un Grupo (Rol) del sistema.

Tabla 9: Administrar Grupos (Eliminar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Usuarios (Ingresar)
Descripción: Permite ingresar un nuevo usuario al sistema con su respectivo rol.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Usuarios y Grupos. • Seleccionar el botón usuarios. • Presionar el botón nuevo. • Ingresar los datos del nuevo usuario. • Seleccionar el rol al cual pertenece el nuevo usuario. • Presionar el botón de Guardar.
Flujo Alternativo: <ul style="list-style-type: none"> • El sistema valida los datos ingresados. • Si los datos son correctos se graba con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se crea un nuevo Usuario con su respectivo rol.

Tabla 10: Administrar Usuarios (Ingresar)
 Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Usuarios (Editar)
Descripción: Permite editar un usuario del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Usuarios y Grupos. • Seleccionar el botón usuarios. • Seleccionar el usuario a modificar y presionar el botón Editar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Editar. • Modificar los datos del Usuario que desee. • Presionar el botón de Guardar. • El sistema valida los datos. • Si los datos son correctos actualiza con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se actualiza un Usuario del sistema.

Tabla 11: Administrar Usuarios (Editar)
 Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Administrar Usuarios (Eliminar)
Descripción: Permite eliminar un usuario del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Usuarios y Grupos. • Seleccionar el botón usuarios. • Seleccionar el usuario a eliminar y presionar el botón Eliminar. • Aparece un mensaje de confirmación con los botones Aceptar, Cancelar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones: <ul style="list-style-type: none"> • Se elimina un Usuario del sistema.

Tabla 12: Administrar Usuarios (Eliminar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Zonas (Ingresar)
Descripción: Permite ingresar una nueva zona al sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Zonas. • Presionar el botón nuevo. • Ingresar los datos de la nueva Zona. • Presionar el botón de Guardar.
Flujo Alternativo: <ul style="list-style-type: none"> • El sistema valida los datos ingresados. • Si los datos son correctos se graba con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se crea una nueva Zona al sistema.

Tabla 13: Administrar Zonas (Ingresar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Zonas (Editar)
Descripción: Permite editar una zona del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Zonas. • Seleccionar la Zona a modificar y presionar el botón Editar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Editar. <ul style="list-style-type: none"> • Modificar los datos de la Zona que desee. • Presionar el botón de Guardar. • El sistema valida los datos. • Si los datos son correctos actualiza con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se actualiza una Zona del sistema.

Tabla 14: Administrar Zonas (Editar)
 Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Zonas (Eliminar)
Descripción: Permite eliminar una zona del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Zonas. • Seleccionar la ubicación a eliminar y presionar el botón Eliminar. • Aparece un mensaje de confirmación con los botones Aceptar, Cancelar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones: <ul style="list-style-type: none"> • Se elimina una Zona del sistema.

Tabla 15: Administrar Zonas (Eliminar)
 Elaborado por: Diego Poaquiza – Investigador

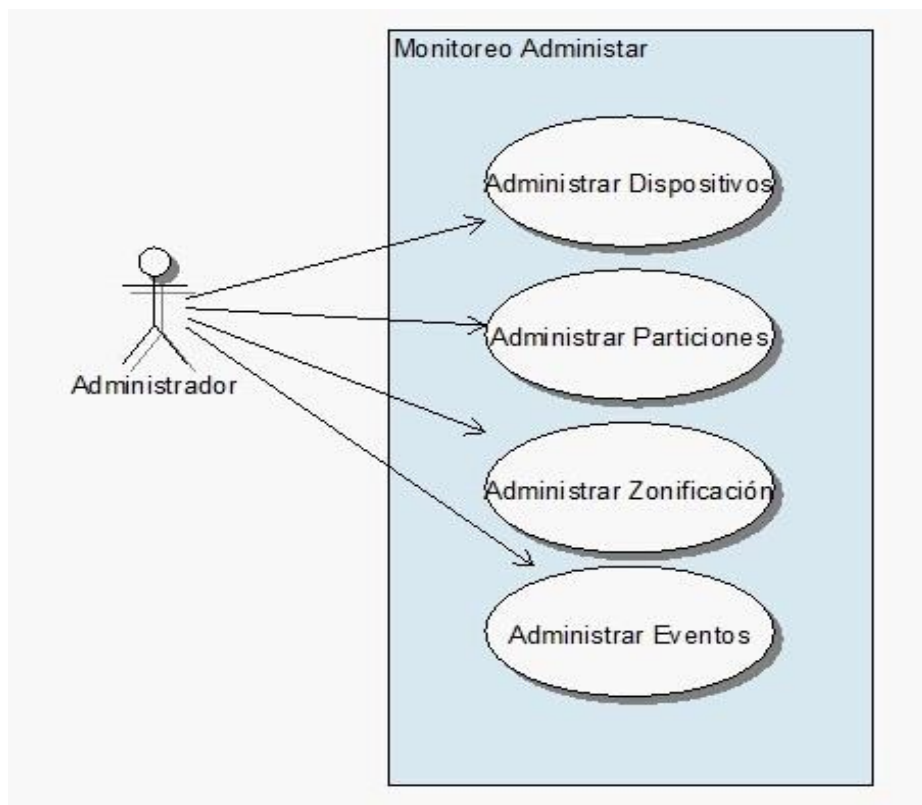


Fig. 9: Diagrama Casos de Uso Dispositivo, Partición, Zonif, y Evento - Administrador
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Dispositivos (Ingresar)

Descripción:

Permite ingresar un nuevo dispositivo o tipo de dispositivo al sistema.

Actores:

Usuario Administrador.

Flujo Normal:

1. Seleccionar la opción Administrar Dispositivos.
2. El sistema muestra las opciones disponibles: Dispositivos y Tipos de Dispositivos.
3. El usuario escoge la opción Dispositivos.
4. El usuario presiona el botón nuevo.
5. Ingresar los datos.
6. Seleccionar el tipo de dispositivo al que pertenece el nuevo dispositivo.
7. Presionar el botón de Guardar.
8. El sistema valida los datos ingresados.
9. Si los datos son correctos se graba con éxito.
10. Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.

Flujo Alternativo:

1. En el paso 3 del flujo normal el usuario escoge la opción Tipos de Dispositivos.
2. El usuario presiona el botón nuevo.
3. Ingresar los datos.
4. Presionar el botón de Guardar.
5. El sistema valida los datos ingresados.
6. Si los datos son correctos se graba con éxito.
7. Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.

Post Condiciones:

- Se crea un nuevo Dispositivo o Tipo de Dispositivo al sistema.

Tabla 16: Administrar Dispositivos (Ingresar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Dispositivos (Editar)**Descripción:**

Permite editar un dispositivo o tipo de dispositivo del sistema.

Actores:

Usuario Administrador.

Flujo Normal:

1. Seleccionar la opción Administrar Dispositivos.
2. El sistema muestra las opciones disponibles: Dispositivos y Tipos de Dispositivos.
3. El usuario escoge la opción Dispositivos.
4. Seleccionar el dispositivo a modificar y presionar el botón Editar.
5. Modifica los datos que desee.
6. Presiona el botón de Guardar.
7. El sistema valida los datos.
8. Si los datos son correctos actualiza con éxito.
9. Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.

Flujo Alternativo:

1. En el paso 3 del flujo normal el usuario escoge la opción Tipos de Dispositivos.
2. Seleccionar el Tipo de Dispositivo a modificar y presionar el botón Editar.
3. Modifica los datos que desee.
4. Presiona el botón de Guardar.
5. El sistema valida los datos.
6. Si los datos son correctos actualiza con éxito.
7. Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.

Post Condiciones:

- Se actualiza un Dispositivo o Tipo de Dispositivo del sistema.

Tabla 17: Administrar Dispositivos (Editar)
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Administrar Dispositivos (Eliminar)
Descripción: Permite eliminar un dispositivo o tipo de dispositivo del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ol style="list-style-type: none"> 1. Seleccionar la opción Administrar Dispositivos. 2. El sistema muestra las opciones disponibles: Dispositivos y Tipos de Dispositivos. 3. El usuario escoge la opción Dispositivos. 4. Selecciona el dispositivo a eliminar y presiona el botón Eliminar. 5. Aparece un mensaje de confirmación con los botones Aceptar, Cancelar. <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Flujo Alternativo: <ol style="list-style-type: none"> 1. En el paso 3 del flujo normal el usuario escoge la opción Tipos de Dispositivos. 2. Seleccionar el Tipo de Dispositivo a eliminar y presionar el botón Eliminar. 3. Aparece un mensaje de confirmación con los botones Aceptar, Cancelar. <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones: <ul style="list-style-type: none"> • Se elimina un Dispositivo o Tipo de Dispositivo del sistema.

Tabla 18: Administrar Dispositivos (Eliminar)
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Administrar Particiones (Ingresar)
Descripción: Permite ingresar una nueva partición al sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Particiones. • Presionar el botón nuevo. • Ingresar los datos de la nueva particiones. • Presionar el botón de Guardar.
Flujo Alternativo: <ul style="list-style-type: none"> • El sistema valida los datos ingresados. • Si los datos son correctos se graba con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se crea una nueva Partición al sistema.

Tabla 19: Administrar Particiones (Ingresar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Particiones (Editar)
Descripción: Permite editar una partición del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Particiones. • Seleccionar la partición a modificar y presionar el botón Editar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Editar. • Modificar los datos de la Partición que desee. • Presionar el botón de Guardar. • El sistema valida los datos. • Si los datos son correctos actualiza con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se actualiza una Partición del sistema.

Tabla 20: Administrar Particiones (Editar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Particiones (Eliminar)
Descripción: Permite eliminar una partición del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Particiones. • Seleccionar la partición a eliminar y presionar el botón Eliminar. • Aparece un mensaje de confirmación con los botones Aceptar, Cancelar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones: <ul style="list-style-type: none"> • Se elimina una Partición del sistema.

Tabla 21: Administrar Particiones (Eliminar)
Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Zonificación (Ingresar)
Descripción: Permite ingresar una nueva zonificación al sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Zonificación. • Presionar el botón nuevo. • Ingresar los datos de la nueva zonificación. • Presionar el botón de Guardar.
Flujo Alternativo: <ul style="list-style-type: none"> • El sistema valida los datos ingresados. • Si los datos son correctos se graba con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se crea una nueva zonificación al sistema.

Tabla 22: Administrar Zonificación (Ingresar)
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Administrar Zonificación (Editar)
Descripción: Permite editar una zonificación del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Zonificación. • Seleccionar la partición a modificar y presionar el botón Editar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Editar. <ul style="list-style-type: none"> • Modificar los datos de la Partición que desee. • Presionar el botón de Guardar. • El sistema valida los datos. • Si los datos son correctos actualiza con éxito. • Si los datos son requeridos o incorrectos el sistema le muestra un mensaje ya sea de campos requeridos o datos erróneos.
Post Condiciones: <ul style="list-style-type: none"> • Se actualiza una Zonificación del sistema.

Tabla 23: Administrar Zonificación (Editar)
 Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Zonificación (Eliminar)
Descripción: Permite eliminar una zonificación del sistema.
Actores: Usuario Administrador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Administrar Zonificación. • Seleccionar la zonificación a eliminar y presionar el botón Eliminar. • Aparece un mensaje de confirmación con los botones Aceptar, Cancelar.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Aceptar. <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. ✓ Cancelar: <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones: <ul style="list-style-type: none"> • Se elimina una Zonificación del sistema.

Tabla 24: Administrar Zonificación (Eliminar)
 Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Administrar Eventos (Eliminar)
Descripción: Permite eliminar un registro del sistema.
Actores: Usuario Administrador.
Flujo Normal:
<ol style="list-style-type: none"> 1. Seleccionar la opción Consultar Evento. 2. El sistema muestra las opciones disponibles: Eventos Dispositivos o Eventos Usuarios. 3. El usuario escoge la opción Eventos Dispositivos. 4. Seleccionar el registro a eliminar y presionar el botón Eliminar. 5. Aparece un mensaje de confirmación con los botones Aceptar, Cancelar.
Flujo Alternativo:
<ol style="list-style-type: none"> 1. En el paso 3 del flujo normal el usuario escoge la opción Eventos Usuarios. 2. Seleccionar el registro a eliminar y presionar el botón Eliminar. 3. Aparece un mensaje de confirmación con los botones Aceptar, Cancelar. <p>✓ Aceptar.</p> <ul style="list-style-type: none"> • Valida si los datos están relacionados. • Si el registro está relacionado muestra un mensaje de error al usuario. • Si el registro no está relacionado, se borra de la lista y de la base de datos. <p>✓ Cancelar:</p> <ul style="list-style-type: none"> • Cierra cuadro de dialogo.
Post Condiciones:
<ul style="list-style-type: none"> • Se elimina un Evento del sistema.

Tabla 25: Administrar Eventos (Eliminar)
Elaborado por: Diego Poaquiza – Investigador

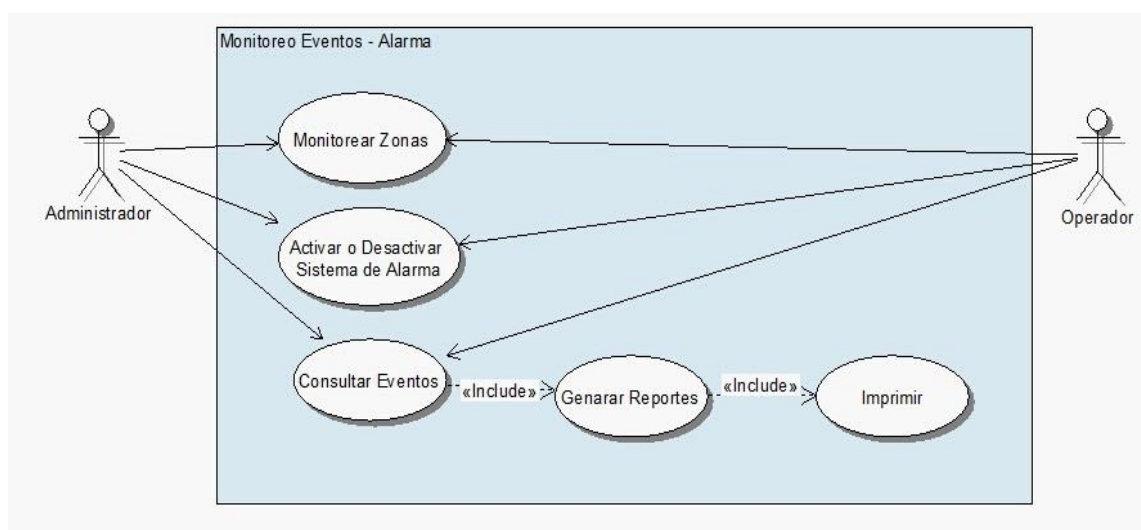


Fig. 10: Diagrama Casos de Uso Eventos Alarma Administrador - Operador
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Activar o Desactivar Sistema de Alarma
Descripción: Permite activar o desactivar el sistema de alarma.
Actores: Usuario Administrador. Usuario operador.
Flujo Normal: <ol style="list-style-type: none"> 1. Seleccionar la opción Monitorear Zonas. 2. El sistema muestra el Estado del Sistema con las opciones de Activado (On) o Desactivado (Off) del sistema de alarma. 3. Dependiendo del estado del sistema en el botón se muestra On u Off. 4. El usuario presiona el botón On. 5. El sistema valida el estado, procede a activar y muestra un mensaje de Sistema Activado. 6.
Flujo Alternativo: <ol style="list-style-type: none"> 1. En el paso 4 del flujo normal el usuario. 2. El usuario presiona el botón Off. 3. El sistema valida el estado, procede a desactivar y muestra un mensaje de Sistema Desactivado.
Post Condiciones: <ul style="list-style-type: none"> • Se activa o desactiva el sistema de alarma.

Tabla 26: Activar o Desactivar Sistema de Alarma
 Elaborado por: Diego Poaquiza – Investigador

Caso de Uso: Monitorear Zonas
Descripción: Permite visualizar los estados de los diferentes dispositivos instalados en las respectivas zonas del CPD.
Actores: Usuario Administrador. Usuario Operador.
Flujo Normal: <ul style="list-style-type: none"> • Seleccionar la opción Monitoreo.
Flujo Alternativo: <ul style="list-style-type: none"> ✓ Se presenta una ventana con los estados de los dispositivos instalados en las zonas respectivas.
Post Condiciones: <ul style="list-style-type: none"> • Se visualiza los estados de los dispositivos instalados en las diferentes zonas.

Tabla 27: Monitorear Zonas
 Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Caso de Uso: Consultar Eventos
<p>Descripción: Permite visualizar los eventos que se realizan en los Dispositivos y Usuarios y generar un reporte en formato pdf.</p>
<p>Actores: Usuario Administrador. Usuario Operador.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. Seleccionar la opción Consultar Eventos. 2. El sistema muestra las opciones disponibles: Eventos Dispositivos o Eventos Usuarios. 3. El usuario escoge la opción Eventos Dispositivos. 4. El usuario hace una búsqueda de acuerdo al criterio. 5. Presionar el botón de Generar reporte. 6. Se presenta una ventana emergente con el reporte de los datos, con las opciones del navegador: 7. Imprimir. 8. Guardar. 9. Cerrar.
<p>Flujo Alternativo:</p> <ol style="list-style-type: none"> 4. En el paso 3 del flujo normal el usuario escoge la opción Eventos Usuarios. 5. El usuario hace una búsqueda de acuerdo al criterio. 6. Presionar el botón de Generar reporte. 7. Se presenta una ventana emergente con el reporte de los datos, con las opciones del navegador: 8. Imprimir. 9. Guardar. 10. Cerrar.
<p>Post Condiciones:</p> <ul style="list-style-type: none"> • Los datos mostrados permanecen filtrados.

Tabla 28: Consultar Eventos
Elaborado por: Diego Poaquiza – Investigador

4.1.5 Diagramas de Secuencia

El diagrama de secuencias de UML muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo, indicarán los módulos o clases que forman parte del programa y las llamadas que se hacen en cada uno de ellos para realizar una tarea determinada.

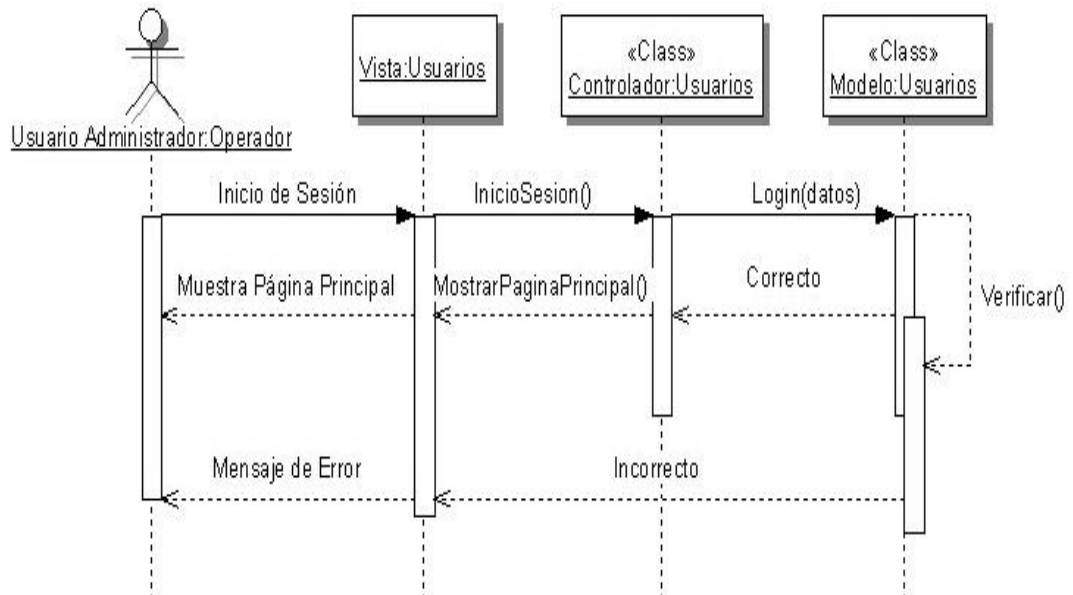


Fig. 11: Inicio de Sesión

Elaborado por: Diego Poaquiza – Investigador

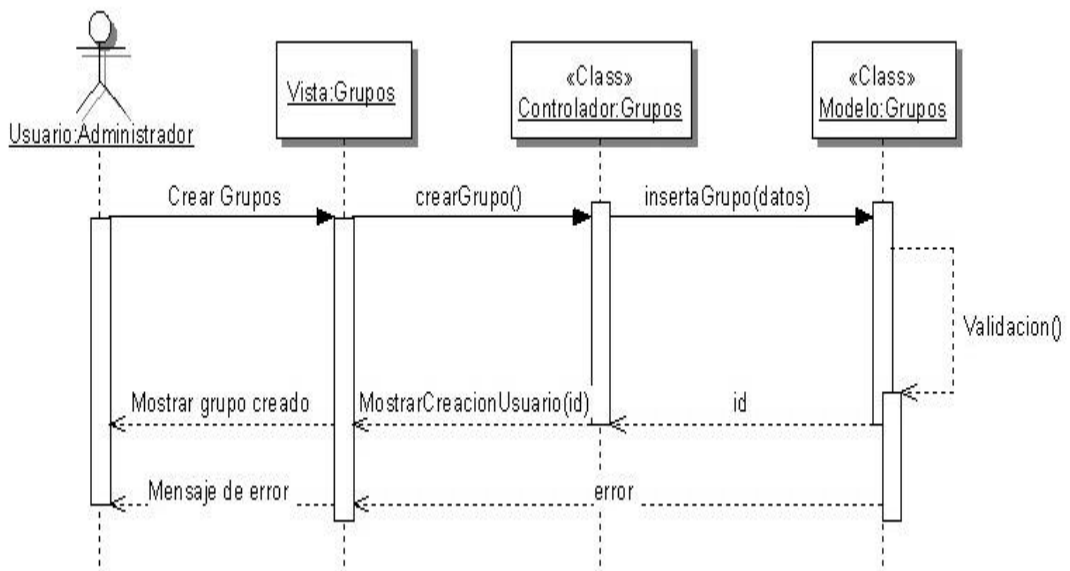


Fig. 12: Ingreso de Nuevo Grupo

Elaborado por: Diego Poaquiza – Investigador

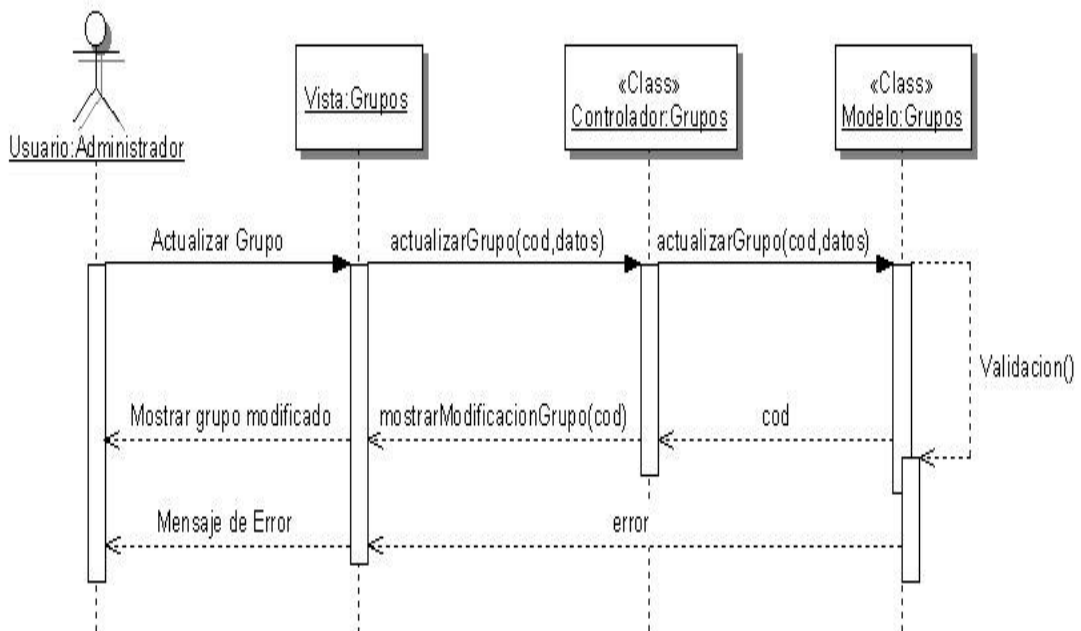


Fig. 13: Modificación de Grupo

Elaborado por: Diego Poquiza – Investigador

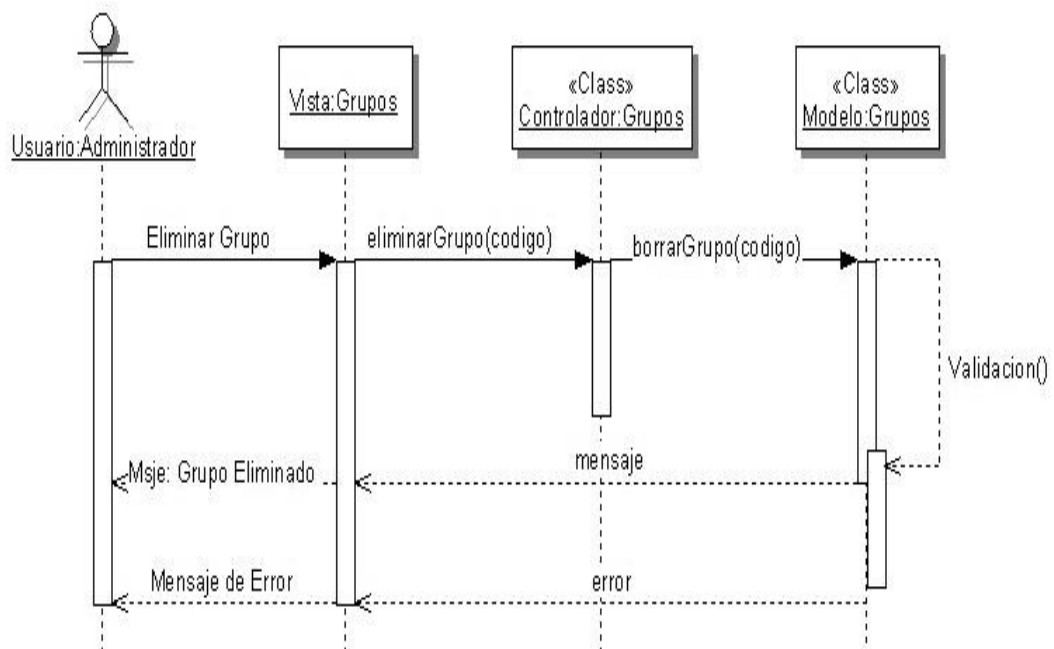


Fig. 14: Eliminación de Grupo

Elaborado por: Diego Poquiza – Investigador

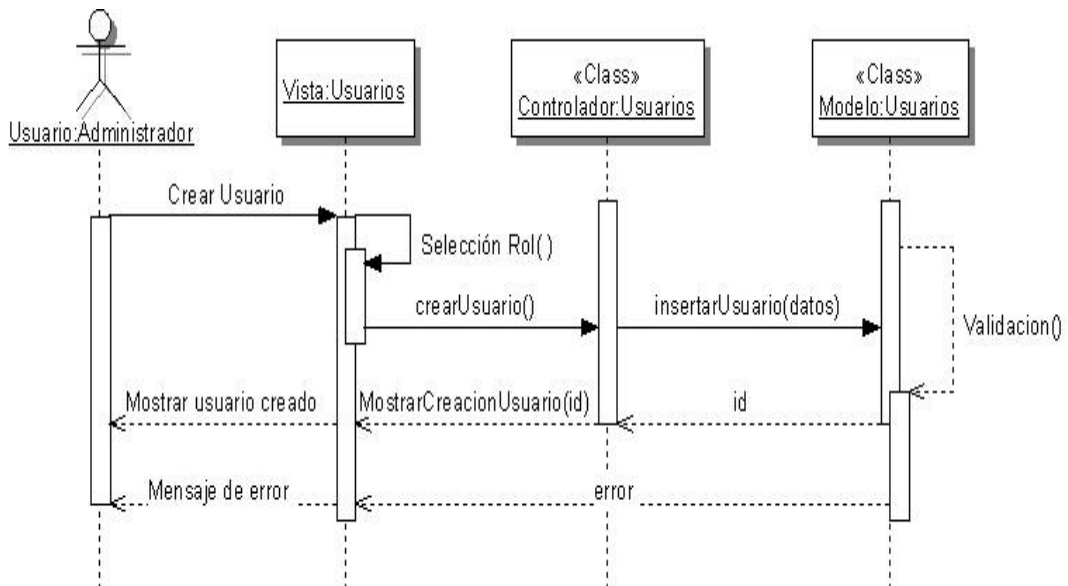


Fig. 15: Ingreso de Nuevo Usuario
 Elaborado por: Diego Poaquiza – Investigador

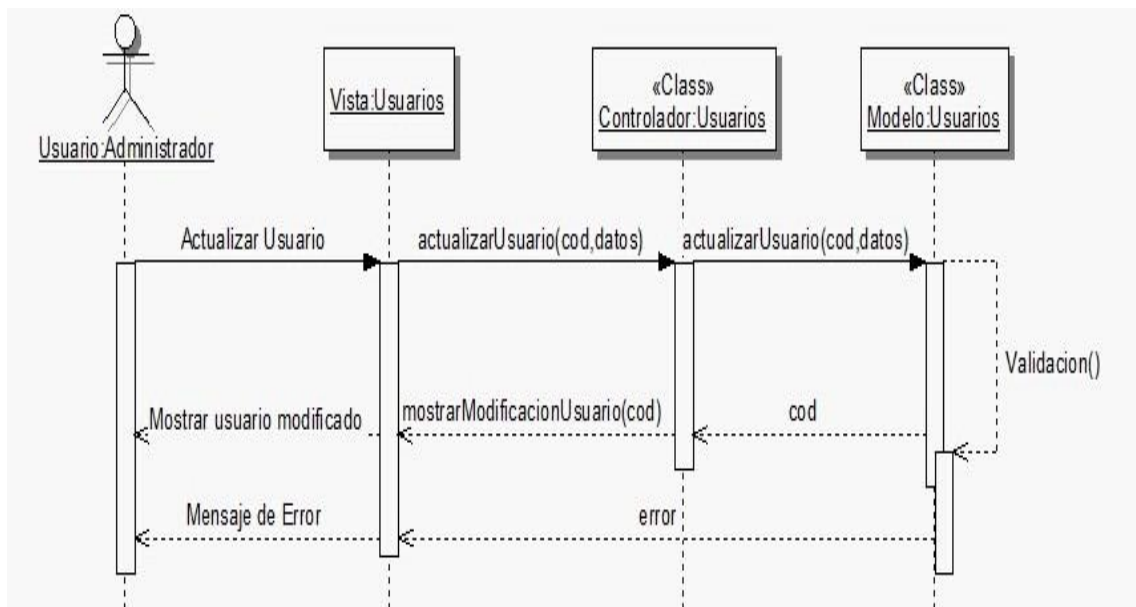


Fig. 16: Modificación de Usuario
 Elaborado por: Diego Poaquiza – Investigador

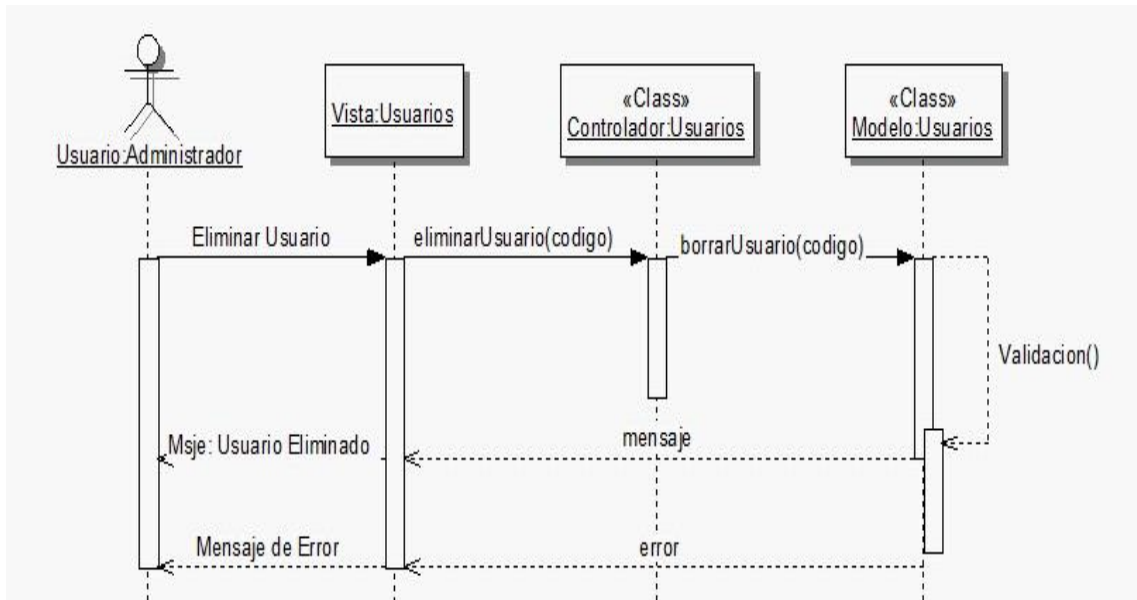


Fig. 17: Eliminación de Usuario

Elaborado por: Diego Poaquiza – Investigador

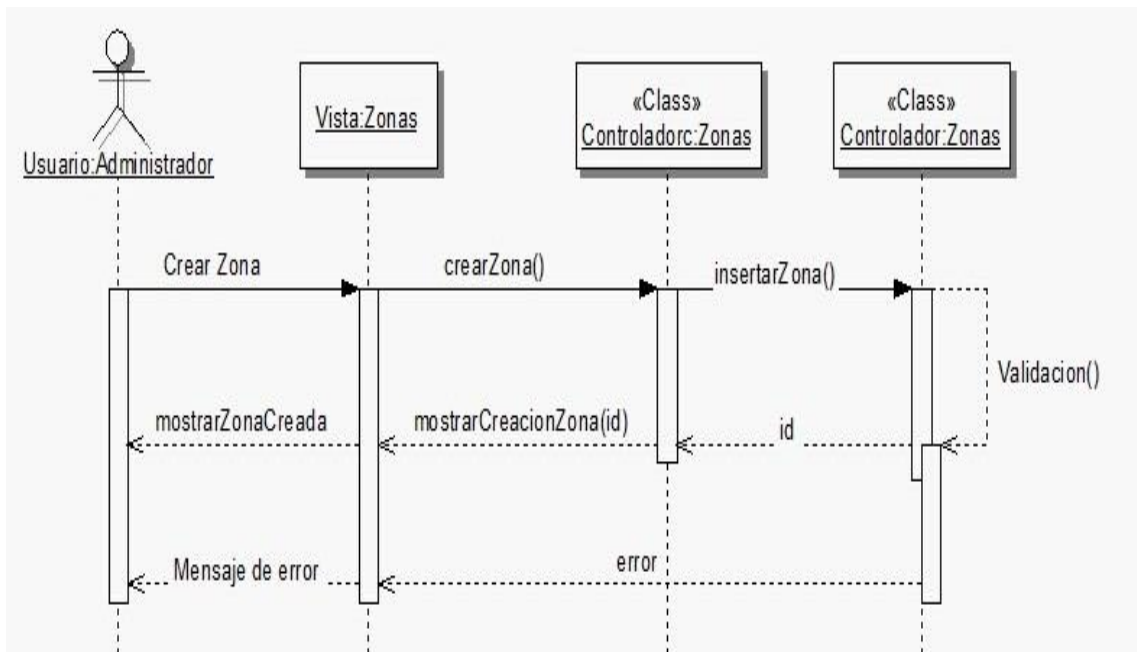


Fig. 18: Ingreso de Nueva Zona

Elaborado por: Diego Poaquiza – Investigador

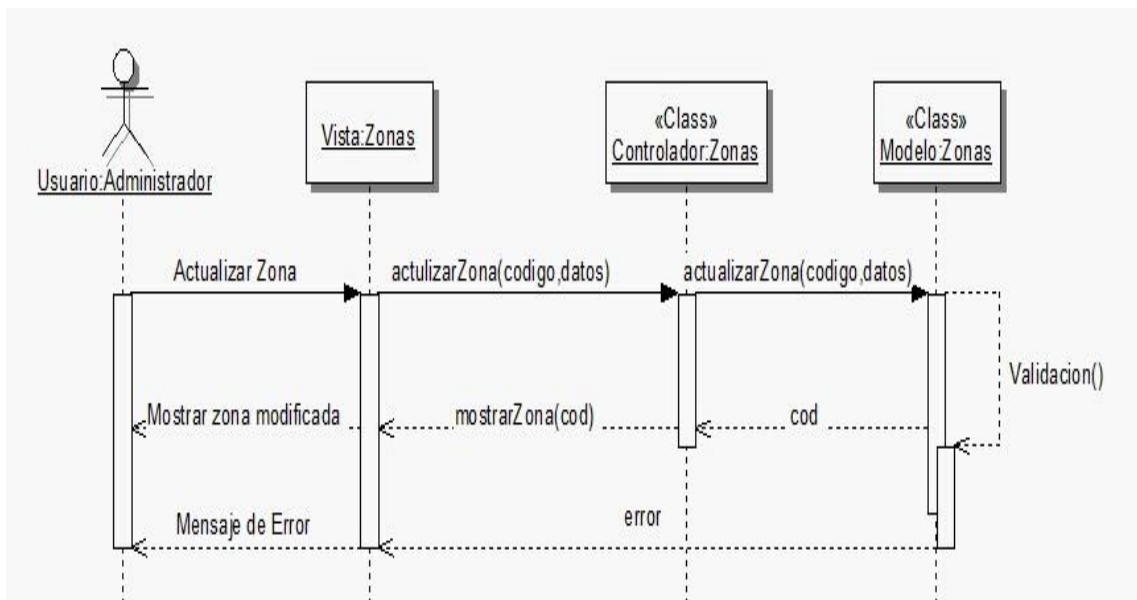


Fig. 19: Modificación de Zona

Elaborado por: Diego Poaquiza – Investigador

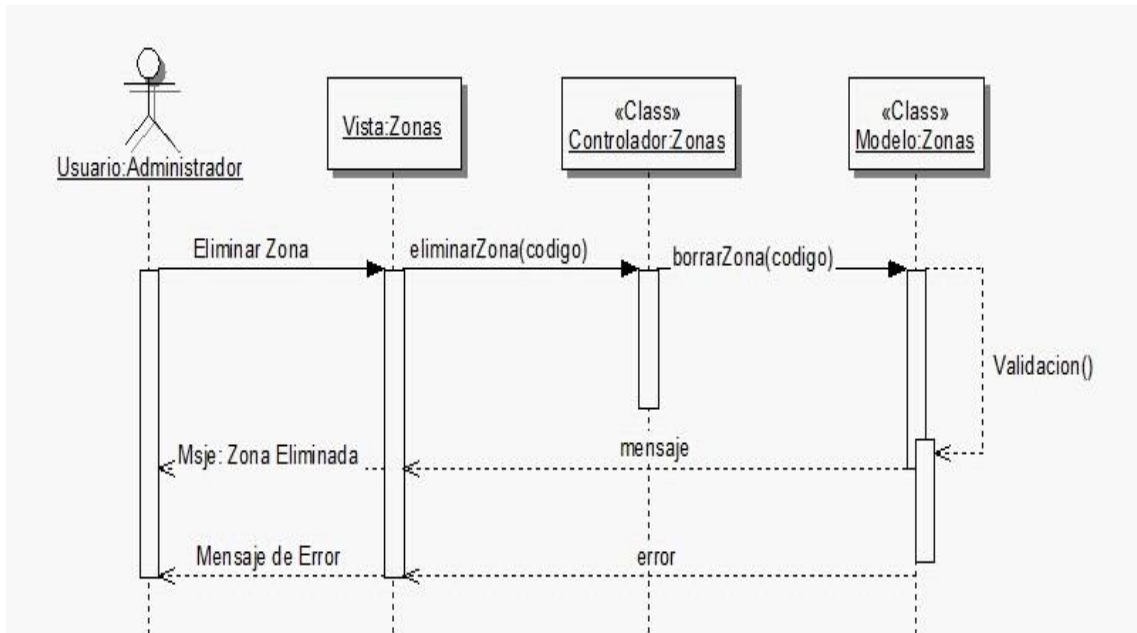


Fig. 20: Eliminación de Zona

Elaborado por: Diego Poaquiza – Investigador

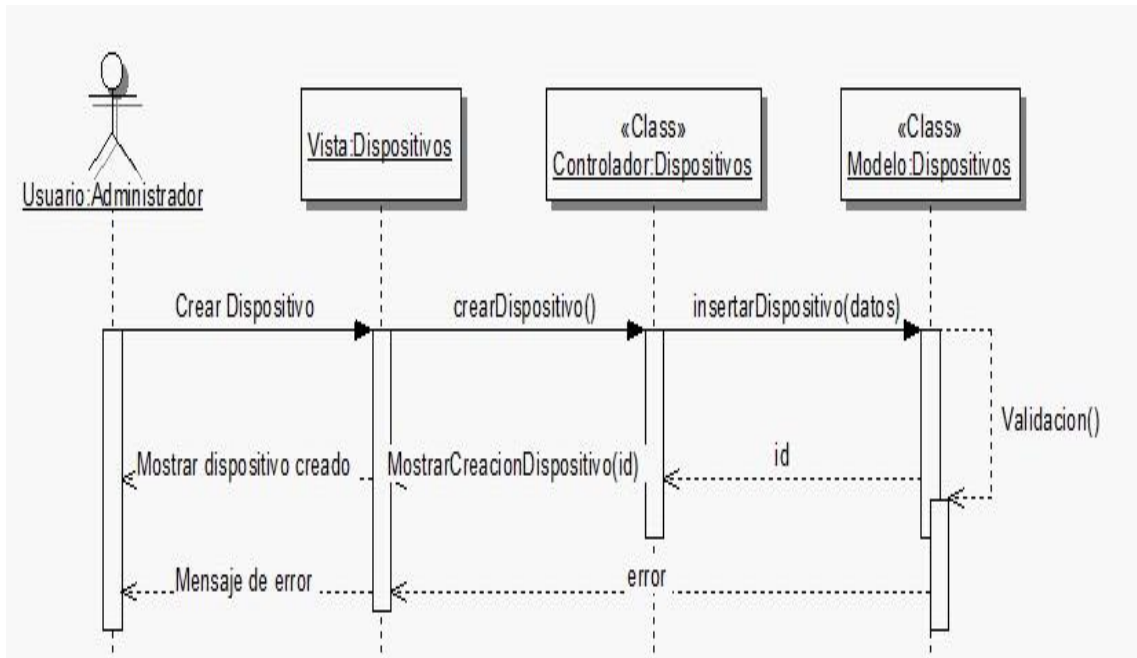


Fig. 21: Ingreso de nuevo Dispositivo

Elaborado por: Diego Poaquiza – Investigador

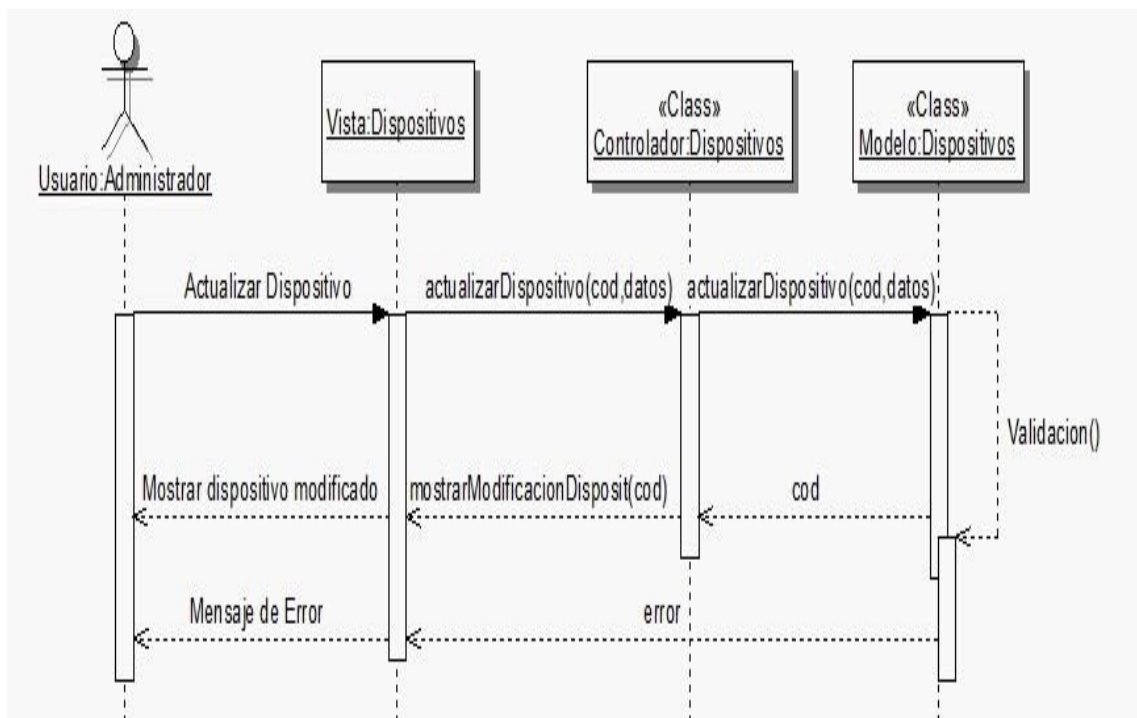


Fig. 22: Modificación de Dispositivo

Elaborado por: Diego Poaquiza – Investigador

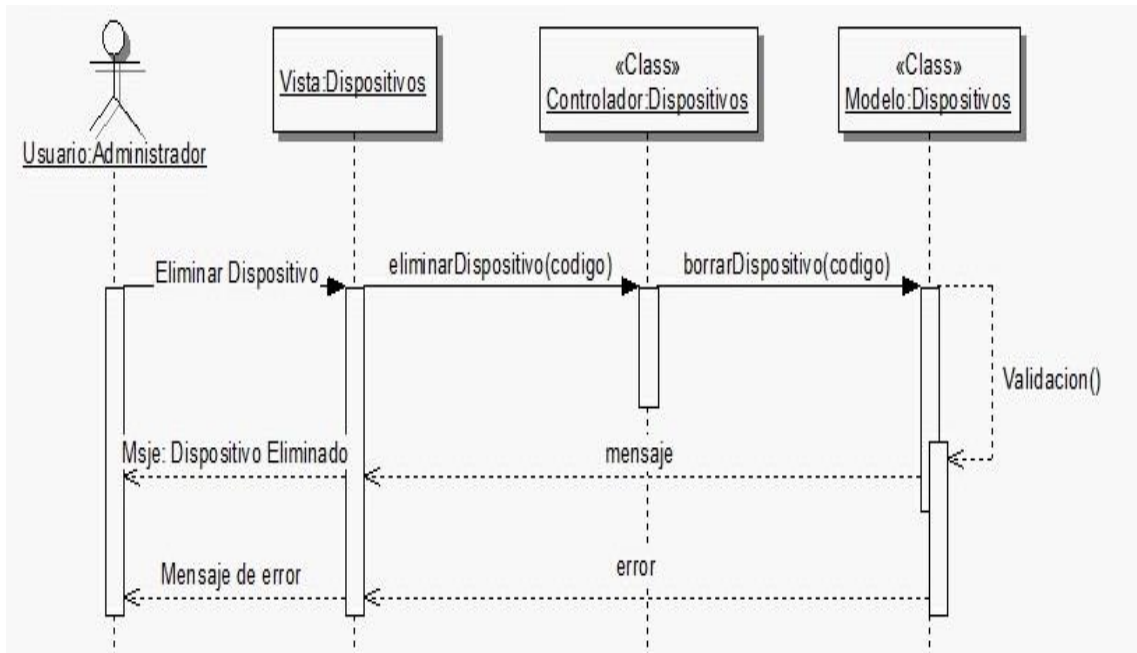


Fig. 23: Eliminación de Dispositivo

Elaborado por: Diego Poaquiza – Investigador

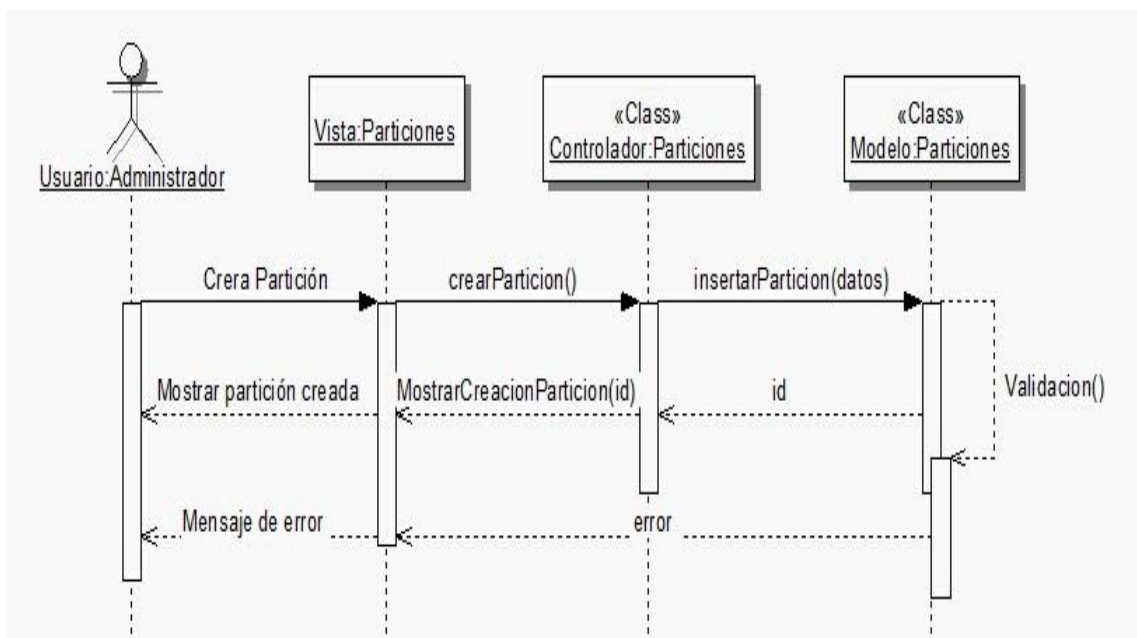


Fig. 24: Ingreso de Nueva Partición

Elaborado por: Diego Poaquiza – Investigador

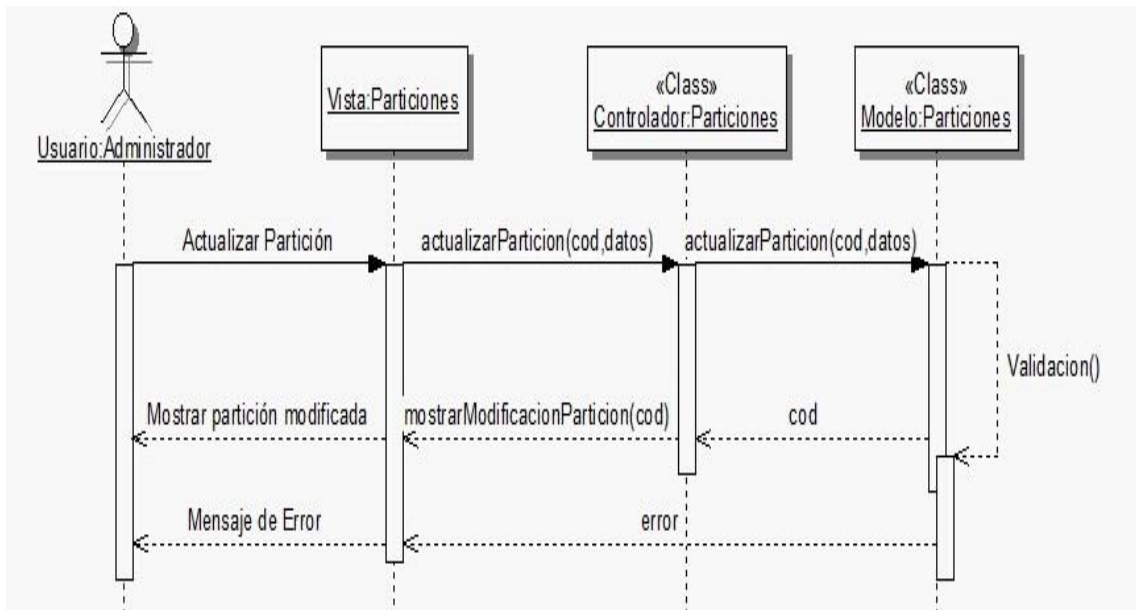


Fig. 25: Modificación de Partición

Elaborado por: Diego Poaquiza – Investigador

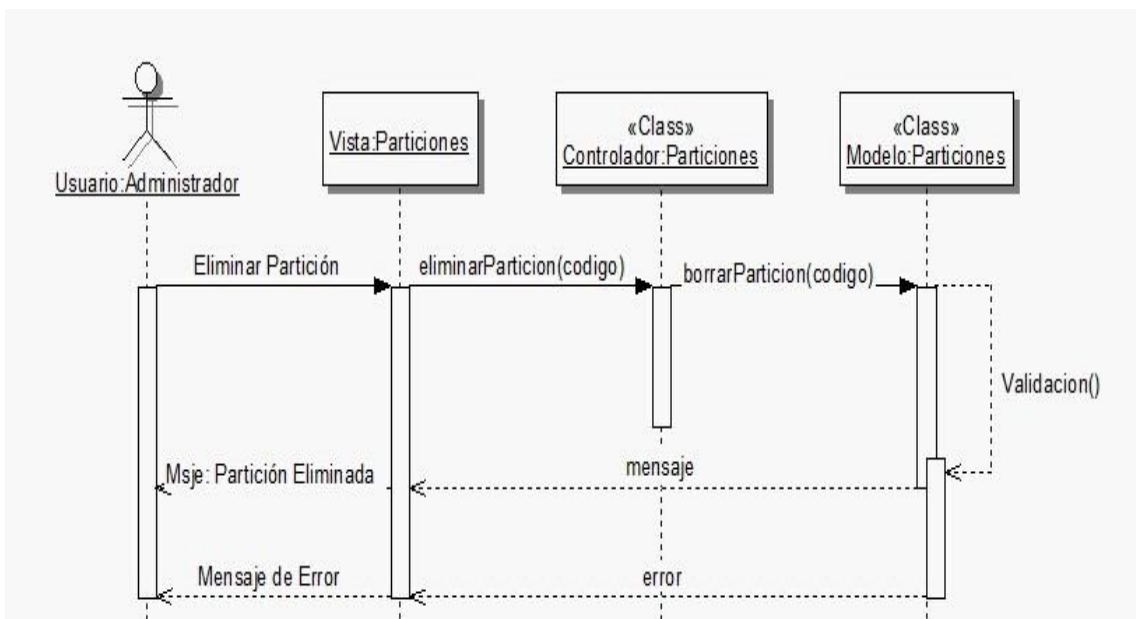


Fig. 26: Eliminación de Partición

Elaborado por: Diego Poaquiza – Investigador

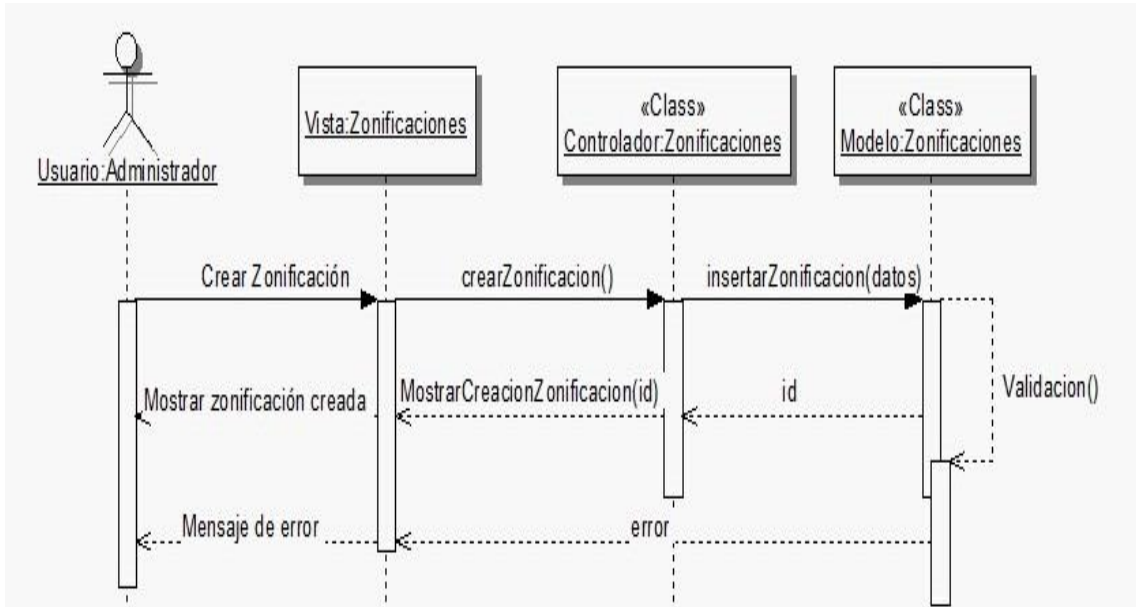


Fig. 27: Ingreso de Nueva Zonificación

Elaborado por: Diego Poaquiza – Investigador

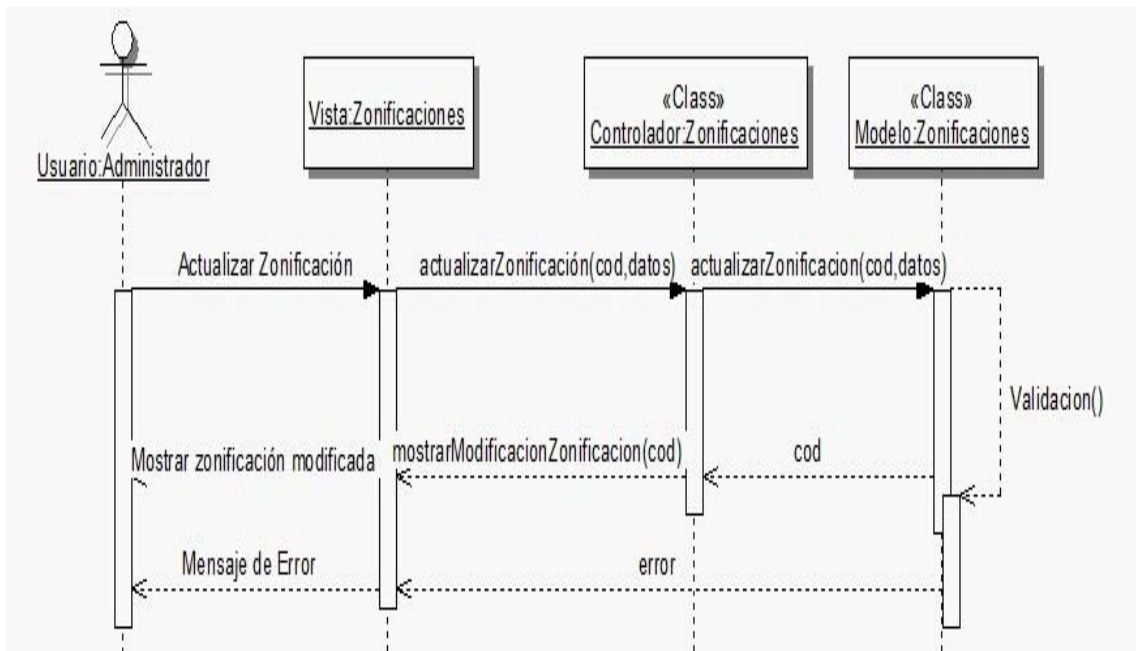


Fig. 28: Modificación de Zonificación

Elaborado por: Diego Poaquiza – Investigador

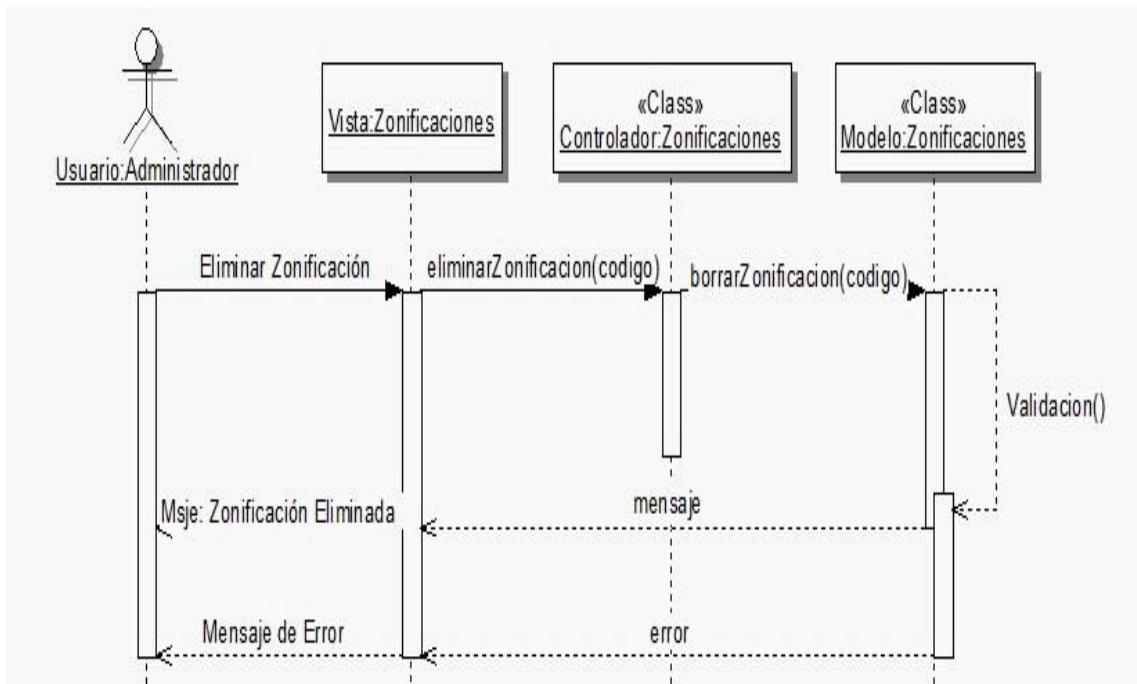


Fig. 29: Eliminación de Zonificación

Elaborado por: Diego Poaquiza – Investigador

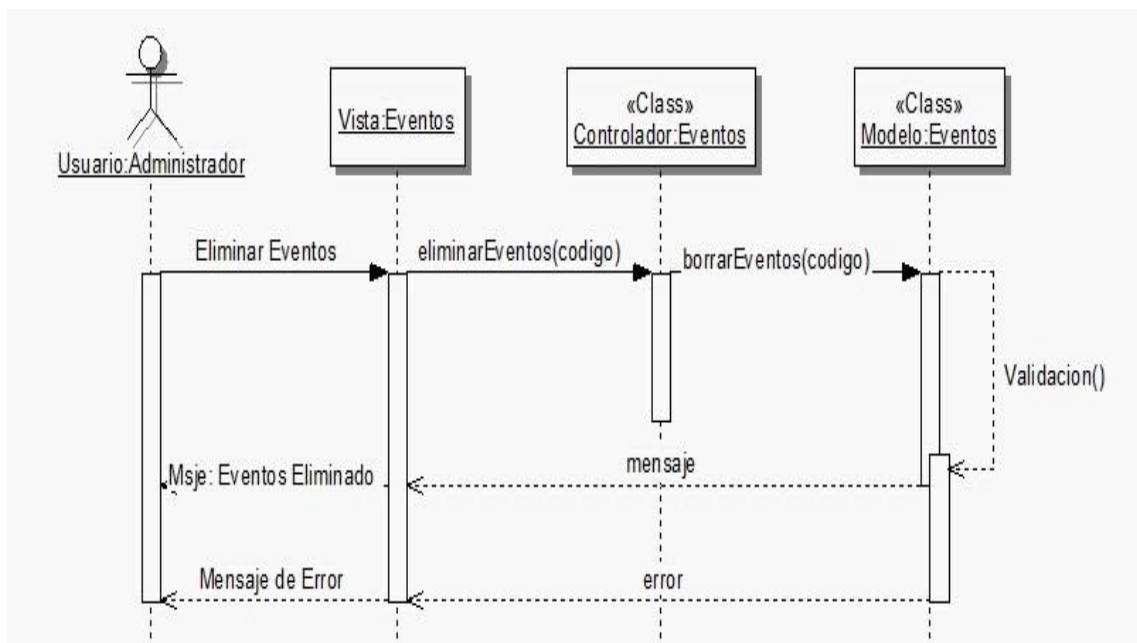


Fig. 30: Eliminación de Evento

Elaborado por: Diego Poaquiza – Investigador

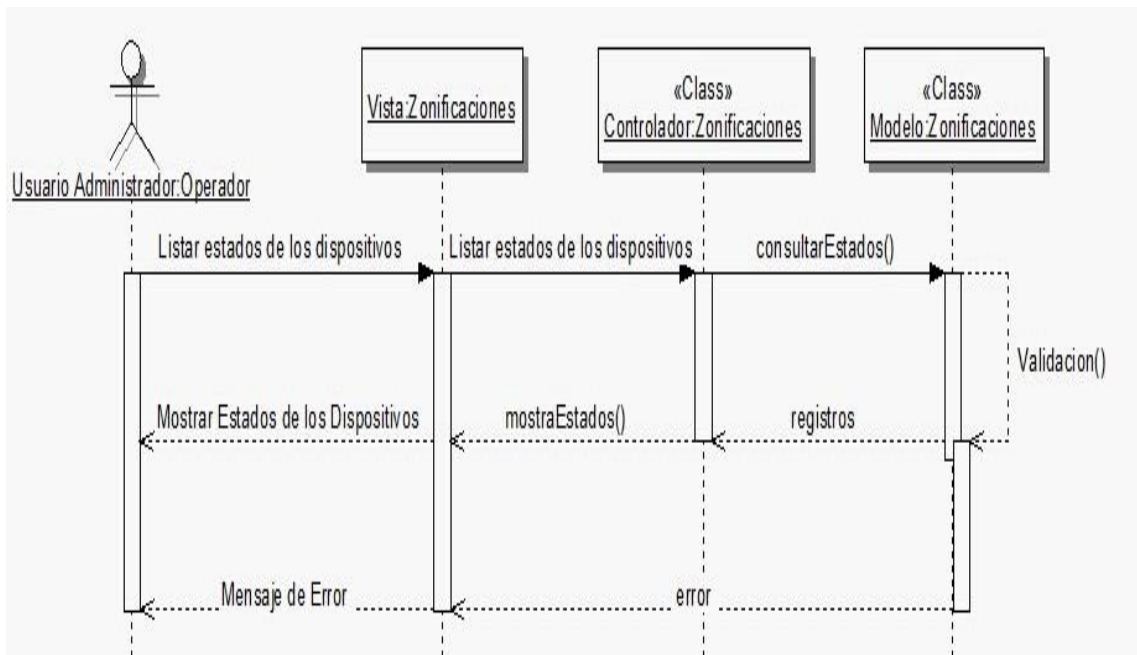


Fig. 31 Moniteo de Zonas

Elaborado por: Diego Poquiza – Investigador

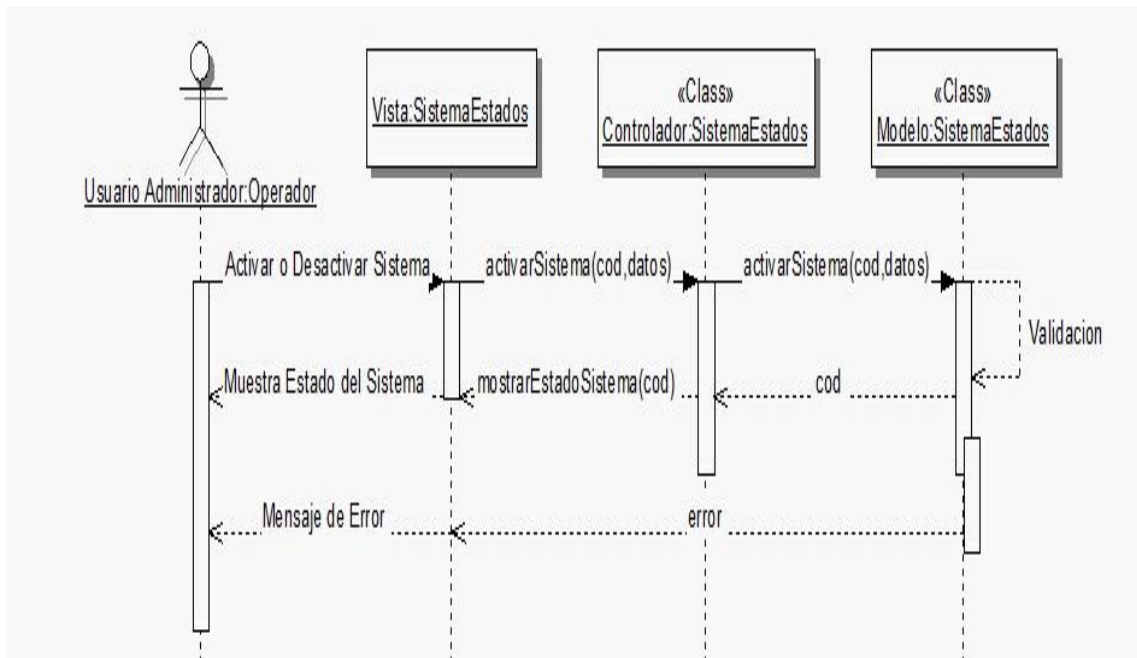


Fig. 32: Activar Sistema de Alarma

Elaborado por: Diego Poquiza – Investigador

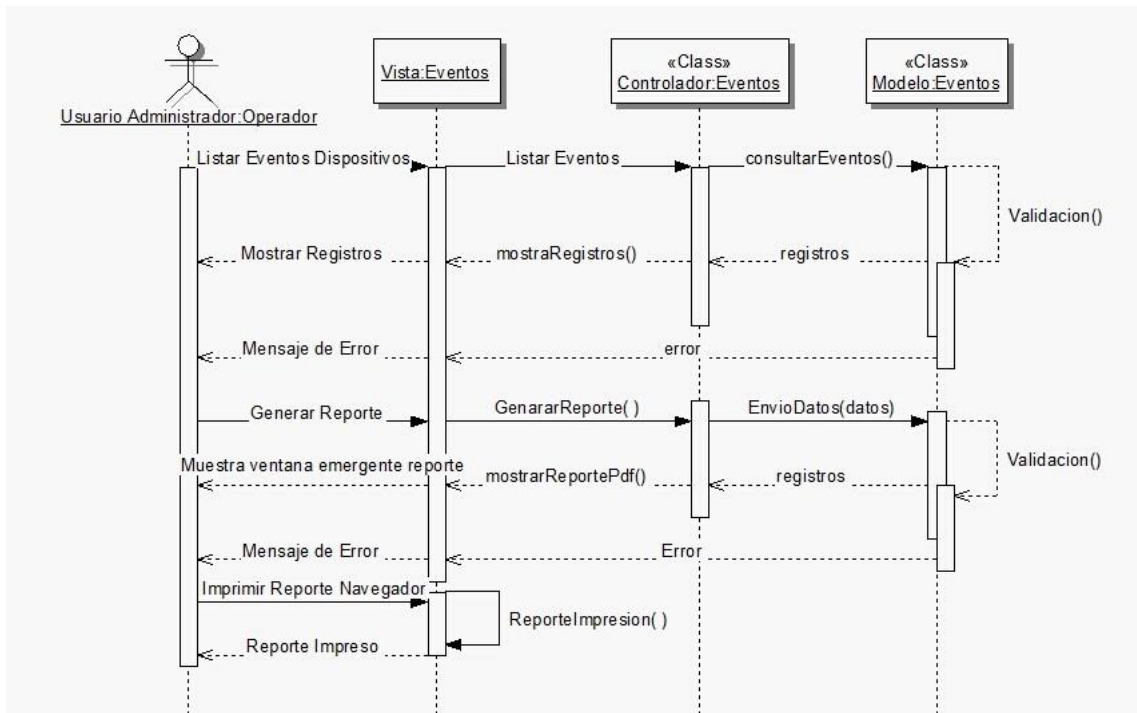


Fig. 33: Consulta de eventos – Reportes

Elaborado por: Diego Poaquiza – Investigador

4.2 Diseño

En esta etapa el diseño del software se describe la estructura de los datos además del detalle y caracterización de la interfaz.

El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

4.2.1 Diseño de la Base de Datos

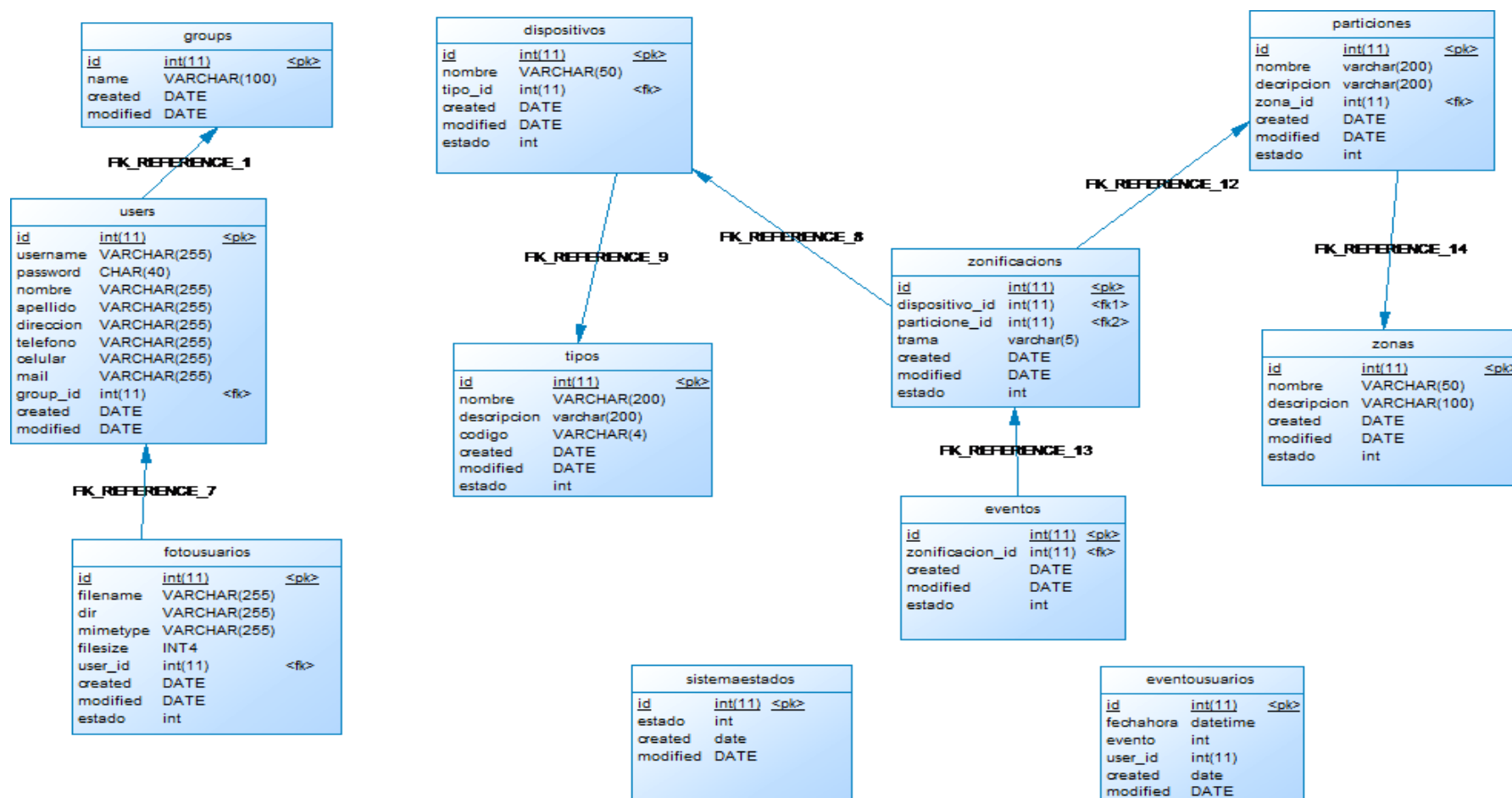


Fig. 34: Base de Datos Modelo Relacional
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Diccionario de Datos

Users					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	INT4	4	Si	No	Si
username	VARCHAR(255)	255	No	No	Si
password	CHAR(40)	40	No	No	Si
nombre	VARCHAR(255)	255	No	No	No
apellido	VARCHAR(255)	255	No	No	No
direccion	VARCHAR(255)	255	No	No	No
telefono	VARCHAR(255)	255	No	Si	No
celular	VARCHAR(255)	255	No	No	No
mail	VARCHAR(255)	255	No	No	No
group_id	INT4	4	No	Si	Si
foto_id	INT4	4	No	Si	No
created	DATE		No	No	No
modified	DATE		No	No	No

Tabla 29: Descripción de la Tabla Users
Elaborado por: Diego Poaquiza – Investigador

Fotousuarios					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
filename	VARCHAR(255)	255	No	No	No
dir	VARCHAR(255)	255	No	No	No
mimetype	VARCHAR(255)	255	No	No	No
filesize	INT4	4	No	No	No
user_id	int(11)	11	No	No	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 30: Descripción de la Tabla Fotousuarios
Elaborado por: Diego Poaquiza – Investigador

Groups					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	INT4	4	Si	No	Si
name	VARCHAR(100)	100	No	No	Si
created	DATE		No	No	No
modified	DATE		No	No	No

Tabla 31: Descripción de la Tabla Groups
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Dispositivos					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
nombre	VARCHAR(50)	50	No	No	No
tipo_id	int(11)	11	No	Si	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 32: Descripción de la Tabla Dispositivos
Elaborado por: Diego Poaquiza – Investigador

Tipos					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
nombre	VARCHAR(200)	200	No	No	No
descripcion	varchar(200)	200	No	No	No
codigo	VARCHAR(4)	4	No	No	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 33: Descripción de la Tabla Tipos
Elaborado por: Diego Poaquiza – Investigador

Eventos					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
zonificacion_id	int(11)	11	No	Si	No
user_id	int(11)	11	No	No	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 34: Descripción de la Tabla Eventos
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Zonas					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
nombre	VARCHAR(50)	50	No	No	No
descripcion	VARCHAR(100)	100	No	No	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 35: Descripción de la Tabla Zonas
Elaborado por: Diego Poaquiza – Investigador

Zonificacions					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
dispositivo_id	int(11)	11	No	Si	No
particione_id	int(11)	11	No	Si	No
trama	varchar(5)	5	No	No	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 36: Descripción de la Tabla Zonificacions
Elaborado por: Diego Poaquiza – Investigador

Particiones					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
nombre	varchar(200)	200	No	No	No
decripcion	varchar(200)	200	No	No	No
zona_id	int(11)	11	No	Si	No
created	DATE		No	No	No
modified	DATE		No	No	No
estado	int		No	No	No

Tabla 37: Descripción de la Tabla Particiones
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Sistemaestados					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
estado	int		No	No	No
created	date		No	No	No
modified	DATE		No	No	No

Tabla 38: Descripción de la Tabla Sistemaestados
Elaborado por: Diego Poaquiza – Investigador

Eventousuarios					
Nombre	Tipo Dato	Longitud	Primary Key	Foreign Key	Obligatorio
id	int(11)	11	Si	No	Si
fechahora	datetime		No	No	No
evento	int		No	No	No
created	date		No	No	No
modified	DATE		No	No	No

Tabla 39: Descripción de la Tabla Eventousuarios
Elaborado por: Diego Poaquiza – Investigador

4.2.2 Diseño de la Interfaz

El diseño de la Interfaz de Usuario se lo realizó cuidadosamente de tal manera que sea entendible y fácil de usar.

A continuación el diseño de las interfaces de las páginas que se utilizó para crear la Aplicación Web de monitoreo.

Interfaz Inicio de Sesión

Permite la autenticación del usuario y contraseña para ingresar a la aplicación de monitoreo, dependiendo de los permisos que se le haya sido otorgado se mostrará los sitio correspondientes.



Fig. 35: Interfaz de la Página Inicio de Sesión

Elaborado por: Diego Poaquiza – Investigador

Interfaz de la Página Maestra

La página maestra permite crear un diseño único para luego ser utilizada en las demás páginas que se vayan a diseñar ayudando así a optimizar el trabajo en la interfaz.

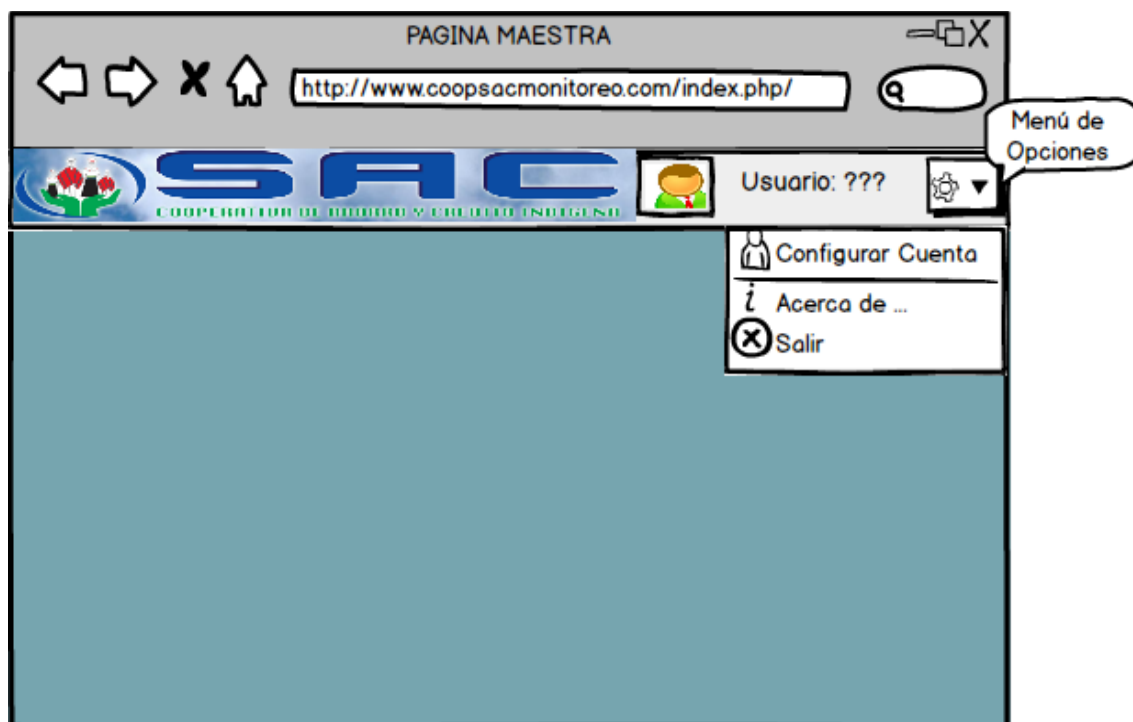


Fig. 36: Interfaz de la Página Maestra

Elaborado por: Diego Poaquiza – Investigador

Menú de opciones: Consta de tres ítems:

➤ **Configurar cuenta:**

Permite dirigir a la página respectiva de configuración de la información personal del usuario logueado.

➤ **Acerca de:**

Muestra un cuadro modal sobre la información de la aplicación web.

➤ **Salir:**

Permite salir de la aplicación.

Usuario: Se muestra la foto y el nombre de usuario logueado.

Interfaz configuración de la cuenta de usuario

En esta interfaz nos permite configurar la información de la cuenta del usuario logueado.



Fig. 37: Interfaz configuración de la cuenta de usuario

Elaborado por: Diego Poquiza – Investigador

Usuario: Se visualiza el nombre y la foto del usuario logueado.

Área de datos: Se muestra una lista de la información personal del usuario logueado.

Área actualización de la información: Se visualiza la información detallada de la fila seleccionada por el usuario para su respectiva actualización.

Botón Editar: Verifica la validación de datos y procede al guardado de los mismos.

Interfaz Administrador

En esta interfaz se muestra la vista del usuario administrador de la aplicación.

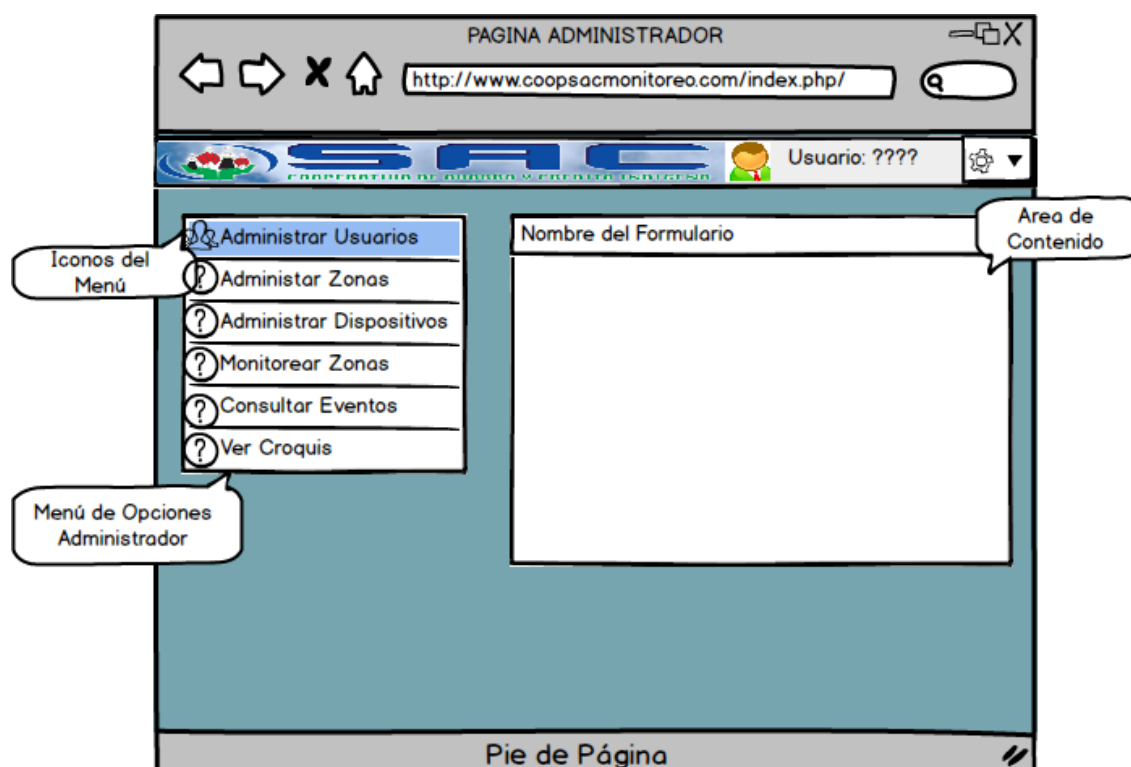


Fig. 38: Interfaz del Administrador

Elaborado por: Diego Poaquiza – Investigador

Menú de Opciones: Se muestra el menú de usuario administrador con sus respectivos iconos representativos respecto al menú.

Nombre del Formulario: En esta sección se muestra el nombre del formulario dependiendo de la opción del menú elegido.

Área de Contenido: Sección destinada para la **gestión de la información**. La información visualizada depende de las opciones del menú.

Interfaz Operador

En esta interfaz se muestra la vista del usuario operador de la aplicación, que será encargado del monitoreo del CPD mediante la aplicación.

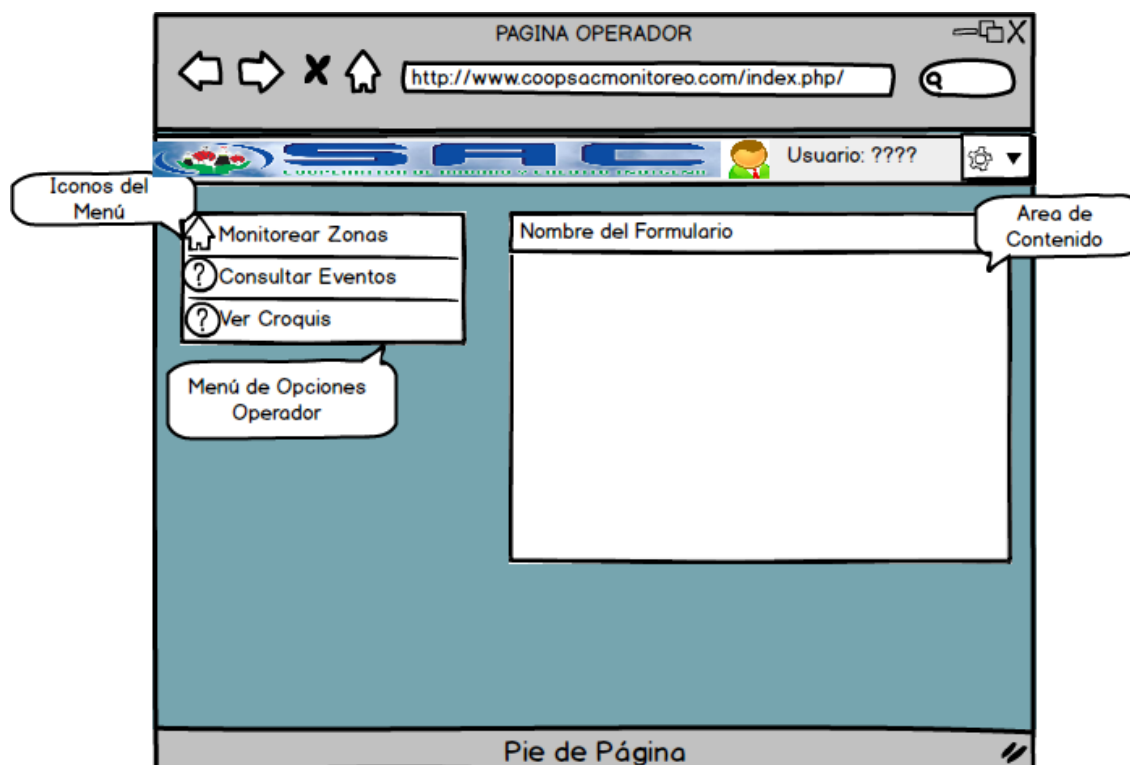


Fig. 39: Interfaz del Operador

Elaborado por: Diego Poaquiza – Investigador

Menú de Opciones: Se muestra el menú de usuario operador con sus respectivos íconos representativos respecto al menú.

Nombre del Formulario: En esta sección se muestra el nombre del formulario dependiendo de la opción del menú elegido.

Área de Contenido: Sección destinada para la **gestión de la información**. La información visualizada depende de las opciones del menú.

Interfaz Gestión Información

En esta sección se muestra la información de todos los datos registrados, respectivos a ese formulario.

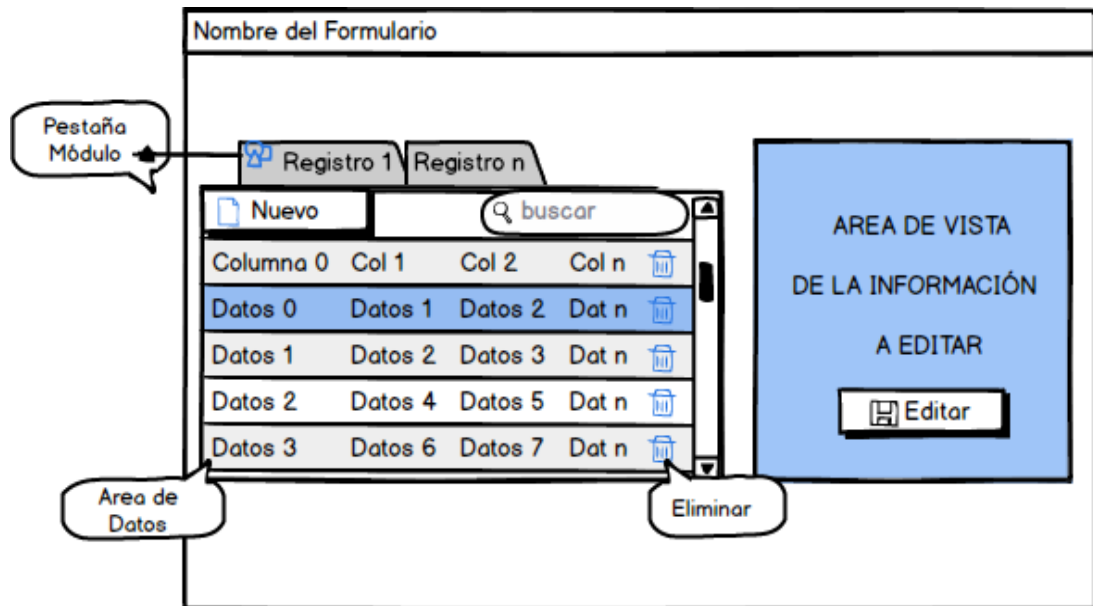


Fig. 40: Interfaz Gestión de Información
Elaborado por: Diego Poaquiza – Investigador

Pestaña Módulo: Se muestra las diferentes pestañas de acuerdo al formulario, depende de las opciones del menú elegido por el usuario.

Botón Nuevo: Muestra un cuadro modal para el ingreso de la información con sus respectivos controles.

Caja de búsqueda: Permite realizar una búsqueda de acuerdo al criterio del usuario, la caja de búsqueda estará disponible de acuerdo al formulario interactuado y cantidad de información.

Botón eliminar: Elimina el registro deseado, con su respectivo control.

Área de datos: Se muestra la lista de información respecto a la pestaña elegida.

Área de vista de la información a editar: Se visualiza la información detalla de la columna seleccionada por el usuario.

Editar: Muestra un cuadro modal con la información del registro seleccionado con sus respectivos controles para la edición de datos.

Ventana modal ingreso de información.

En esta interfaz nos permite el ingreso de un nuevo registro en base al formulario con el que se está interactuando.

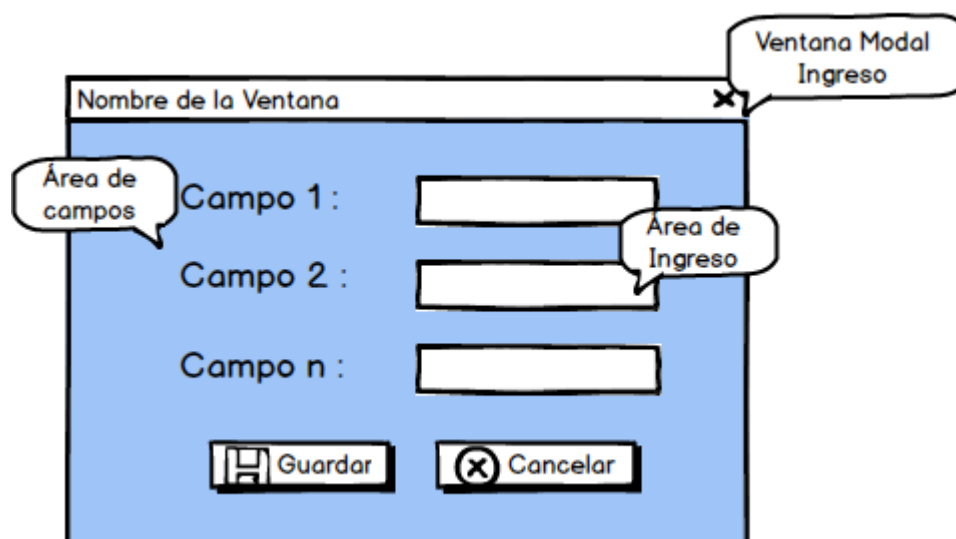


Fig. 41: Ventana modal ingreso de información
Elaborado por: Diego Poaquiza – Investigador

Nombre de la Ventana: Se muestra el nombre de la ventana, según el formulario con el que se está interactuando.

Área Campos: Contiene los campos de descripción de los cuadros de texto correspondientes al formulario interactuado.

Área de Ingreso: Contiene los cuadros de textos con sus respectivos controles para el ingreso de la nueva información.

Botón Guardar: Verifica la validación de datos y procede al guardado de los mismos.

Botón Cancelar: Cancela la inserción de nueva información cerrando la ventana modal.

Botón X: Cierra el cuadro modal sin realizar cambio alguno.

Ventana modal modificación de información.

En esta interfaz nos permite editar un registro almacenado en la base de datos en base al formulario con el que se está interactuando.

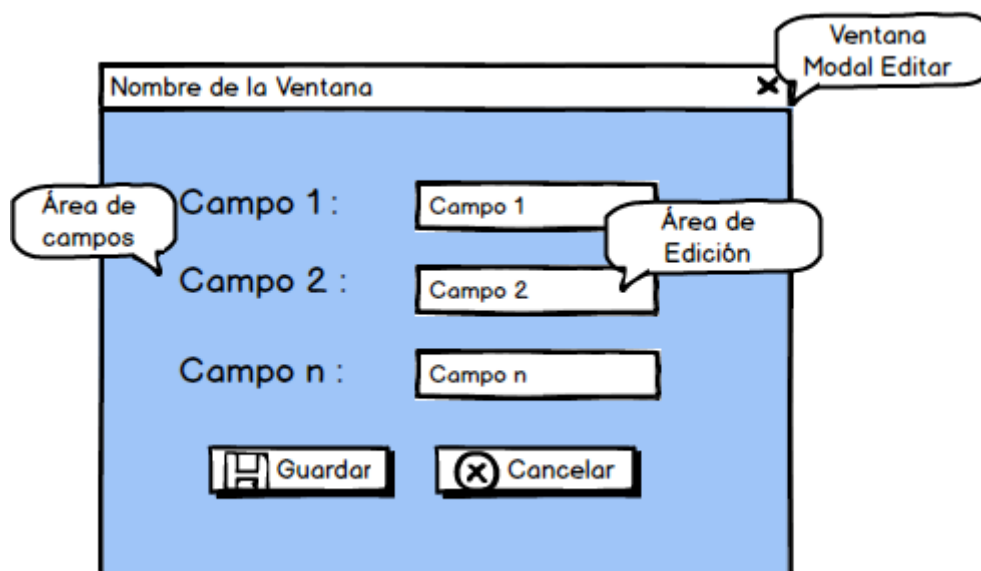


Fig. 42: Ventana modal modificación de información

Elaborado por: Diego Poaquiza – Investigador

Nombre de la Ventana: Se muestra el nombre de la ventana, según el formulario con el que se está interactuando.

Área Campos: Contiene los campos de descripción de los cuadros de texto correspondientes al formulario interactuado.

Área de Edición: Contiene los cuadros de textos con los datos correspondientes al formulario interactuado con sus respectivos controles para su respectiva edición de la información.

Botón Guardar: Verifica la validación de datos y procede a la actualización de la información de los datos editados en la base de datos.

Botón Cancelar: Cancela la inserción de nueva información cerrando la ventana modal.

Botón X: Cierra el cuadro modal sin realizar cambio alguno.

Cuadro de dialogo de Eliminación

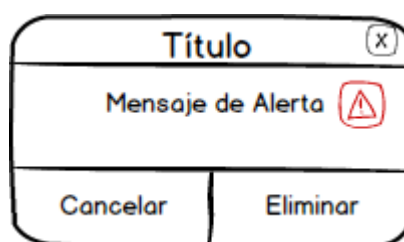


Fig. 43: Cuadro de dialogo de Eliminación

Elaborado por: Diego Poaquiza – Investigador

Título: Se muestra el nombre de la ventana, según el formulario con el que se está interactuando para su eliminación.

Mensaje de Alerta: Visualiza un mensaje de alerta del registro a eliminar de acuerdo al formulario con el que se interactúa.

Cancelar: Cancela la eliminación del registro cerrando la ventana modal.

Eliminar: Valida el registro a eliminar y procede a su eliminación de la base de datos.

Botón x: Cierra el cuadro de dialogo sin realizar cambio alguno.

Interfaz Monitoreo de Zonas

Permite al usuario realizar el monitoreo del Centro de Procesamiento de Datos mediante los sensores instalados y realizar la zonificación respectiva.

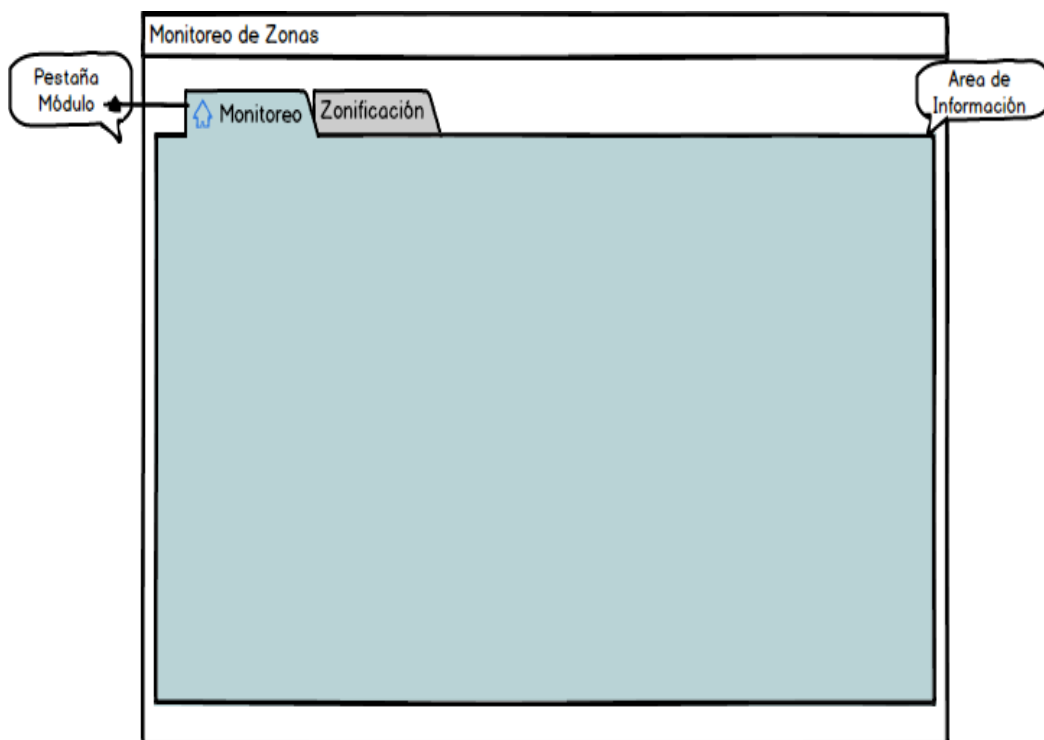


Fig. 44: Interfaz Monitoreo de Zonas

Elaborado por: Diego Poaquiza – Investigador

Pestaña Módulo: Se muestra las diferentes pestañas de acuerdo al formulario, depende de las opciones del menú elegido por el usuario.

Área de Información: Muestra los estados de cada uno de los dispositivos instalados en el Centro de Procesamiento de Datos.

Interfaz Monitoreo

Permite al usuario activar o desactivar el sistema de alarma y realizar el monitoreo del Centro de Procesamiento de Datos mediante los sensores instalados.



Fig. 45: Interfaz de Monitoreo

Elaborado por: Diego Poaquiza – Investigador

Pestaña Módulo: Se muestra el nombre de la pestaña a interactuar.

Botón estados: Muestra el estado actual del Sistema de Alarma, además permite activar o desactivar con dar click sobre el botón.

Área de datos: Se muestra la lista de información respecto a la pestaña elegida.

Imagen Dispositivo: Visualiza la imagen de los dispositivos instalados en el CPD, la imagen cambia dependiendo del estado y tipo de dispositivo.

Área Zonificación: Contiene la información de la zonificación de los dispositivos instalados en el CPD, mostrando la información de su distribución y estado, la información depende del tipo de dispositivo.

Interfaz Croquis

Permite al usuario realizar el monitoreo del CPD respectivo por medio de una cámara y un croquis el cual muestra el estado de mismo de acuerdo a la información arrojada por los sensores.

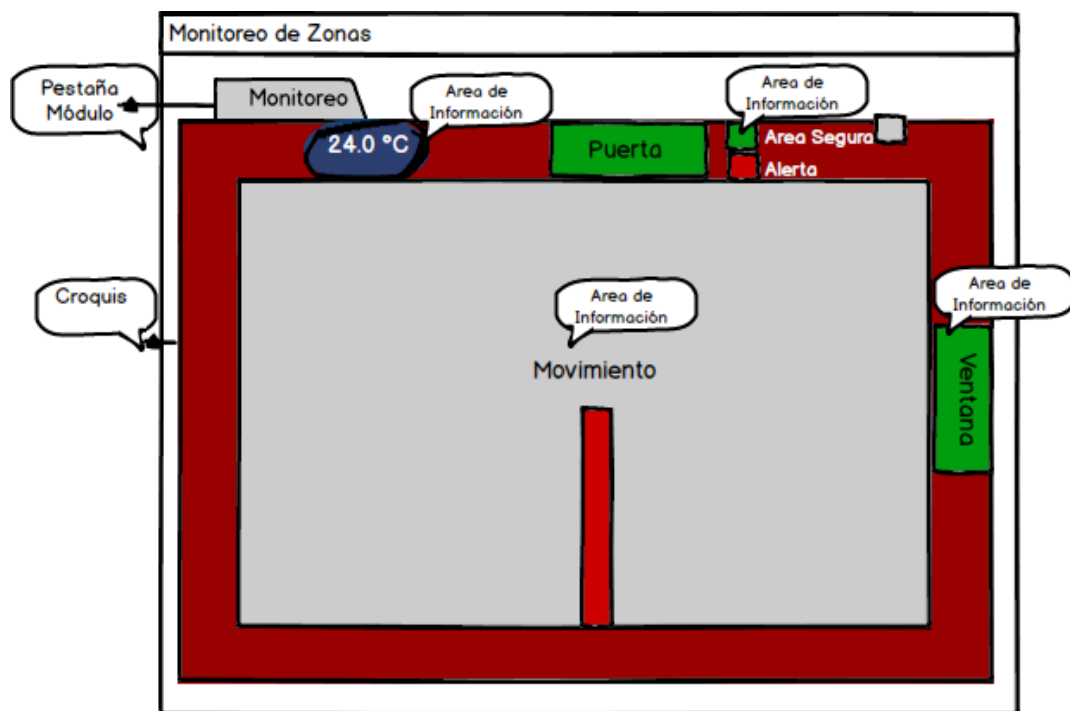


Fig. 46: Interfaz Monitoreo mediante Croquis

Elaborado por: Diego Poaquiza – Investigador

Pestaña Módulo: Se muestra el nombre de la pestaña a interactuar, además que al dar click sobre el mismo permite abrir una ventana emergente con la visualización de la imagen ofrecida por la cámara instalada en el CPD.

Croquis: Visualiza al usuario el croquis del CPD con la respectiva información de los dispositivos instalados.

Área de Información: Se muestra la lista de información con los estados de la información emitida por los respectivos sensores.

Interfaz Zonificación

En esta sección se muestra la información de la zonificación y permita al usuario la gestión de la misma de los dispositivos a instalarse en el CPD.

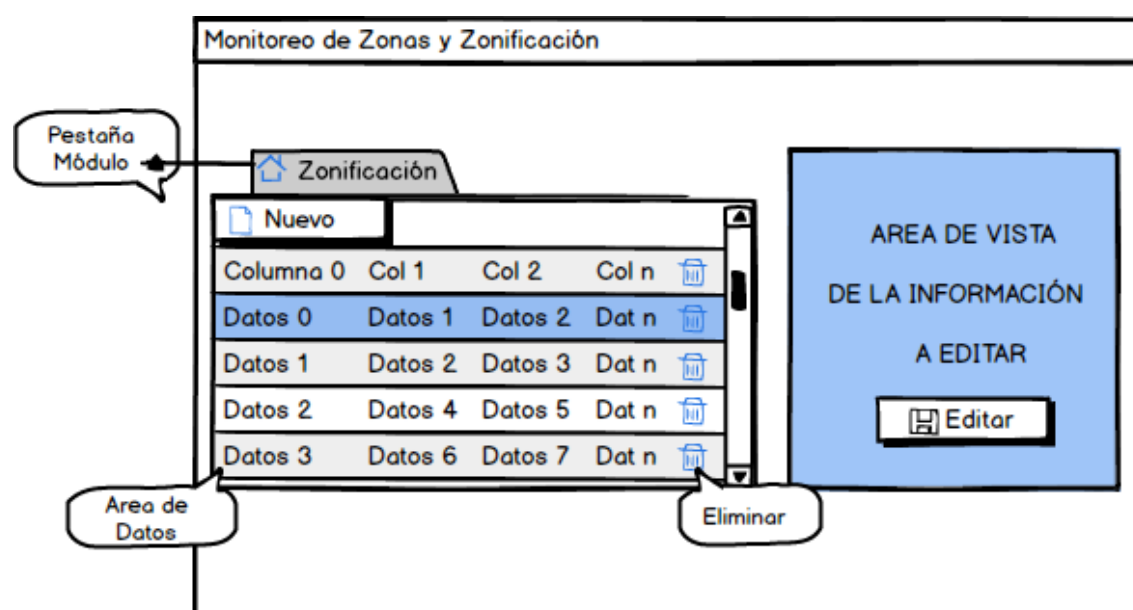


Fig. 47: Interfaz Zonificación

Elaborado por: Diego Poaquiza – Investigador

Pestaña Módulo: Se muestra el nombre de la pestaña a interactuar.

Botón Nuevo: Muestra un cuadro modal para el ingreso de la información con sus respectivos controles.

Botón eliminar: Elimina el registro deseado, con su respectivo control.

Área de datos: Se muestra la lista de información respecto a la pestaña elegida.

Área de vista de la información a editar: Se visualiza la información detalla de la columna seleccionada por el usuario.

Editar: Muestra un cuadro modal con la información del registro seleccionado con sus respectivos controles para la edición de datos.

Interfaz Consultar de Eventos

En esta sección se muestra la información de todos los eventos suscitados en los dispositivos como los eventos realizados por los usuarios.

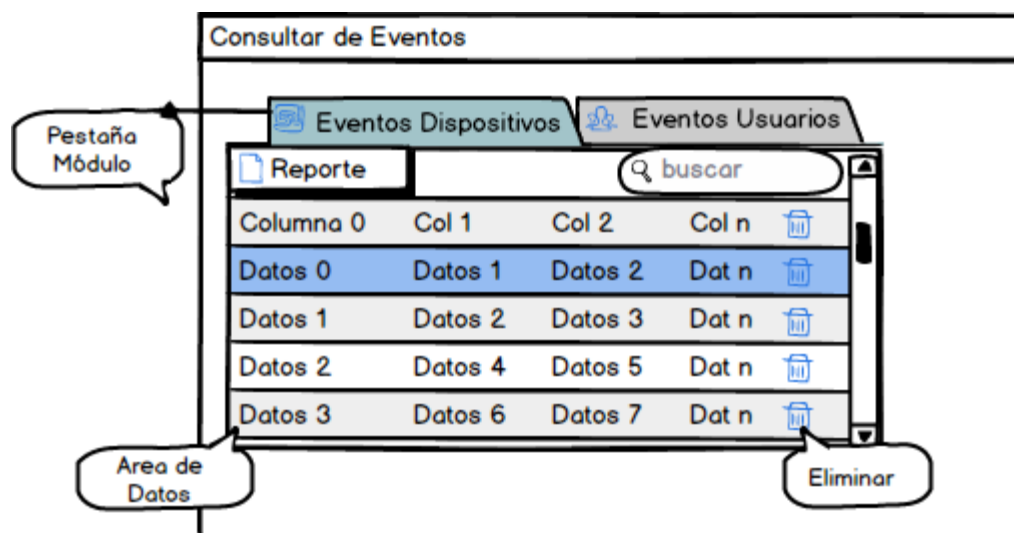


Fig. 48: Interfaz Consultar de Eventos
Elaborado por: Diego Poaquiza – Investigador

Pestaña Módulo: Se muestra las diferentes pestañas del formulario eventos.

Botón reporte: Muestra una ventana emergente, con el reporte de los datos filtrados en formato pdf.

Caja de búsqueda: Permite realizar una búsqueda de acuerdo al criterio del usuario.

Botón eliminar: Elimina el registro deseado, con su respectivo control.

Área de datos: Se muestra la lista de información respecto a la pestaña elegida.

Interfaz Reporte de Datos.

Permite visualizar en la ventana del navegador un documento de formato .pdf con los datos filtrados por la búsqueda realizada por el usuario, con las opciones de guardar e imprimir el documento generado.



Fig. 49: Interfaz Reporte de Datos
Elaborado por: Diego Poaquiza – Investigador

Título del Reporte: Muestra el nombre del reporte de acuerdo al formulario interactuado.

Información del Reporte: Se muestra un detalle de información del reporte.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Área de Datos: Visualiza el conjunto de registros previamente filtrados por la búsqueda realizada por el usuario.

Barra de Herramientas: Es la barra de navegador que permite realizar las acciones de impresión, guardado entre otras, dependiendo del navegador utilizado se mostrará la barra de tareas de diferente manera.

4.3 Implementación

En esta etapa se realiza la programación o implementación propiamente dicha del código fuente, los requerimientos del sistema y diseño la cual debe traducirse en una forma entendible y legible para la máquina.

Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

Antes de realizar la implementación de la Base de Datos, Interfaz Gráfica del Usuario, así como la Arquitectura de programación con el código fuente de la aplicación, se procedió a detallar la forma de capturar los datos que emiten los sensores con su implementación hardware y software, así como el análisis y la manera de tratar la información capturada por algún lenguaje de programación, información que será importante para el respectivo tratamiento y uso en la Aplicación web.

A continuación se detalla la forma de captura de datos que emiten los sensores y la captura de los mismos.

En primer lugar se procedió al análisis de los tipos de señales que emiten los diferentes sensores instalados en CPD.

Para su análisis se procedió a revisar las características de dichos dispositivos, verificando su modelo, tipo y fabricante.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Para una mayor comprensión se procedió a la revisión del datasheet respectivo de los dispositivos electrónicos a analizar el cual permitió saber sobre su funcionamiento, tipos de señales y demás características propias de los dispositivos.

Para la captura de los datos se realizó la implementación respectiva del hardware, software y comunicación que se detallan a continuación:

4.3.1 Implementación Hardware Arduino Uno

Se decidió utilizar un dispositivo de adquisición de datos denominado **Arduino Uno** debido a su costo económico, hardware libre, además de su potente utilización en el desarrollo de objetos interactivos ya que puede tomar información del entorno a través de sus entradas analógicas y digitales, controlar luces, motores, sensores y otros actuadores de manera factible.

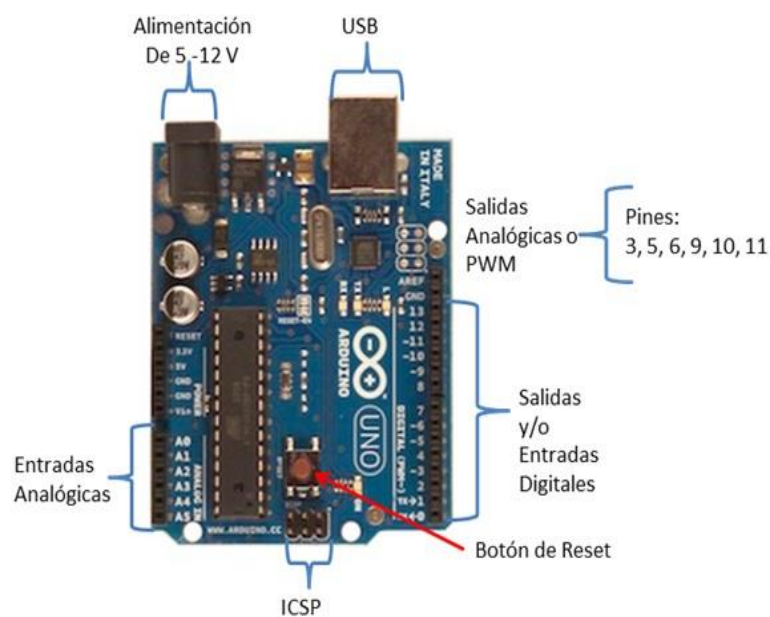


Fig. 50: Arduino Uno

Elaborado por: <http://www.tallerdecontrol.com/wp-content/uploads/2013/02/arduino5.jpg>

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Antes de proceder a la obtención de los datos emitidos por los dispositivos se procedió a conectar cada uno de los sensores a los pines respectivos de la placa sean estos analógicos o digitales dependiendo de los requerimientos necesarios para comunicarnos con el sistema de desarrollo IDE.

Pines digitales de entrada y salida

En estos pines conectamos la patilla de dato del Sensor. Desde ellos podremos leer la información del Sensor. Hay 14 pines digitales que pueden utilizarse como entrada o salida.

Pines analógicos

Esta placa contiene 6 pines de entrada analógicos. Los elementos que se conecten aquí suelen tener mayor precisión que los digitales pero su uso requiere de una lógica levemente mayor.

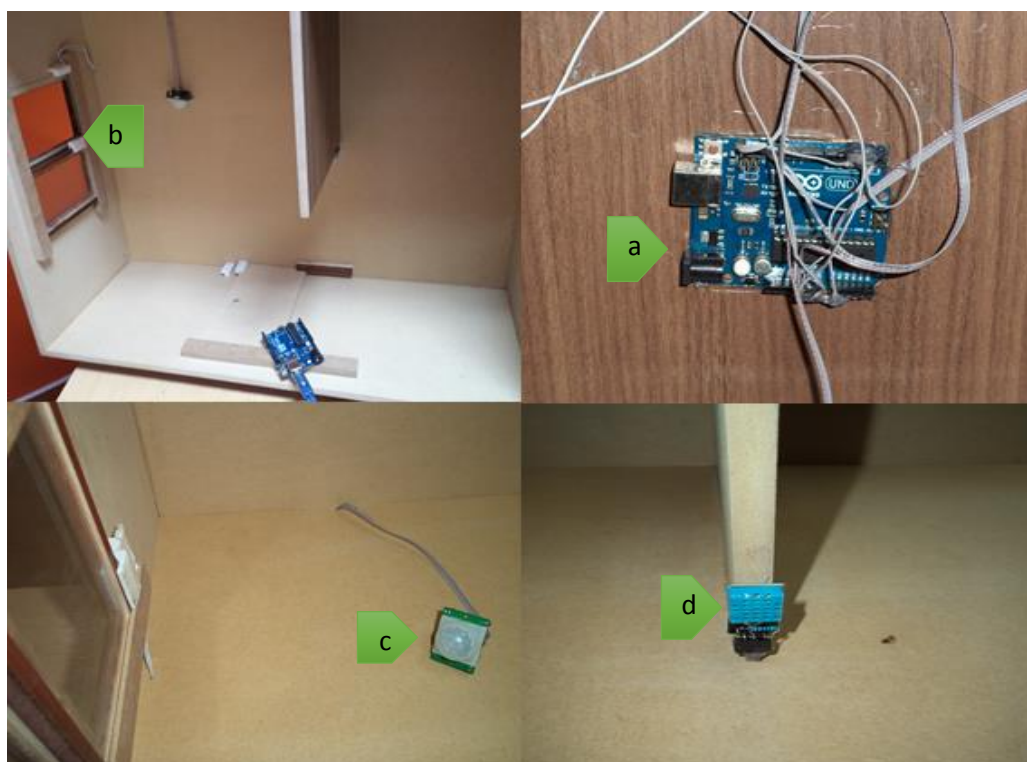


Fig. 51: Arduino con Sensores

Elaborado por: Diego Poaquiza – Investigador

- a) **Arduino Uno.**
- b) **Sensor de Contacto 943.**
- c) **Sensor de Movimiento PIR DYP-ME003.**
- d) **Sensor de Temperatura DHT11.**

4.3.2 Implementación Software Arduino Uno

Una vez determinado la forma de captura de datos se procedió a la instalación del entorno de desarrollo integrado (IDE) cuyo entorno de desarrollo están disponibles para Windows, Mac y Linux, que sirve para la programación respectiva del Arduino encargada de leer la información de los sensores conectados a la placa, para posteriormente ser enviada a través del puerto serie y pueda ser leída e interpretada para su respectivo tratamiento por cualquier pc.

La instalación y configuraciones del entorno de desarrollo ID de la Tarjeta Electrónica Programable Arduino Uno se encuentran en el **Anexo 2 (Página 131-132)**.

Entorno de Desarrollo IDE

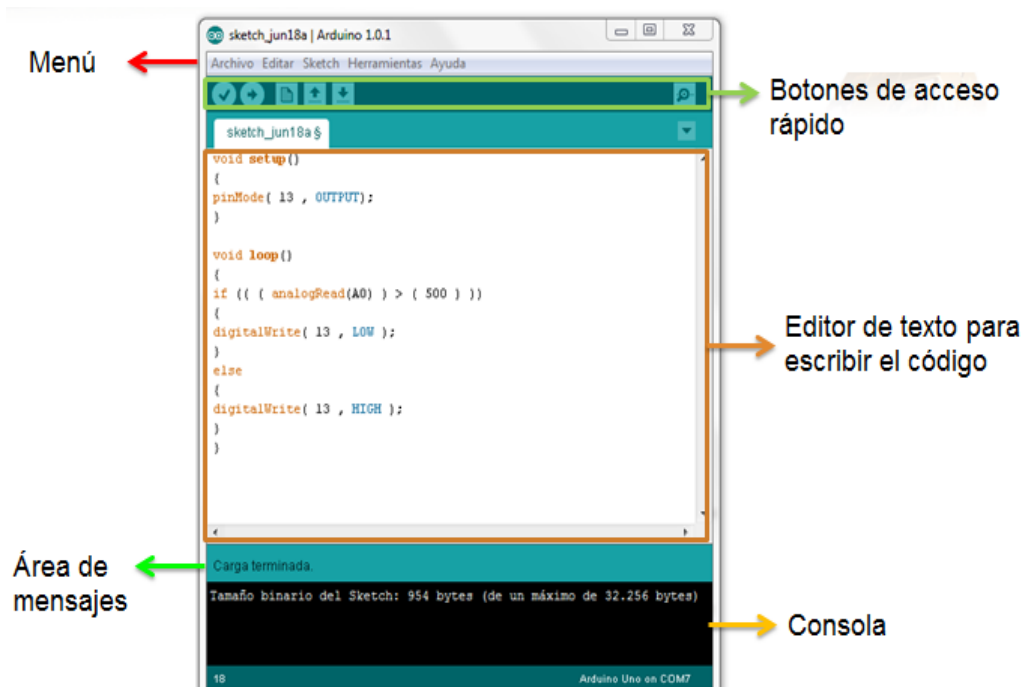


Fig. 52: Entorno de Desarrollo IDE

Elaborado por: Diego Poquiza

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Una vez instalado el entorno de desarrollo IDE propio del Arduino Uno se procedió a la programación para la obtención de la información emitidos por los sensores conectados a los pines de la placa, posteriormente dicha programación fue cargada a la memoria del Arduino Uno para su respectivo proceso de obtención y envío de datos por el puerto serie, cuyo programa correspondiente a la obtención de los datos se encuentra en el **Anexo 3 (Página 133-134)**.

Comunicación Serial con Arduino Uno

Para realizar la comunicación, Arduino Uno posee como principal característica la habilidad de comunicarse con la computadora a través del puerto serie. Esto se conoce como comunicación serial. Debido a que el uso de este puerto ha quedado un poco en desuso a favor de la tecnología USB, Arduino cuenta con un convertidor de Serial a USB que permite a la placa ser reconocida por nuestra computadora como un dispositivo conectado a un puerto COM incluso cuando la conexión física sea mediante USB.

Análisis de la información capturada que emiten los sensores con el fin de utilizar dicha información para su respectivo uso y procesamiento de la misma.

Una vez capturados los datos de los sensores conectados a los pines de la placa, mediante programación realizada en el entorno de desarrollo integrado (IDE) se procedió al análisis y procesamiento de la información llegando a obtener los siguientes resultados:

Los datos arrojados por los sensores una vez cargada la respectiva programación a la memoria del Arduino, para la obtención de datos se procedió a la lectura del puerto serie mediante el entorno de desarrollo IDE propio del Arduino obteniendo información binaria tanto para el **Sensor de Movimiento PIR DYP-ME003 y Contacto Magnético modelo 943** representada a continuación.

Código binario natural

- 1** **Sensor de movimiento (se interpreta como estado Activo)**
- 0** **Sensor de movimiento (se interpreta como estado Inactivo)**
- 1** **Contacto Magnético ventana (se interpreta como estado Activo)**
- 0** **Contacto Magnético ventana (se interpreta como estado Inactivo)**
- 1** **Contacto Magnético puerta (se interpreta como estado Activo)**
- 0** **Contacto Magnético puerta (se interpreta como estado Inactivo)**

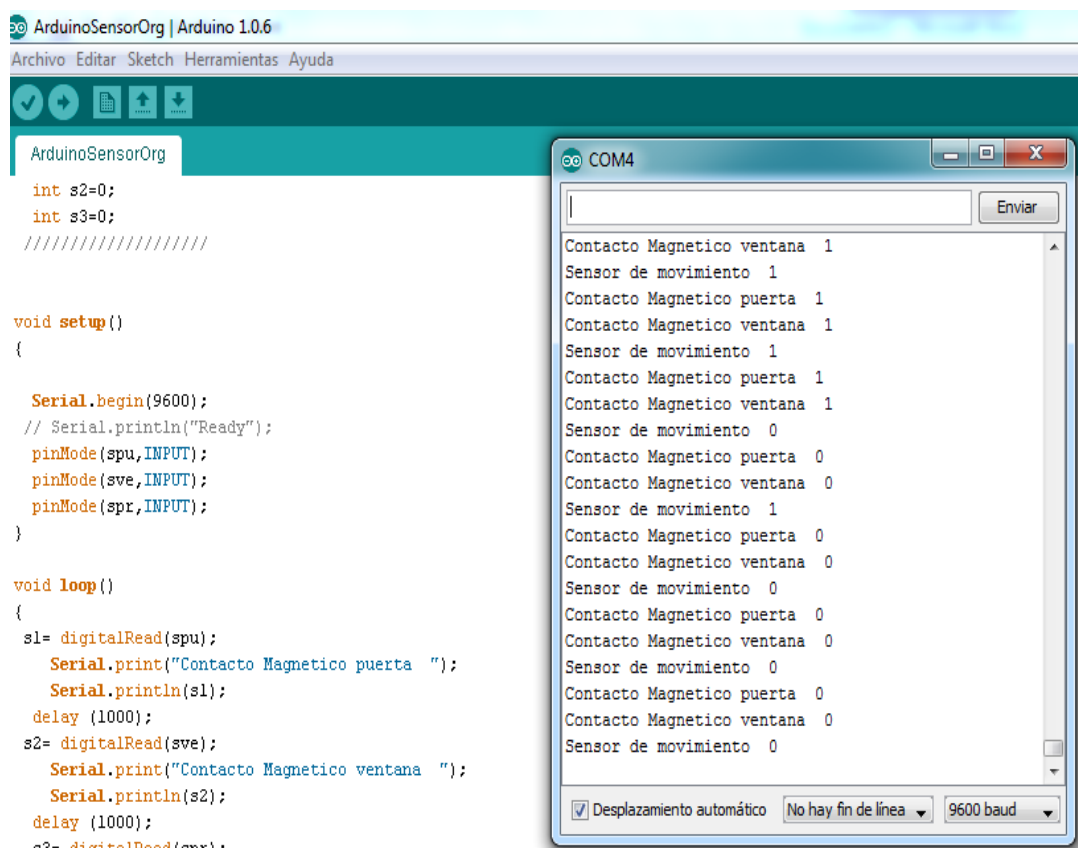


Fig. 53: Información capturada sensores de contacto y movimiento

Elaborado por: Diego Poaquiza – Investigador

Estos valores varían en un determinado tiempo dependiendo del estado del dispositivo.

Para el sensor **Temperatura y Humedad DHT11** se obtuvo Código decimal binario representado a continuación.

Temperatura 0111 0011 0010

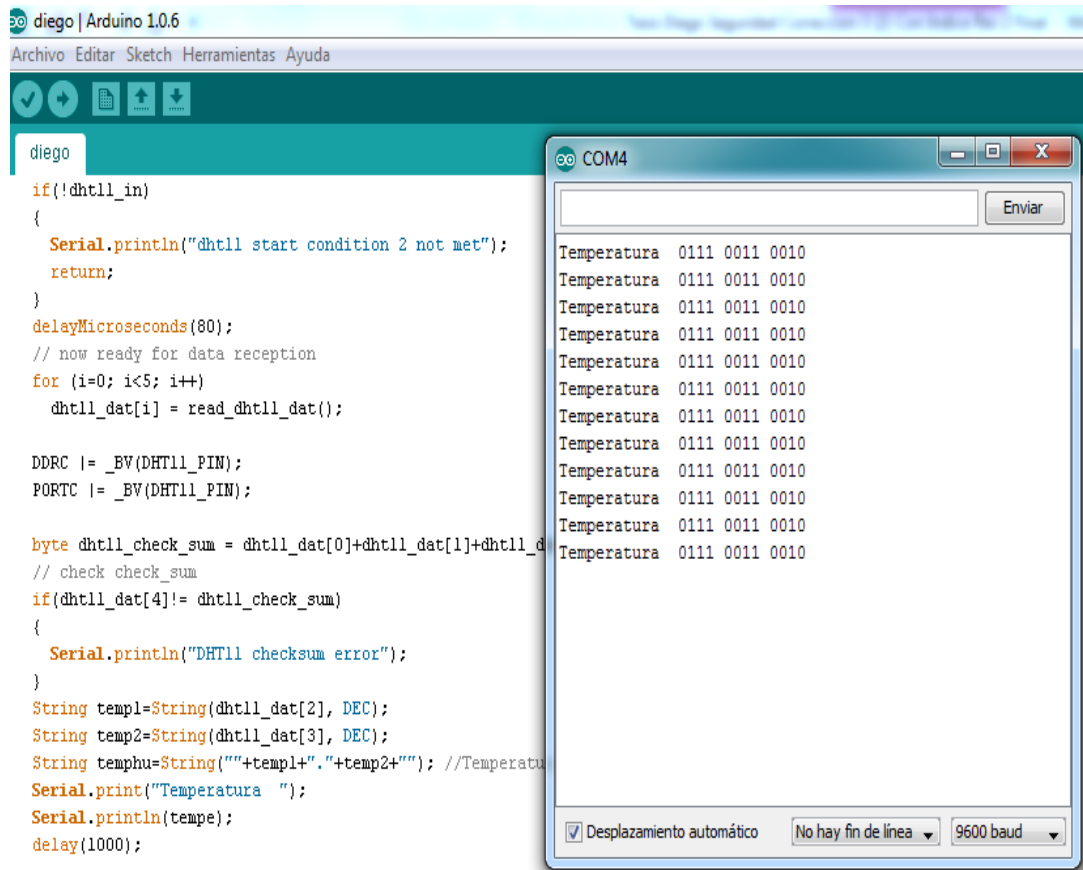


Fig. 54: Información capturada sensor de temperatura

Elaborado por: Diego Poaquiza – Investigador

Para una mayor comprensión de la información sobre la temperatura se procedió, en la programación del Arduino haciendo uso de la librería **DHT11_PIN** para el sensor de temperatura y humedad que es la encargada de transformar la información decimal binario a datos más entendibles dando como resultado.

Temperatura 21,0

```

diego | Arduino 1.0.6
Archivo Editar Sketch Herramientas Ayuda

diego

DDRC |= _BV(DHT11_PIN);
PORTC |= _BV(DHT11_PIN);

byte dht11_check_sum = dht11_dat[0]+dht11_dat[1]+dht11_dat[2]+dht11_dat[3];
// check check_sum
if(dht11_dat[4] != dht11_check_sum)
{
  Serial.println("DHT11 checksum error");
}
String temp1=String(dht11_dat[2], DEC);
String temp2=String(dht11_dat[3], DEC);
String tempuhu=String(temp1+"."+temp2); //Temperatura

s1= digitalRead(spu); //Puerta
s2= digitalRead(sve); //Ventana
s3= digitalRead(spr); //Prescencia

//Serial.println(tempuhu);
Serial.println(s1);
Serial.println(s2);
Serial.println(s3);
delay(1000);
  
```

COM4

```

Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
Temperatura 21.0
  
```

Desplazamiento automático

Fig. 55: Información temperatura tratada
 Elaborado por: Diego Poaquiza – Investigador

Cabe aclarar que toda la información obtenida mediante el puerto serie, se lo obtiene en modo texto para lo cual será necesario tratar la información para su respectivo uso y tratamiento en algún lenguaje de programación.

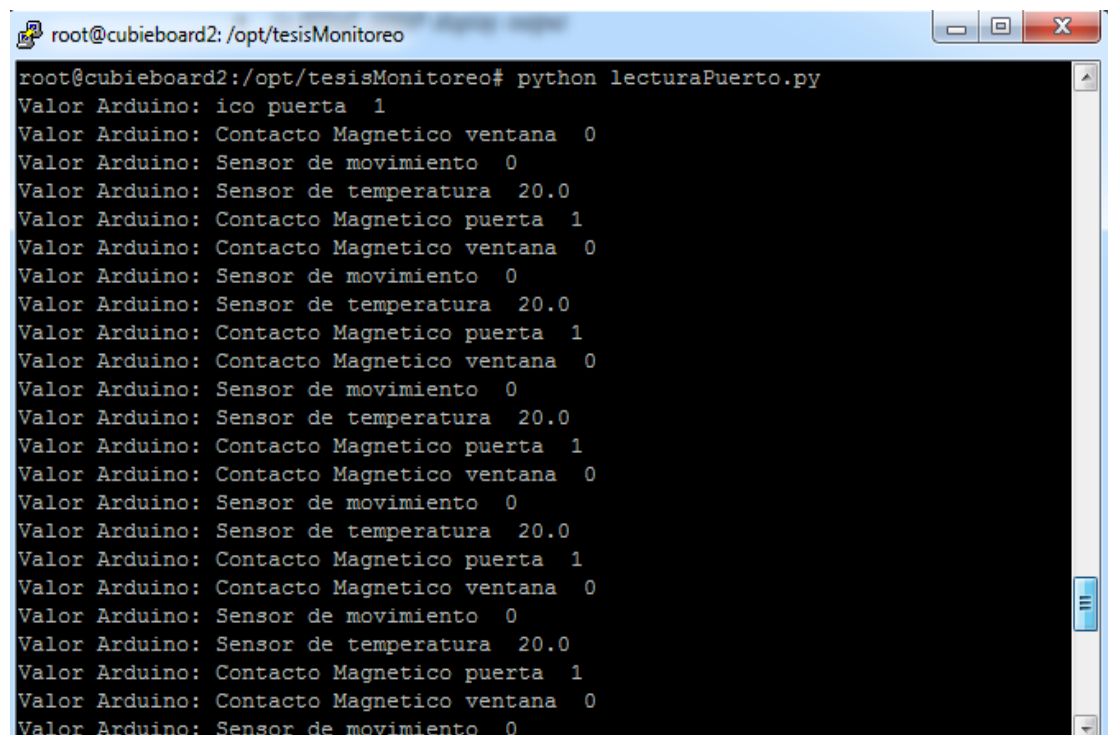
Una vez realizado el análisis de la información arrojada por los sensores se establece a continuación la manera de tratar los datos obtenidos con el fin de que dicha información pueda ser interpretada por algún lenguaje de programación.

Como se explicó anteriormente para el proceso de obtención y envío de datos por el puerto serie se realizó mediante la respectiva programación que se encuentra en el **Anexo 3**.

Para obtener los datos que envía el Arduino Uno por el puerto serie desde la PC se realizó la conexión con la computadora mediante la **Comunicación Serial con Arduino Uno** donde puede ser leída por algún lenguaje de programación instalada.

Lectura del puerto serie desde pc

Como Pc se utiliza la cubieboard A20, una vez hecha la conexión con la pc, se procedió a la lectura del puerto serie mediante el software de programación Python, dicha programación correspondiente se encuentra en el **Anexo 4 (Página 135)**, obteniendo como resultado.



```
root@cubieboard2: /opt/tesisMonitoreo
root@cubieboard2:/opt/tesisMonitoreo# python lecturaPuerto.py
Valor Arduino: ico puerta 1
Valor Arduino: Contacto Magnetico ventana 0
Valor Arduino: Sensor de movimiento 0
Valor Arduino: Sensor de temperatura 20.0
Valor Arduino: Contacto Magnetico puerta 1
Valor Arduino: Contacto Magnetico ventana 0
Valor Arduino: Sensor de movimiento 0
Valor Arduino: Sensor de temperatura 20.0
Valor Arduino: Contacto Magnetico puerta 1
Valor Arduino: Contacto Magnetico ventana 0
Valor Arduino: Sensor de movimiento 0
Valor Arduino: Sensor de temperatura 20.0
Valor Arduino: Contacto Magnetico puerta 1
Valor Arduino: Contacto Magnetico ventana 0
Valor Arduino: Sensor de movimiento 0
Valor Arduino: Sensor de temperatura 20.0
Valor Arduino: Contacto Magnetico puerta 1
Valor Arduino: Contacto Magnetico ventana 0
Valor Arduino: Sensor de movimiento 0
```

Fig. 56: Lectura del puerto serie mediante Python

Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Para una mayor facilidad y tratamiento de los datos enviados por dicho puerto, así como poder utilizar dicha información en la aplicación web, se procedió a realizar un respectivo tratamiento de la información para que pueda ser leída de una manera tratable por el lenguaje de programación Python y proceder a su respectivo uso, procesamiento y análisis de la misma.

Como se indicó anteriormente la información recibida por dicho puerto es en forma de texto ya sea binaria o decimal, por lo que se procedió a generar una trama que me permite definir de una manera clara el nombre del microcontrolador (Arduino Uno) donde están conectados los sensores, fecha, descripción y los dispositivos cuya información tendrá el nombre del dispositivo, el tipo, el valor y el estado, así poder saber el tipo de dispositivo que se encuentra conectado y su información emitida que será necesario para su respectivo uso en la aplicación Web.

Para su respectivo tratamiento se procedió armar la trama respectiva en la programación del Arduino una cadena que contenga un formato tipo JSON para el intercambios de datos basado en texto plano, cuyo formato JSON básicamente describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos.

Cabe recalcar que una de las mayores ventajas que tiene el uso de JSON (JavaScript Object Notation) es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías, al ser independiente de cualquier lenguaje de programación, los servicios que comparten información por éste método, no necesitan hablar el mismo idioma, es decir, el emisor puede ser Java y el receptor PHP etc., cada lenguaje tiene su propia librería para codificar y decodificar cadenas de JSON.

Es por eso que se decidió utilizar la notación JSON para el intercambio de información en texto plano entre el lenguaje de programación Arduino y Python.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Armado de la cadena Json en Arduino.

```
[{
  'idArduino':'MDGmonitoreo1',
  'date':'14/11/2014',
  'descripcion': 'temperatura, contacto, movimiento',
  'dispositivos': [
    {'nombre':'MDG01movimiento','tipo_id':'1','estado':'1','valor':''},
    {'nombre':'MDG01temperatura','tipo_id':'2','estado':'1','valor':''},
    {'nombre':'MDG01Contacto','tipo_id':'3','estado':'1','valor':''},
    {'nombre':'MDG01Contacto2','tipo_id':'3','estado':'1','valor':''}
  ]
}]
```

El código para la obtención de la información de los dispositivos y armado de la cadena JSON en el lenguaje de programación IDE del Arduino se encuentra en el **Anexo 5 (Página 135-136-137)**.

A continuación se muestra la salida de la información en formato JSON.

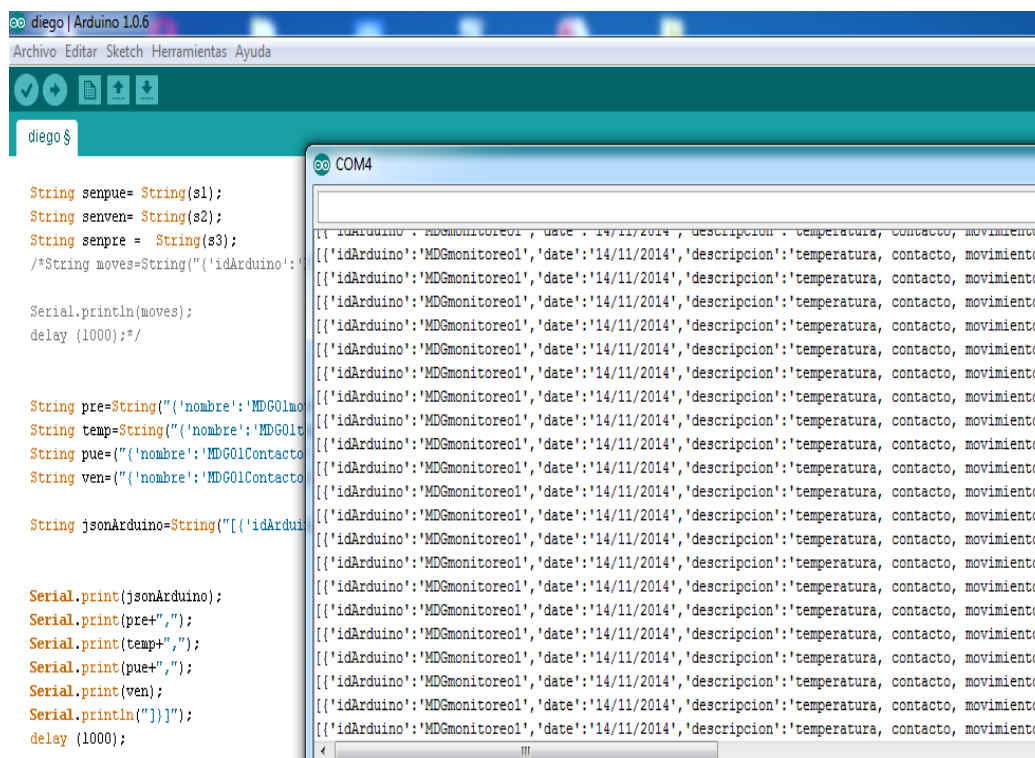


Fig. 57: Obtención de la información formato JSON
Elaborado por: Diego Poaquiza – Investigador

Una vez realizado el armado de la trama que contenga una cadena de formato de tipo JSON para los intercambios de datos se procedió a la lectura del puerto serie, mediante el lenguaje de programación Python se obtuvo los datos enviados mediante Arduino lo cual se realizó la decodificación de la cadena JSON mediante la propia librería propia de Python y así poder usar la información como variables tratables para la Aplicación Web.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

El código en el lenguaje de programación Python encarga de la lectura del puerto serie, obtención de la información y tratamiento de la misma para su respectivo uso, procesamiento y análisis se encuentra en el **Anexo 6 (Página 137-138-139)**.

4.3.3 Implementación de la Base de datos, Interfaz de usuario y Código fuente de la Aplicación Web

Implementación de la Base de Datos

La implementación de la base de datos se realizó mediante el sistema gestor de base de datos Mysql Workbench 6.1 cuya implementación se encuentra en el **Anexo 7 (Página 140-141-142)**.

Implementación de la Interfaz Gráfica del Usuario

A continuación se muestra algunas de las páginas de la aplicación web implementadas, las mismas que fueron desarrolladas en la etapa de diseño.

Página de Inicio de Sesión.

Interfaz de Inicio de Sesión para los diferentes usuarios registrados en la aplicación web, con sus respectivos privilegios.

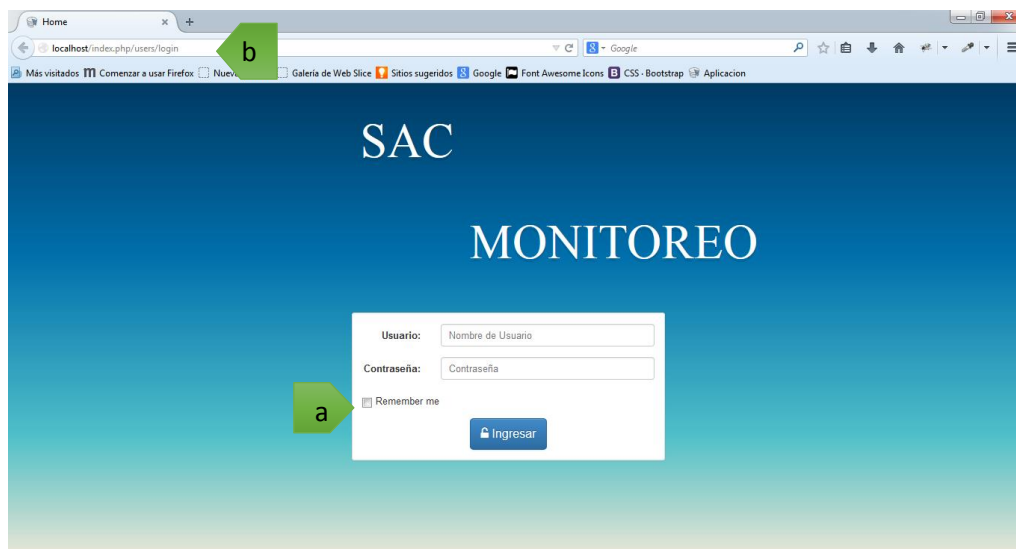


Fig. 58: Implementación Interfaz Inicio de Sesión
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

a) **Inicio de Sesión:** Consta de controles necesarios para el ingreso y validación de los datos del usuario de la aplicación.

b) **Navegador:** Muestra la url de navegación de la Aplicación Web de monitoreo.

La programación del botón de Ingreso a la aplicación web se describe a continuación.

```
public function login() {
    if ($this->request->is('post')) {
        if ($this->Auth->login()) {
            $parametros = array('data'=>array(
                'Eventousuario'=>array(
                    'fechahora'=>date("Y-m-d H:i:s",time()),
                    'evento'=>'6',
                    'user_id'=>$this->Auth->user('id')
                ),
            );
            $this->requestAction(array('controller'=>'eventousuarios','action'=>'add'),$parametros);
            $this->redirect(array('controller' => 'aplicacion', 'action' => 'index'));
        }
        $this->Session->setFlash(__('Usuario O contraseña Invalido'), 'alert', array(
            'plugin' => 'BoostCake',
            'class' => 'alert-danger'
        ));
        $this->redirect(array('controller' => 'users', 'action' => 'login'));
    }
}
```

Página Principal Administrador.

Contiene el menú del administrador permitiendo realizar todas las tareas correspondientes en la aplicación web.

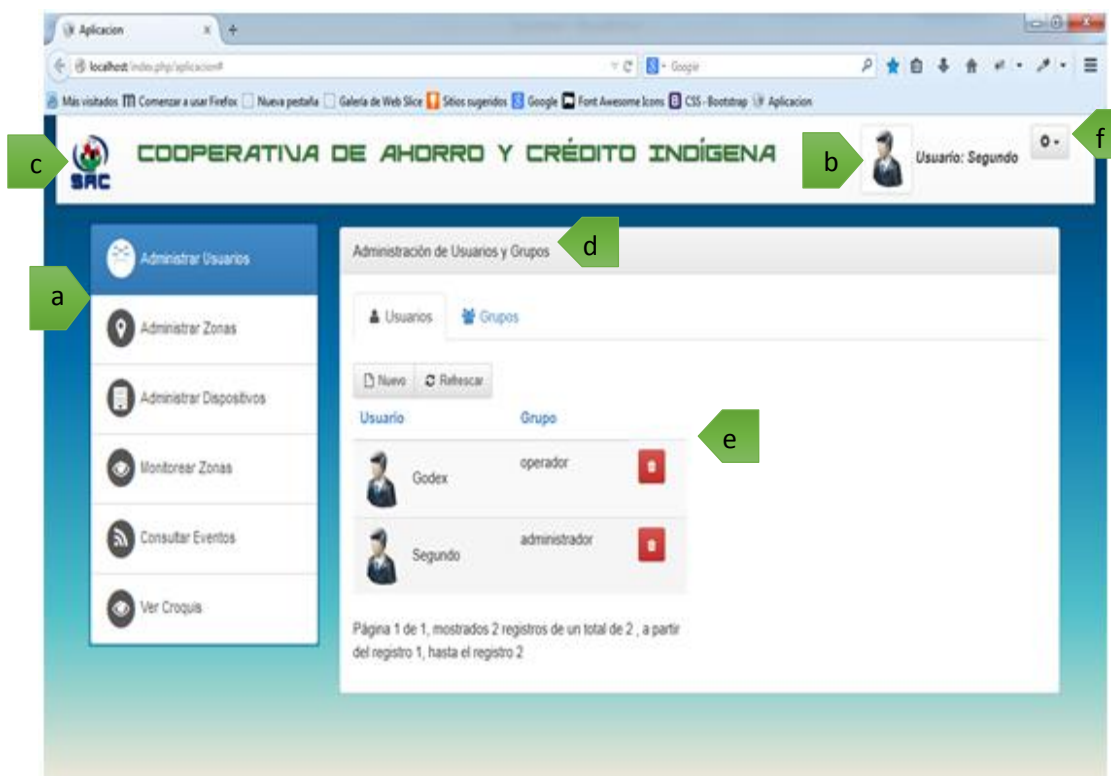


Fig. 59: Implementación Interfaz Página Administrador

Elaborado por: Diego Poaquiza – Investigador

- a) **Menú de Opciones:** Permite ingresar a los diferentes formularios para su respectiva gestión.
- b) **Usuario:** Muestra la foto del usuario y el nombre del usuario logueado.
- c) **Logo:** Visualiza el emblema de la Cooperativa en la interfaz.
- d) **Formularios:** En esta sección se muestra el nombre del formulario dependiendo de la opción del menú elegido.

- e) **Área de Contenido:** Sección destinada para la gestión de la información. La información visualizada depende de las opciones del menú.
- f) **Configurar cuenta:** Permite dirigir a la página respectiva de configuración de la información personal, acerca del sistema así como muestra la opción de cerrar la sesión del usuario.

Página Monitoreo de Zonas.

Permite al usuario activar o desactivar el sistema de alarma y realizar el monitoreo del Centro de Procesamiento de Datos mediante los sensores instalados.

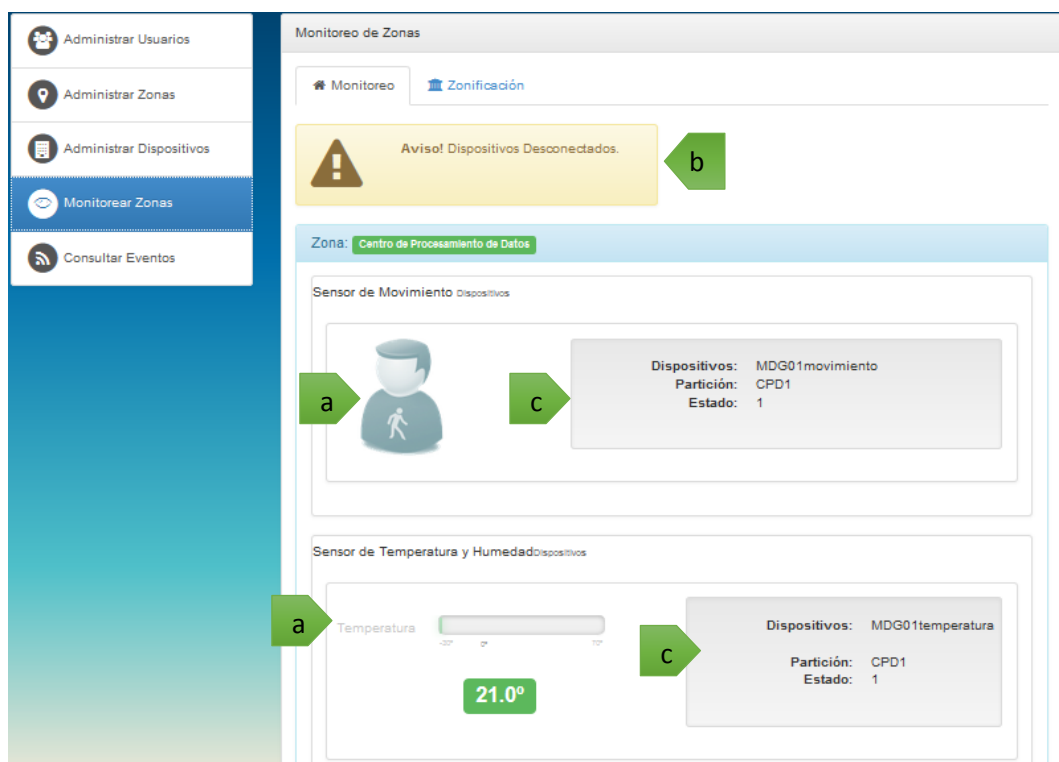


Fig. 60: Implementación Página Monitoreo de Zonas

Elaborado por: Diego Poaquiza – Investigador

- a) **Área Imagen Dispositivo:** Visualiza la imagen de los dispositivos instalados en el CPD, la imagen cambia dependiendo del estado y tipo.

b) Estados: Muestra el estado actual del Sistema de Alarma, además muestra un botón que permite activar o desactivar el sistema siempre y cuando los dispositivos estén conectados al sistema.

c) Área de información: Contiene la información de la zonificación de los dispositivos instalados en el CPD, mostrando la información de su distribución y estado, la información depende del tipo de dispositivo.

Página Monitoreo de zonas mediante Croquis y Cámara web.

Permite al usuario realizar el monitoreo del CPD respectivo por medio de una cámara y un croquis el cual muestra el estado de mismo de acuerdo a la información arrojada por los sensores.

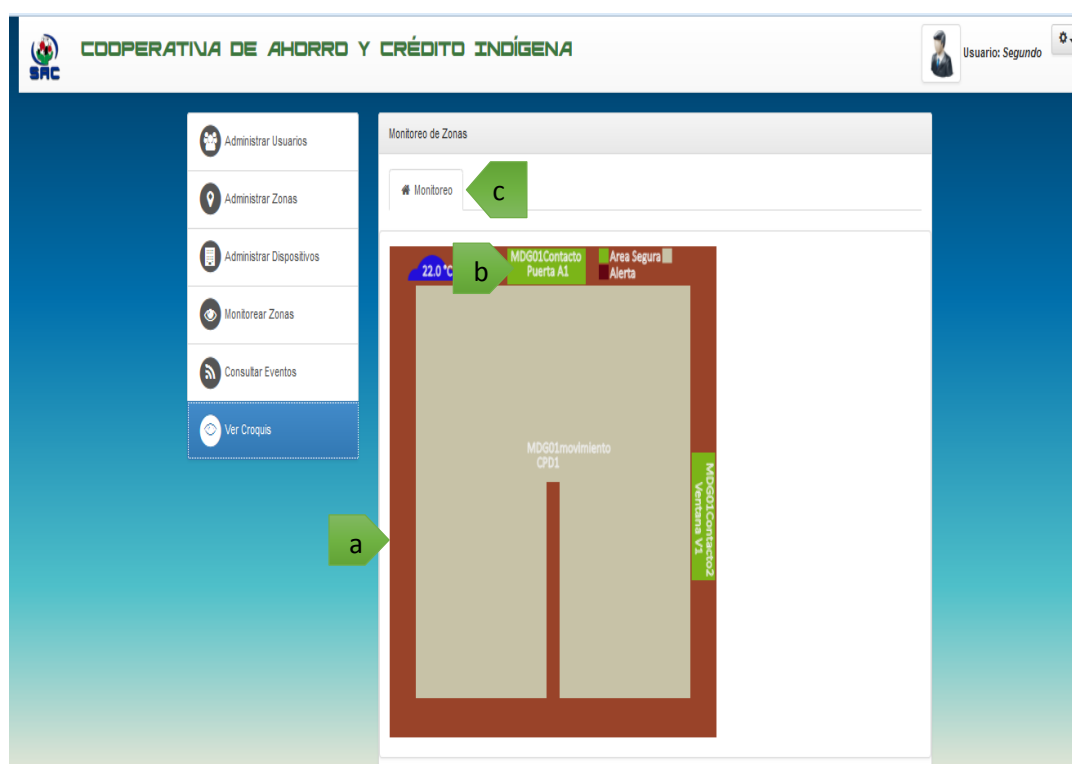


Fig. 61: Implementación Página Monitoreo de zonas mediante Croquis y Cámara web.

Elaborado por: Diego Poaquiza – Investigador

a) Croquis: Visualiza al usuario el croquis del CPD con la respectiva información de los dispositivos instalados.

b) Área de Información: Se muestra la lista de información con los estados de la información emitida por los respectivos sensores.

c) Pestaña Módulo: Se muestra el nombre de la pestaña a interactuar, además que al dar click sobre el mismo permite abrir una ventana emergente con la visualización de la imagen ofrecida por la cámara instalada en el CPD.

Ventana emergente con la información emitida por la Cámara web.

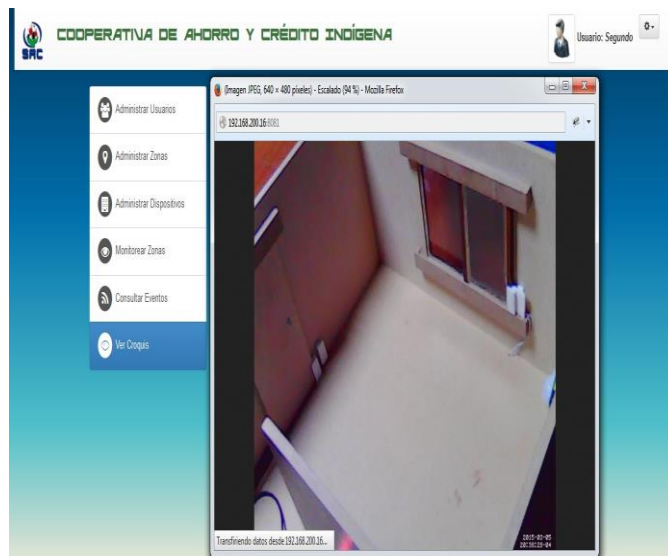


Fig. 62: Implementación ventana emergente cámara web
Elaborado por: Diego Poaquiza – Investigador

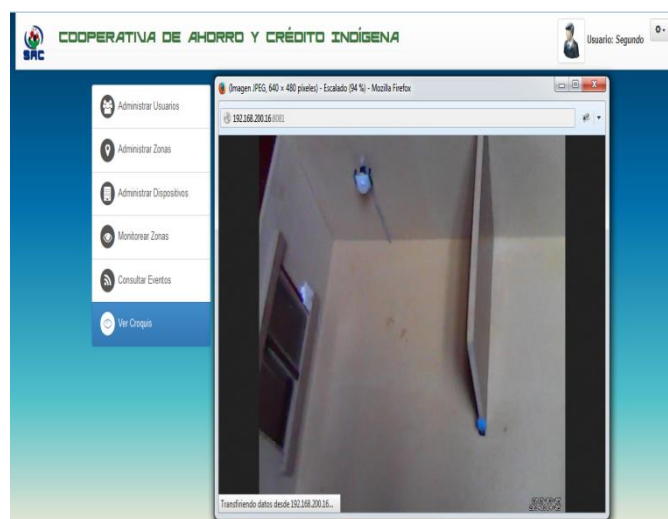


Fig. 63: Implementación ventana emergente cámara web
Elaborado por: Diego Poaquiza – Investigador

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Implementación de la Arquitectura de programación

La arquitectura MVC utilizada en la programación para el desarrollo de la aplicación web divide a la aplicación en 3 niveles de atracción Modelo, Vista y Controlador, esta arquitectura de software es la encargada de separar la lógica de negocio de la interfaz de usuario.

El Modelo es la encargada de Acceso de forma directa a los datos actúa como intermediario con la Base de Datos.

La Vista es la encargada de mostrar la información enviado por el Controlador al Usuario de una forma humanamente legible y entendible.

El Controlador es el intermediario entre el Modelo y la Vista es el encargado de controlar las interacciones del Usuario, solicitando los datos al Modelo y entregándolos a la Vista para que esta lo presente al usuario.

A continuación se presenta una parte del código de la aplicación de cada uno de los niveles de la tabla Zonas de la Base de Datos Monitoreo.

Modelo Zona

```

<?php
App::uses('AppModel', 'Model');
class Zona extends AppModel {
    public $validate = array(
        'nombre' => array(
            'notEmpty' => array(
                'rule' => array('notEmpty'),
                'message' => 'Ingrese el nombre de la zona',
            ),
            'alphaNumeric' => array(
                'rule' => array('custom', '/^[a-zA-Z[:space:]]*$/'),
                'required' => true,
                'message' => 'Campo nombre sólo letras'
            ),
            'between' => array(
                'rule' => array('between', 5, 50),
                'message' => 'Entre 5 y 50 caracteres'
            )
        ),

        'estado' => array(
            'notEmpty' => array(
                'rule' => array('notEmpty'),
                'message' => 'Ingrese el estado',
            ),
        ),
    );

    public $hasMany = array(
        'Particione' => array(
            'className' => 'Particione',
            'foreignKey' => 'zona_id',
            'dependent' => false,
            'conditions' => '',
            'fields' => '',
            'order' => '',
            'limit' => '',
            'offset' => '',
            'exclusive' => '',
            'finderQuery' => '',
            'counterQuery' => ''
        )
    );
}

```

Fig. 64: Implementación Código Modelo Zonas

Elaborado por: Diego Poaquiza – Investigador

Vista Zona

index.ctp: Muestra la información de los registros almacenados en la Base de Datos de la tabla Zona y visualiza un botón de eliminación por cada registro visualizado.

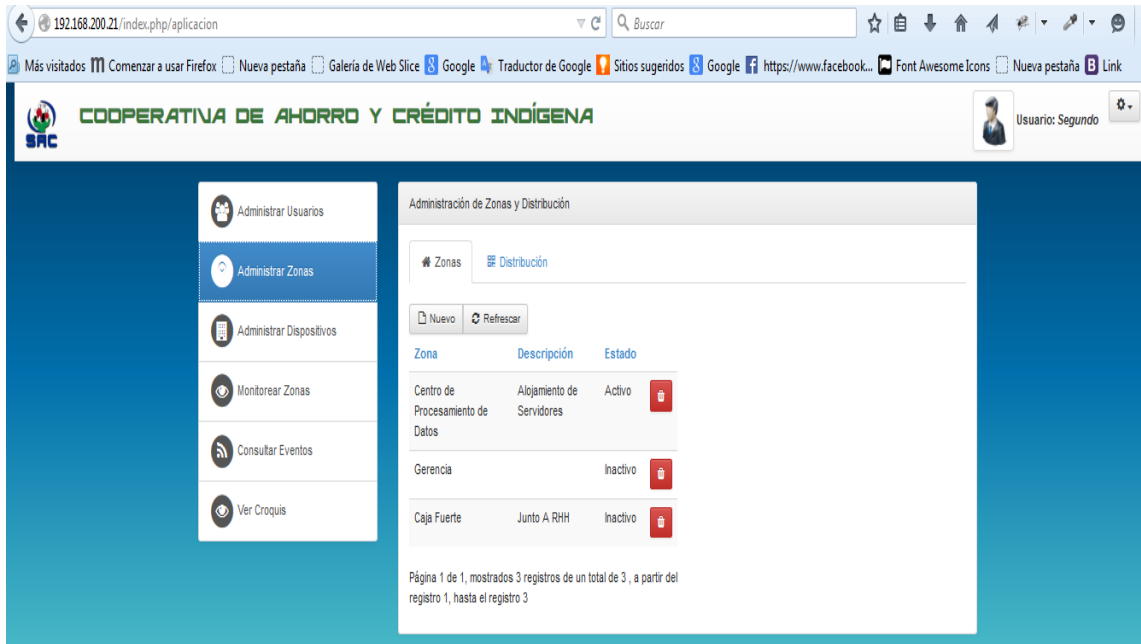


Fig. 65: Vista Index de la Zona
Elaborado por: Diego Poaquiza – Investigador

```

<?php
    $this->Paginator->options(array(
        'update' => '#tablaListaZonas',
        'evalScripts' => true
    ));
?>
<div class="zonas index table-responsive">
    <table class="table table-hover table-striped">
        <thead>
            <tr>
                <th><?php echo $this->Paginator->sort('nombre','Zona'); ?></th>
                <th><?php echo $this->Paginator->sort('descripcion','Descripción'); ?></th>
                <th><?php echo $this->Paginator->sort('estado','Estado'); ?></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($zonas as $zona): ?>

```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

```

<tr id="mostrarZona<?php echo $zona['Zona']['id'];?>">
  <td><?php echo h($zona['Zona']['nombre']); ?>&nbsp;</td>
  <td><?php echo h($zona['Zona']['descripcion']); ?>&nbsp;</td>
  <td><?php echo h($zona['Zona']['estado']); ?>&nbsp;</td>
  <!-- BOTON ELIMINAR ZONA-->
  <td><button id="btnEliminarZona<?php echo $zona['Zona']['id'];?>"
    type="button" class="btn btn-danger btn-sm tol" title="Eliminar">
    <i class="fa fa-trash"></i></button>
  <!--MODAL ELIMNAR ZONA-->
<div id="modalEliminarZona<?php echo $zona['Zona']['id'];?>" class="modal fade"
tabindex="-1" role="dialog" aria-labelledby="mySmallModalLabel"
aria-hidden="true">
  <div class="modal-dialog modal-sm">
    <div class="modal-content">
      <div class="modal-header">button type="button" class="close"
      data-dismiss="modal">
      <span aria-hidden="true">&times;</span>
      <span class="sr-only">Close</span>
    </button>
    <h4 class="modal-title">Eliminar Zona</h4>
    </div>
    <div class="modal-body">
      <p>Desea eliminar la Zona <strong> <?php echo zona['Zona']['nombre']?></strong>
      </p>
    </div>
    <div class="modal-footer">
      <button type="button" class="btn btn-default" data-dismiss="modal">Cancelar
    </button>
    <?php echo $this->Js->submit('Eliminar',array(
      'url'=> array('controller'=>'zonas','action'=>'delete',$zona['Zona']['id']),
      'method'=>'post','async'=>true,
      'success'=>'var a = eval("(" + data + ")");
      if(a["estado"]){
        $("#modalEliminarZona'.$zona['Zona']['id'].'").modal("hide");
        $("#modalEliminarZona'.$zona['Zona']['id'].'").on("hidden.bs.modal",
        function (e) {
          $("#btnRefrescarZonas").trigger("click");
        });
      }
    );
  
```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

```

}
    mensajesDiego(a["html"][0],a["tipomsg"]);
    'class'=>'btn btn-danger', 'div'=>false
    )
); ?>
</div> <!-- Modal-footer-->
</div> <!-- Modal-content-->
</div> <!-- Modal-dialog -->
</div> <!-- Modal Eliminar Usuario-->
</td>
</tr>
<?php $this->Js->get('#btnEliminarZona'.$zona['Zona']['id']);
    $this->Js->event('click',
        $('#modalEliminarZona'.$zona['Zona']['id'].').modal("show");'
    );
$this->Js->get('#mostrarZona'.$zona['Zona']['id']);
    $this->Js->event('click',
        $this->Js->request(
            array('controller'=>'zonas','action'=>'view',$zona['Zona']['id']),
            array( 'method'=>'get',
                'async'=>true,
                'update'=>'#ajaxInfoZonas)
        )
    );
?>
<?php endforeach; ?>
</tbody>
</table>
<p>
<?php echo $this->element('paginador'); ?>
<?php echo $this->element('paginador2'); ?>
</div>
<script type="text/javascript">

$(function(){
    $('.tol').tooltip();
});
</script>

```

edit.ctp: Muestra la información a editar del registro seleccionado por el usuario y visualiza los botones de guardar y cancelar para su respectiva acción.

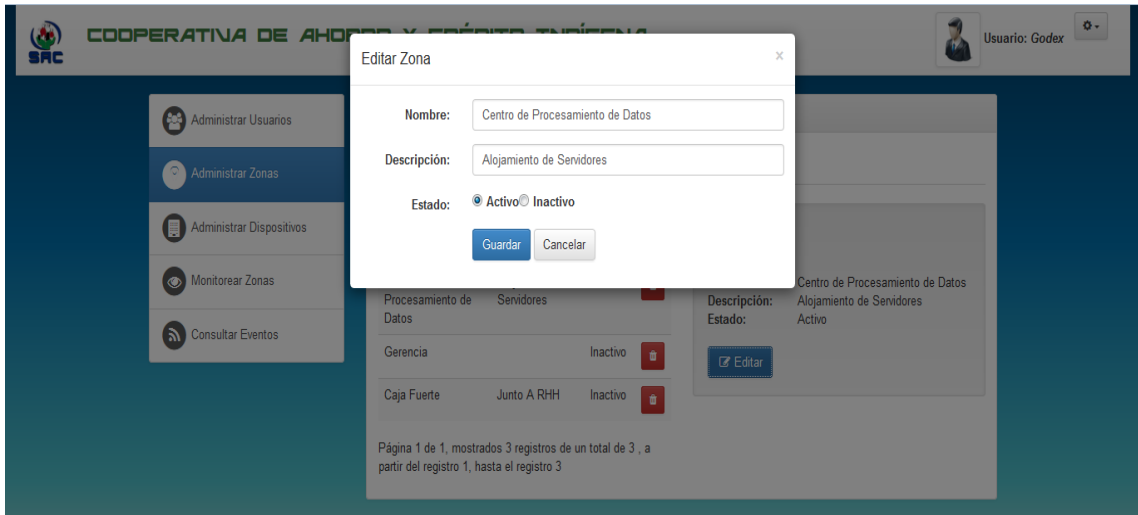


Fig. 66: Vista Editar registro de la Zona

Elaborado por: Diego Poaquiza – Investigador

```

<div class="zonas form">
  <?php echo $this->Form->create('Zona', array(
    'inputDefaults' => array(
      'div' => 'form-group',
      'label'=>array(
        'class' => 'col col-md-3 control-label'
      ),
      'wrapInput' => array('class'=>'col-md-9'),
      'class' => 'form-control'
    ),
    'class' => 'form-horizontal ',
  ));
  ?>
  <fieldset>
  <?php
    echo $this->Form->input('id');
    echo $this->Form->input('nombre',array(
      'label' => array(
        'text' => 'Nombre:'
      ),
      'placeholder'=>'Ingrese nombre'
    )
  
```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

```

);
echo $this->Form->input('descripcion',array(
    'label' => array(
        'text' => 'Descripción:'
    ),
    'placeholder'=>'Ingrese Descripción'
)
);
echo $this->Form->input('estado',array(
    'label' => array(
        'text' => 'Estado:'
    ),
    'placeholder'=>'Ingrese Estado'
)
);
?>
</fieldset>
<div class="form-group">
    <div class="col col-md-9 col-md-offset-3">
        <?php echo $this->Js->submit('Guardar',
            array(
                'url'=> array('controller'=>'zonas','action'=>'edit'),
                'method'=>'post',
                'async'=>true,
                'success'=>'
                    var a = eval("(" + data + ")");
                    if(a["estado"]){
                        $("#modalEditarZona").modal("hide");
                        $("#btnRefrescarZonas").trigger("click");
                    }
                    mensajesDiego(a["html"][0],a["tipomsg"]);
                '
            );
            'class'=>'btn btn-primary',
            'div'=>false
        )
    );
?>
    <button type="button" class="btn btn-default" data-
dismiss="modal">Cancelar</button>
    </div>
</div>
<?php echo $this->Form->end(); ?>
</div>

```

add.ctp: Muestra los campos del formulario Zonas para guardar un nuevo registro en la Base de datos, contiene los botones guardar y cancelar con su respectiva acción.

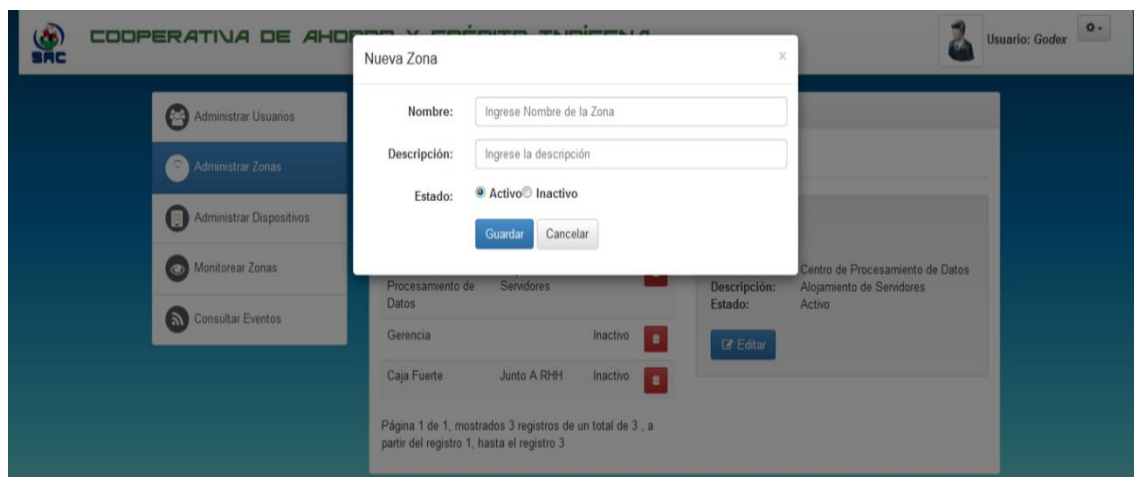


Fig. 67: Vista guardar nuevo registro de la Zona

Elaborado por: Diego Poaquiza – Investigador

```

<div class="zonas form">
<?php echo $this->Form->create('Zona', array(
    'inputDefaults' => array(
        'div' => 'form-group',
        'label'=>array(
            'class' => 'col col-md-3 control-label'
        ),
        'wrapInput' => array('class'=>'col-md-9'),
        'class' => 'form-control'
    ),
    'class' => 'form-horizontal ',
));
?>
<fieldset>
<?php
echo $this->Form->input('nombre', array(
    'label' => array(
        'text' => 'Nombre:'
    ),
    'placeholder'=>'Ingrese Nombre de la Zona'
)
);

```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

```

echo $this->Form->input('descripcion',array(
    'label' => array(
        'text' => 'Descripción:'
    ),
    'placeholder'=>'Ingrese la descripción'
)
);
echo $this->Form->input('estado',array(
    'label' => array(
        'text' => 'Estado:'
    ),
    'placeholder'=>'Elija Estado'
)
);
?>
<div class="form-group">
    <div class="col col-md-9 col-md-offset-3">
        <?php echo $this->Js->submit('Guardar',
            array(
                'url'=> array('controller'=>'zonas','action'=>'add'),

                'method'=>'post',
                'async'=>true,
                'success'=>'
                    var a = eval("(" + data + ")");

                    if(a["estado"]){
                        $("#modalNuevaZona").modal("hide");
                        $("#btnRefrescarZonas").trigger("click");
                    }
                    mensajesDiego(a["html"][0],a["tipomsg"]);
                '
            );
            'class'=>'btn btn-primary',
            'div'=>false
        )
    );
    ?>
    <button type="button" class="btn btn-default" data-
dismiss="modal">Cancelar</button>
    </div>
</div>
<?php echo $this->Form->end(); ?>
</div>

```


view.ctp: Muestra la descripción completa del registro seleccionado por el usuario, con un botón editar si se desea hacer alguna modificación sobre el registro.



Fig. 68: Vista registro seleccionado de la Zona

Elaborado por: Diego Poaquiza – Investigador

```

<div class="row">
<div class="col-md-12">
<div class="zonas view well">
<h3><?php echo __('Zona:'); ?></h3>
<dl class="dl-horizontal">
<dt><?php echo __('Zona:'); ?></dt>
<dd>
<?php echo h($zona['Zona']['nombre']); ?>
</dd>
<dt><?php echo __('Descripción:'); ?></dt>
<dd>
<?php echo h($zona['Zona']['descripcion']); ?>
</dd>
<dt><?php echo __('Estado:'); ?></dt>
<dd>
<?php echo h($zona['Zona']['estado']); ?>
</dd>
</dl>
<div>
<button id="btnEditarZona" type="button" class="btn btn-primary"><i
class="fa fa-edit fa-fw"> </i> Editar</button>
</div>
</div> <!-- WELL-->
</div>
</div>

```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Controlador Zonas

A continuación se describe el código del controlador de Zonas con cada una de acciones de gestión que puede realizar el usuario sobre el formulario.

```

<?php
App::uses('AppController', 'Controller');
class ZonasController extends AppController {
public $components = array('Paginator','RequestHandler');
public $uses = array('Zona','Particione');
public function index() {
    $this->Paginator->settings = $this->paginate;
    $this->Zona->recursive = 0;
    $this->set('zonas', $this->Paginator->paginate());
}
public function view($id = null) {
    if (!$this->Zona->exists($id)) {
        throw new NotFoundException(__('Invalid zona'));
    }
    $options = array('conditions' => array('Zona.' . $this->Zona->primaryKey =>
$id));
    $this->set('zona', $this->Zona->find('first', $options));
}
public function add() {
    $a = array('estado' => FALSE, 'datos'=>null,
'html'=>null,'tipomsg'=>'blackgloss');
    if ($this->request->is('post')) {
        $this->autoRender = FALSE;
        $this->Zona->set($this->request->data);
        if ($this->Zona->validates()) {
            $this->Zona->create();
            if ($this->Zona->save($this->request->data)) {
                $a['html'][0]='Zona guardada correctamente';
                $a['estado']=true;
                $a['tipomsg']='success';
            } else {
                $a['html'][0]='No se pudo guardar la Zona';
                $a['tipomsg']='danger';
            }
        }
    }
    else {
        $errores = $this->Zona->validationErrors;
        $acu=0;
        foreach ($errores as $indice => $subarray) {

```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

```

        $a['html'][$acu++] = $subarray[0];
    }
    $a['tipomsg']='warning';
}
return json_encode($a);
} //post
}
public function edit($id = null) {
    $a = array('estado' => FALSE, 'datos'=>null,
'html'=>null,'tipomsg'=>'blackgloss');
    if (!$this->Zona->exists($id)) {
        $a['html'][0]='Zona invalida';
        $a['tipomsg']='danger';
    }

if ($this->request->is(array('post', 'put'))) {
    $this->autoRender = FALSE;//Poner
    $this->Zona->set($this->request->data);
    if ($this->Zona->validates()) {
        if ($this->Zona->save($this->request->data)) {
            $a['html'][0]='Zona guardada correctamente';
            $a['estado']=true;
            $a['tipomsg']='success';
        } else {
            $a['html'][0]='No se pudo guardar la Zona';
            $a['tipomsg']='danger';
        }
    }
}
else {
    $errores = $this->Zona->validationErrors;
    $acu=0;
    foreach ($errores as $indice => $subarray) {
        $a['html'][$acu++] = $subarray[0];
    }
    $a['tipomsg']='warning';
}
return json_encode($a);

```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

```

    }
else {
    $options = array('conditions' => array('Zona.' . $this->Zona->primaryKey =>
    $id));
    $this->request->data = $this->Zona->find('first', $options);
    }
}
public function delete($id = null) {
    $a = array('estado' => FALSE, 'datos'=>null,
    'html'=>null,'tipomsg'=>'blackgloss');
    $this->autoRender = FALSE;
    $this->Zona->id = $id;
    if (!$this->Zona->exists()) {
        $a['html'][0]='Zona invalida';
        $a['tipomsg']='danger';
    }
    $this->request->allowMethod('post', 'delete','get');
    $conteoZonas=array('conditions' => array('Particione.zona_id'=>$id));
    $select=$this->Particione->find('count',$conteoZonas);

    if (!$select) {
        if ($this->Zona->delete()) {
            $a['html'][0]='Zona eliminada correctamente';
            $a['estado']=true;
            $a['tipomsg']='success';
        } else {

            $a['html'][0]='No se pudo eliminar la zona, intenta nuevamente';
            $a['tipomsg']='danger';
        }
    }
    else{
        $a['html'][0]='No se puede eliminar Zona, La Zona está asignada a
Particiones';
        $a['tipomsg']='danger';
    }
    return json_encode($a);
}
}

```

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

4.4 Pruebas

Las pruebas son básicamente una etapa destinada a la identificación de posibles fallos de implementación y calidad del software.

4.4.1 Pruebas de Caja Blanca

Las pruebas de caja blanca se centran en la estructura interna del software en cada una de los detalles procedimentales por lo que su diseño está fuertemente ligado al código fuente.

Se derivan de casos de prueba que aseguren que durante la prueba se han ejecutado todas las sentencias del programa al menos una vez y que se ejercitan todas las condiciones lógicas.

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo
- Se utilizan las decisiones en su parte verdadera y en su parte falsa
- Se ejecuten todos los bucles en sus límites
- Se utilizan todas las estructuras de datos internas

Prueba del camino básico

El método del camino básico permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

ANÁLISIS	DISEÑO	IMPLEMENTACIÓN	PRUEBAS
----------	--------	----------------	---------

Grafo de flujo del siguiente código del programa

Validación de la edición de un registro.

Validación	
public function edit(\$id = null) {	0
\$a = array('estado' => FALSE, 'datos'=>null, 'html'=>null,'tipomsg'=>'blackgloss');	1
if (!\$this->Zona->exists(\$id)) {	2
\$a['html'][0]='Zona invalida'; \$a['tipomsg']='danger'; }	3
if (\$this->request->is(array('post', 'put')))	4
\$this->autoRender = FALSE; \$this->Zona->set(\$this->request->data);	5
if (\$this->Zona->validates()) {	6
if (\$this->Zona->save(\$this->request->data)) {	7
\$a['html'][0]='Zona editada correctamente'; \$a['estado']=true; \$a['tipomsg']='success';}	8
else {	9
\$a['html'][0]='No se pudo editada la Zona'; \$a['tipomsg']='danger'; }	10
}	11
else {	12
\$errores = \$this->Zona->validationErrors; \$acu=0; foreach (\$errores as \$indice => \$subarray) { \$a['html'][\$acu++] = \$subarray[0]; } \$a['tipomsg']='warning'; }	13
return json_encode(\$a);	14
}	15
\$options = array('conditions' => array('Zona.' . \$this->Zona->primaryKey => \$id)); \$this->request->data = \$this->Zona->find('first', \$options);}	16
}	17

Tabla 40 Prueba de Caja Blanca

Elaborado por: Diego Poaquiza – Investigador

Grafo del flujo de validación

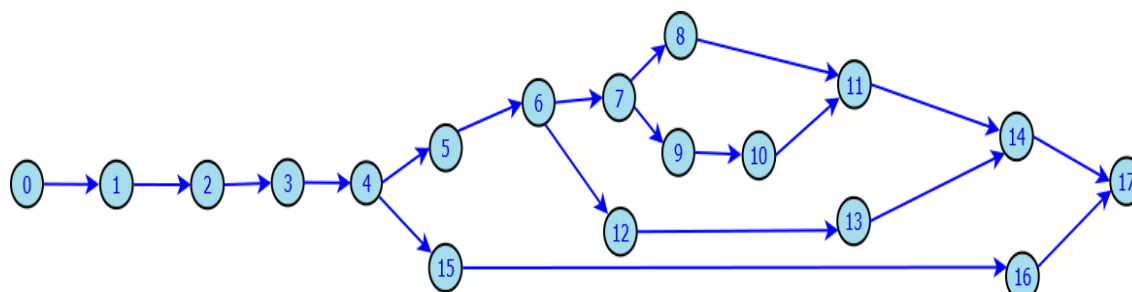


Fig. 69: Grafo de Validación
Elaborado por: Diego Poquiza – Investigador

Complejidad Ciclomática

$$V(G) = A \text{ (arista)} - N \text{ (nodos)} + 2$$

$$V(G) = 20 - 18 + 2 = 4$$

Caminos Básicos

#1: 0-1-2-3-4-15-16-17

#2: 0-1-2-3-4-5-6-7-8-11-14-17

#3: 0-1-2-3-4-5-6-7-9-10-11-14-17

#4: 0-1-2-3-4-5-6-12-13-14-17

Para la comprobación de la prueba de camino básico se tomará el camino #1.

Comprobación camino #1	
Pasos	Nodos
Función edit del Controlador Zonas recibe un parámetro \$id del registro a editar.	0
Declaramos una variable para guardar información de los mensajes de validación a mostrar al usuario en caso de cualquier evento.	1
Comprobamos si no existe la id de la zona en la base de datos del registro a editar.	2
Si no existe la id de la zona, asignamos la información de Zona inválida y tipo de mensaje danger para mostrar al usuario.	3
Comprobamos si el tipo de llamada para la edición del registro que se hizo mediante el formulario es post o put.	4

Caso contrario.	15
Fin de camino.	17

Tabla 41 Comprobación camino básico
Elaborado por: Diego Poaquiza – Investigador

4.4.2 Pruebas de Caja Negra

Esta prueba se lleva cabo sobre la interfaz del sistema, por lo tanto estas son completamente indiferentes del comportamiento interno y a su estructura, estas pruebas se enfocan en la funcionalidad y requerimientos especificados del software.

- Comprobación de que cada una de las páginas se ajuste a los datos necesitados y requeridos.
- Que la información sea mostrada de forma correctamente, rápida, adecuada y entendida por los usuarios.
- Nunca visualizar información innecesaria y secreta en ninguna caso.
- Entrada y salida de datos.
- Validación de cajas de texto para el ingreso de información correcta.

Prueba de caja negra en una de las interfaces del formulario dispositivo.

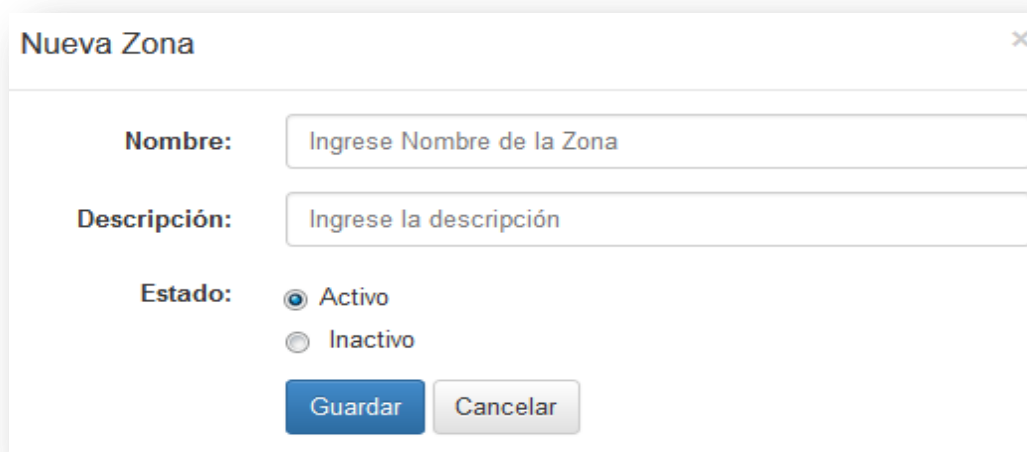
Información mostrada de forma correctamente, rápida, adecuada y entendida por los usuarios.



Fig. 70: Prueba de Caja Negra información mostrada correctamente
Elaborado por: Diego Poaquiza – Investigador

Ingreso de Zonas: Validación de los campos a ingresar, el resultado esperado se basa en los siguientes casos:

- Si el campo es requerido.



The screenshot shows a window titled "Nueva Zona" with a close button (X) in the top right corner. It contains three input fields: "Nombre:" with the placeholder text "Ingrese Nombre de la Zona", "Descripción:" with the placeholder text "Ingrese la descripción", and "Estado:" with two radio button options: "Activo" (selected) and "Inactivo". At the bottom, there are two buttons: "Guardar" (blue) and "Cancelar" (grey).

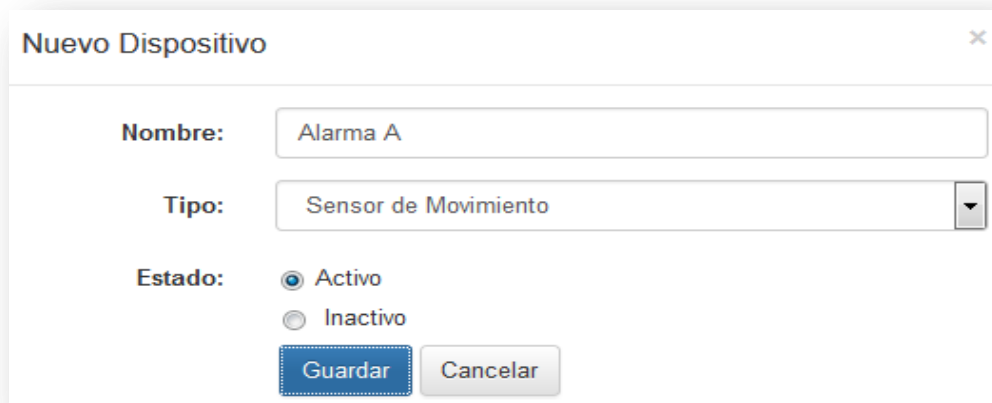
Fig. 71: Prueba de Caja Negra validación de campos requeridos
Elaborado por: Diego Poaquiza – Investigador

El mensaje mostrado al usuario:

Advertencia !! Ingrese el nombre del Dispositivo x

Fig. 72: Prueba de Caja Negra mensaje de advertencia
Elaborado por: Diego Poaquiza – Investigador

- En el caso de ser único y está registrado en la base de datos.



The screenshot shows a window titled "Nuevo Dispositivo" with a close button (x) in the top right corner. The form contains three fields: "Nombre:" with the value "Alarma A", "Tipo:" with a dropdown menu set to "Sensor de Movimiento", and "Estado:" with radio buttons for "Activo" (selected) and "Inactivo". At the bottom, there are two buttons: "Guardar" (highlighted with a blue border) and "Cancelar".

Fig. 73: Prueba de Caja Negra validación de campos únicos
Elaborado por: Diego Poaquiza – Investigador

El mensaje mostrado al usuario:

Advertencia !! Nombre del dispositivo esta registrado x

Fig. 74: Prueba de Caja Negra mensaje de advertencia campo único
Elaborado por: Diego Poaquiza – Investigador

➤ Campo alfanumérico.



The screenshot shows the same "Nuevo Dispositivo" window. The "Nombre:" field now contains "Sensor de Contacto t?=_/". The "Tipo:" dropdown is still "Sensor de Movimiento" and "Estado:" is still "Activo". The "Guardar" button is highlighted with a blue border, indicating a validation error.

Fig. 75: Prueba de Caja Negra validación de campos alfanuméricos
Elaborado por: Diego Poaquiza – Investigador

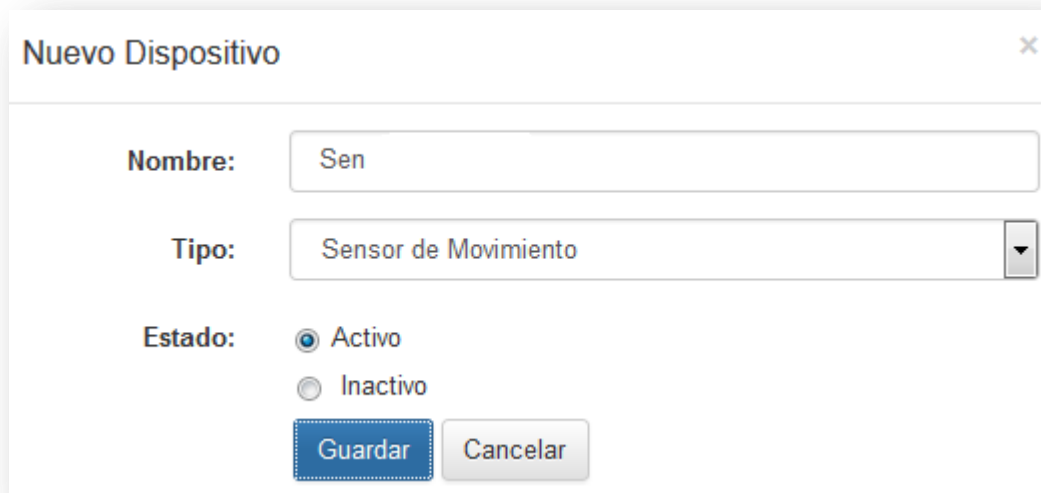
El mensaje mostrado al usuario:

Advertencia !! Campo nombre solo letras x

Fig. 76: Prueba de Caja Negra mensaje de advertencia campo alfanumérico

Elaborado por: Diego Poaquiza – Investigador

- Número mínimo y máximo de caracteres.



Nuevo Dispositivo

Nombre: Sen

Tipo: Sensor de Movimiento

Estado: Activo Inactivo

Guardar Cancelar

Fig. 77: Prueba de Caja Negra validación cantidad de caracteres

Elaborado por: Diego Poaquiza – Investigador

El mensaje mostrado al usuario:

Advertencia !! Campo nombre entre 6 y 60 caracteres x

Fig. 78: Prueba de Caja Negra mensaje de advertencia cantidad de caracteres

Elaborado por: Diego Poaquiza – Investigador

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Con el desarrollo del proyecto se logró cumplir el objetivo principal propuesto, que consiste en desarrollar una aplicación web que permita el monitoreo del Centro de Procesamiento de Datos mediante sensores, permitiendo así un monitoreo en tiempo real de manera oportuna y eficaz en cuanto a seguridad se refiere.
- La utilización de la investigación de campo, usando como técnica la observación y la entrevista facilitó la comprensión detallada y análisis de la información recolectada sobre los requerimientos necesarios para el desarrollo de la aplicación web.
- La metodología en cascada se utilizó para el desarrollo de la aplicación web ya que permitió descomponer la actividad global en fases permitiendo que cada una de las etapas fueran realizadas correctamente teniendo una mejor administración de las actividades a realizarse.
- La utilización del Lenguaje Unificado de Modelado (UML), facilitó la comprensión de cómo la aplicación debe trabajar y su interacción con los usuarios.
- La utilización del Framework Cakephp open source basado en la arquitectura MVC permitió el desarrollo de la aplicación web, gracias a las características

que ofrece, numerosa comunidad y cantidad de documentación que se logró encontrar, concluyendo con el desarrollo de la aplicación web sin alguna restricción.

- Las herramientas y metodología aplicada para el desarrollo e implementación de la aplicación web, han sido adecuadas y de gran ayuda permitiendo que el proyecto se desarrolle de acuerdo a los objetivos planteados y requerimientos.
- Con las pruebas se pudo encontrar y corregir errores que permitieron asegurar el correcto funcionamiento de la aplicación web.
- La implementación de la aplicación web permitió un control y monitoreo oportuno en tiempo real del CPD mediante los sensores instalados.
- Por otra parte se mejoró el sistema de alarma adicionando contactos magnéticos en las puertas y ventanas lo cual permitió mejorar la seguridad en el CPD.
- Para el procesamiento y adquisición de los datos de los sensores, se utilizó la tarjeta Arduino Uno, debido que en este sistema se puede trabajar con la velocidad de comunicación del puerto, trabajar con señales analógicas y digitales lo cual fue de gran ayuda para la realización del proyecto.
- La utilización de un mini ordenador llamado Cubieboard A20 y conocimiento de una nueva tecnología que ha ido ganando espacio para el desarrollo de proyectos, ha permitido que la aplicación web se monte en dicho ordenador sin ningún problema asegurando su funcionamiento, además de la ventaja que se puede trasladar e implementar de manera fácil, a un costo menor y en cualquier lugar debido a su tamaño.
- La utilización del software y hardware libre permitió reducir los costos de la del proyecto de tal manera que puede ser accesible por la mayoría de usuarios.

- Gracias a los avances tecnológicos y el conocimiento adquirido se puede desarrollar aplicaciones complejas de monitoreo y control utilizando dispositivos electrónicos.

5.2 RECOMENDACIONES

- Es necesario instruir y capacitar al personal que será encargado de la utilización y administración de la aplicación.
- Se recomienda a la persona encargada de la administración de la aplicación obtener de forma constante y periódica respaldos de la base de datos para evitar pérdida de información, ya que muchas veces los equipos informáticos suelen ser vulnerables a ciertos daños físicos o lógicos, los mismos que traerían graves inconvenientes.
- En caso de que se desee implementar dispositivos en toda la institución a futuro, el servidor debe ser migrado a otro de mejor tecnología y características para un buen funcionamiento.
- Los usuarios que tienen acceso a la aplicación, deben tener cuidado en el manejo de su usuario y contraseña, puesto que el acceso de personas no autorizadas al sistema podría provocar daños en el correcto manejo y administración de la aplicación.
- Se recomienda a la persona encargada del sistema de alarma hacer revisiones periódicas del sistema, para verificar que los dispositivos electrónicos de seguridad estén trabajando en buenas condiciones, así evitar inconsistencias en el sistema de alarma y en la aplicación web.
- Ciertos componentes electrónicos requieren que estén continuamente energizados por lo cual se recomienda utilizar una UPS, la cual se encargará de entregar energía en caso de que exista corte del suministro público.

Bibliografía

- [1] W. G. Perez Nata, "Aplicación Web para la distribución de espacios disponibles de parqueo en la Universidad Técnica de Ambato campus Huachi Chico," Ambato - Ecuador, 2014.
- [2] G. D. Ruiz Mesias, "Sistema de control y monitoreo para mejorar los procesos de administración de los laboratorios de las Carreras de Sistemas, Electrónica e Industrial en la FISEI-UTA," Ambato - Ecuador, 2014.
- [3] C. Masaquiza and L. Saquinga, "Diseño y construcción de un sistema de control, monitoreo y vigilancia por cámara web, via internet, para la alimentación de animales domésticos," Latacunga - Ecuador, 2013.
- [4] J. Macías, L. Reyes, F. Reyes and D. Roca, "Desarrollo e implementación de un sistema de monitoreo y control mediante cámaras para la investigación, el registro. y evolución de las especies en peligro de extinción del jardín botánico de la Universidad Técnica de Manabí," MANABÍ – ECUADOR, 2012.
- [5] M. Rodríguez, "Monitoreo y Evaluación: Definición, Funciones y Uso," [En línea]. Disponible: http://www.pasca.org/userfiles/M1_T7_RODRIGUEZ_PA.pdf.
- [6] G. Escrivá Gascó, R. M. Romero Serrano and D. J. Ramada, "Seguridad informática," Macmillan Iberia, S.A, 2013. [En línea]. Disponible: <http://site.ebrary.com/lib/utasp/Doc?id=10820963&ppg=34>.
- [7] C. Arriazu Serrano and Ó. J. Ayuso De La Torre, Desarrollo de productos electrónicos (I-II), Centro de Publicaciones Ministerio de Educación y Ciencia, 2000.
- [8] "Adquisición de Datos," [En línea]. Disponible: <http://www.ni.com/data-acquisition/what-is/esa/>.
- [9] J. Muñoz Guerrero, Sistemas de seguridad, Madrid: Paraninfo, 2006.
- [10] "Seguridad Física," [En línea]. Disponible: http://www.ewh.ieee.org/reg/9/files/noticieeros/Noticieero_82.pdf.
- [11] "Qué es un Sistema de Alarma," [En línea]. Disponible: <https://www.adt.com.mx/negocios/monitoreo-de-sistema-de-alarma>.
- [12] "Monitoreo," [En línea]. Disponible: <http://www.alarm-city.com/Pagina.php?cmd=view&id=52>.
- [13] S. Luján Mora, Programación de aplicaciones web, Alicante: Club Universitario, 2002.
- [14] "Servidores Web," [En línea]. Disponible: http://www.ecured.cu/index.php/Servidores_Web.

- [15] M. E. Rodríguez González, "Gestión de datos: bases de datos y sistemas gestores de bases de datos," España: Editorial UOC, 2013. [En línea]. Disponible: <http://site.ebrary.com/lib/utasp/Doc?id=10853383&ppg=9>.
- [16] E. Quero Catalinas, Fundamentos de programación, Madrid: Paraninfo, 2001.
- [17] L. Masanti, Diseño Web, Buenos Aires: Estrada.
- [18] A. Cobo, P. Gómez y R. Rocha, PHP y MYSQL Tecnologías para el desarrollo web, España: Díaz de Santos, 2005.
- [19] "Manual de Jquery," [En línea]. Disponible: http://dmaspv.com/files/page/07042011180222_manual%20de%20jquery%20en%20pdf%20desarrollowebcom.pdf.
- [20] G. Emmanuel, Javascript: Conceptos básicos y avanzados, Barcelona : ENI, 2009.
- [21] J. M. Alarc, Fundamentos de Javascript Y Ajax, Krasis Consulting.
- [22] "CS5,"[En línea]. Disponible: <http://books.google.com.ec/books?id=UTU9kis3LwsC&pg=PT325&dq=definicion+de+css&hl=es&sa=X&ei=urFKVP-4M7DbASs1YGwDg&ved=0CBoQ6AEwAA#v=onepage&q=definicion%20de%20css&f=false>.
- [23] M. Moro y R. Adolf, MARKETING DIGITAL, Paraninfo S.A., 2014.
- [24] "Font-Awesome," [En línea]. Disponible: <http://fontawesome.github.io/Font-Awesome/>.
- [25] "Introducción a JSON," [En línea]. Disponible: <http://www.json.org/json-es.html>.
- [26] V. Eslava, El nuevo php conceptos avanzados, España: Bubok publishing, 31 de Octubre, 2013.
- [27] A. Fernández, Python 3 al descubierto, RC LIBROS, 2012 .
- [28] "Exp-Tech.de," [En línea]. Disponible: <http://www.exp-tech.de/service/datasheet/DHT11.PDF>.
- [29] "Señales Analógicas y Digitales," [En línea]. Disponible: <http://www.fceia.unr.edu.ar/enica3/da-ad.pdf>.
- [30] "Que es Arduino," [En línea]. Disponible: <http://arduino.cc/en/pmwiki.php?n=Guide/Introduction>.
- [31] "Cubieboard Docs," [En línea]. Disponible: <http://docs.cubieboard.org/products/start#a20-cubieboard>.

[32] R. S. Pressman, Ingeniería del Software: Un enfoque práctico.

ANEXOS

ANEXO 1: GUÍA DE OBSERVACIÓN.
UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E
INDUSTRIAL

GUÍA DE OBSERVACIÓN
Objetivo: Recolectar información sobre los procesos de seguridad física utilizados en el Centro de Procesamientos de Datos.
Tipo de Observación: Observación directa.
INDICADORES A OBSERVAR
➤ Control de Acceso Físico.
➤ Dispositivos de Seguridad Física.
➤ Encendido del sistema de Alarma.
➤ Apagado del sistema de Alarma.
➤ Activación de la Sirena del Sistema de Alarma.

ANEXO 2: INSTALACIÓN ENTORNO DE DESARROLLO IDE ARDUINO UNO

PASO 1

Descargar la IDE de Arduino de la página oficial según el sistema operativo.

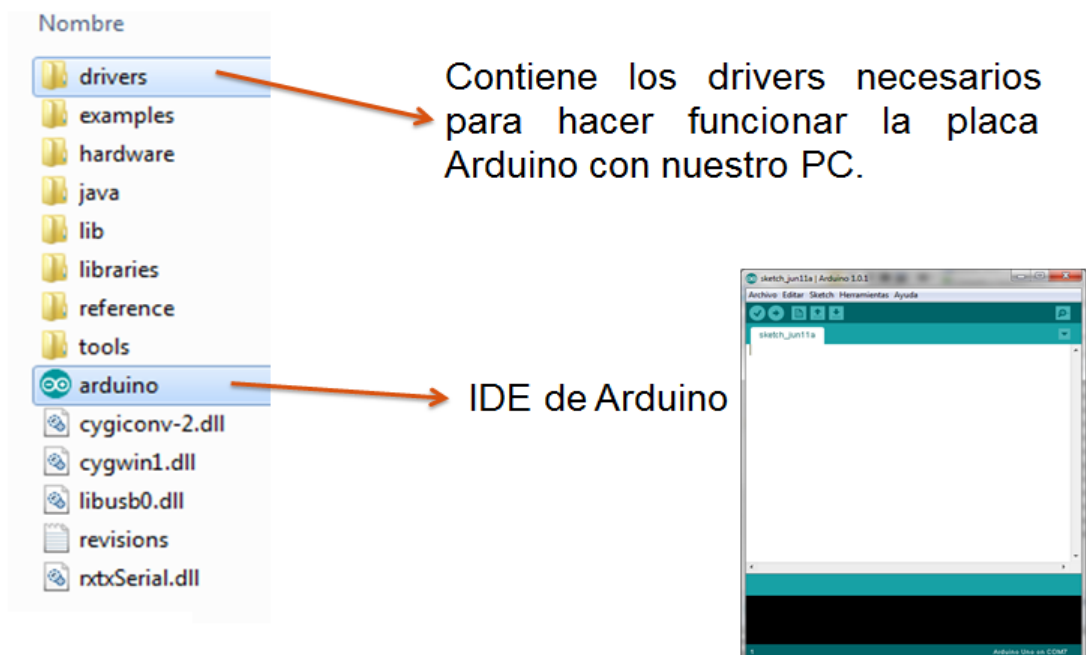
Link de descarga:

<http://arduino.cc/es/Main/Software> (Español)

<http://arduino.cc/en/Main/Software> (Ingles)

Disponible para: Windows, Mac OS X, Linux

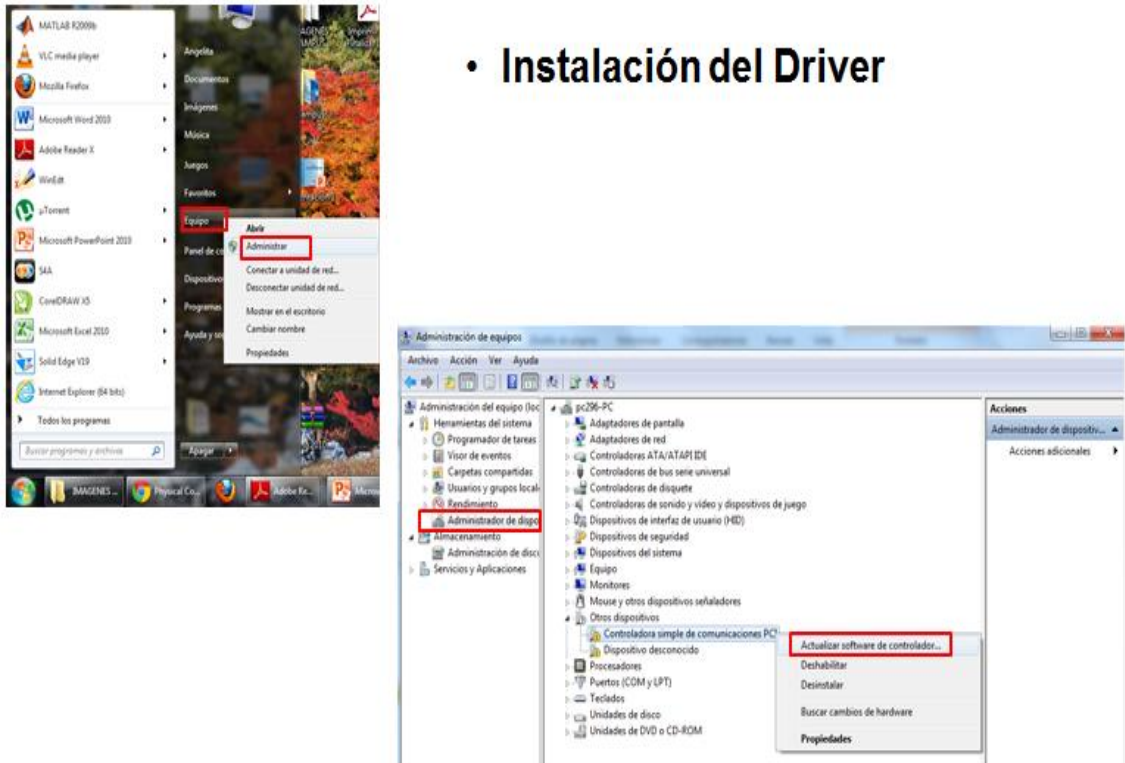
Instalación de Arduino en Windows.



Pantalla 1: Instalación Software Arduino Uno
Elaborado por: Diego Poaquiza - Investigador

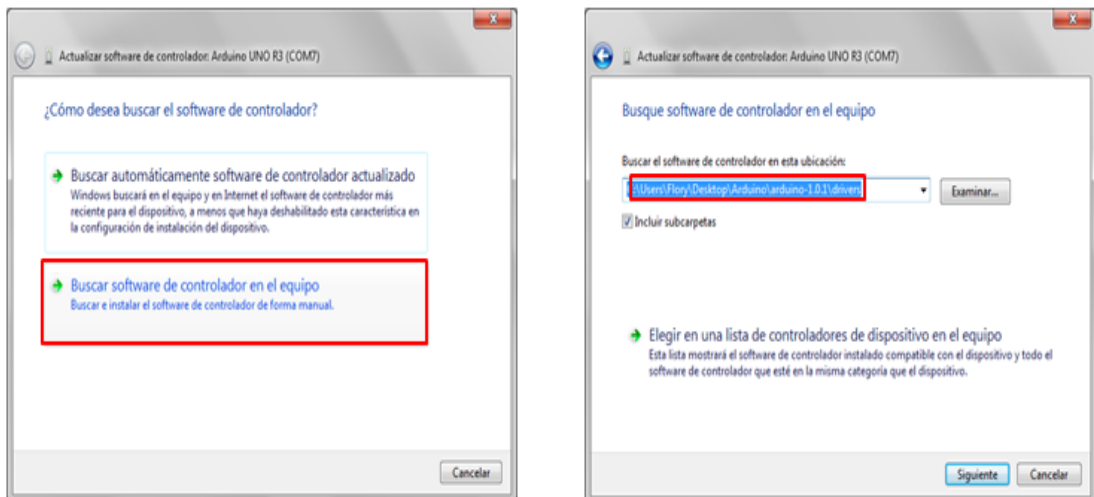
Instalación del Driver: Vamos a administrar dispositivos (Inicio-clic derecho en equipo - administrar- administrar dispositivos) buscamos en otros dispositivos los elementos desconocidos y con clic derecho seleccionamos actualizar controlador:

• Instalación del Driver



Pantalla 2: Instalación Driver Arduino Uno
Elaborado por: Diego Poaquiza - Investigador

Ahora seleccionamos buscar software de controlador en el equipo y buscamos la ubicación de carpeta Drivers.



Pantalla 3: Instalación Driver Arduino Uno
Elaborado por: Diego Poaquiza - Investigador

ANEXO 3: Programación Arduino Uno

El código que se presenta a continuación se encarga del proceso de obtención y envío de datos por el puerto serie.

```
#define DHT11_PIN 0
const int spu= 4;
int sve= 3;
int spr= 2;
int s1= 0;
int s2=0;
int s3=0;
int valor=0;
byte read_dht11_dat()
{
  byte i = 0;
  byte result=0;
  for(i=0; i< 8; i++)
  {
    while(!(PINC & _BV(DHT11_PIN))); // wait for 50us
    delayMicroseconds(30);
    if(PINC & _BV(DHT11_PIN))
      result |= (1<<(7-i));
    while((PINC & _BV(DHT11_PIN))); // wait '1' finish
  }
  return result;
}
void setup()
{
  DDRC |= _BV(DHT11_PIN);
  PORTC |= _BV(DHT11_PIN);
  Serial.begin(9600);
  pinMode(spu,INPUT);
  pinMode(sve,INPUT);
  pinMode(spr,INPUT);
}
void loop()
{
  byte dht11_dat[5];
  byte dht11_in;
  byte i;
  PORTC &= ~_BV(DHT11_PIN);
  delay(18);
  PORTC |= _BV(DHT11_PIN);
  delayMicroseconds(40);
  DDRC &= ~_BV(DHT11_PIN);
  delayMicroseconds(40);
```

```

dht11_in = PINC & _BV(DHT11_PIN);
if(dht11_in)
{
  Serial.println("error en el sensor");
  delay(100);
  return;
}
delayMicroseconds(80);
dht11_in = PINC & _BV(DHT11_PIN);
if(!dht11_in)
{
  Serial.println("dht11 start condition 2 not met");
  return;
}
delayMicroseconds(80);
for (i=0; i<5; i++)
  dht11_dat[i] = read_dht11_dat();
  DDRC |= _BV(DHT11_PIN);
  PORTC |= _BV(DHT11_PIN);
byte dht11_check_sum = dht11_dat[0]+dht11_dat[1]+dht11_dat[2]+dht11_dat[3];
// check check_sum
if(dht11_dat[4]!= dht11_check_sum)
{
  Serial.println("DHT11 checksum error");
}
String temp1=String(dht11_dat[2], DEC);
String temp2=String(dht11_dat[3], DEC);
String tempphu=String(""+temp1+"."+temp2+""); //Temperatura

s1= digitalRead(spu); //Puerta
s2= digitalRead(sve); //Ventana
s3= digitalRead(spr); //Presencia
Serial.println(tempphu);
Serial.println(s1);
Serial.println(s2);
Serial.println(s3);
delay(1000);
}

```

ANEXO 4: Programación Python

El código que se presenta a continuación se encarga de la lectura del puerto serie, y obtención de la información en modo texto.

Lectura del puerto serie:

```
#!/usr/bin/python
# Importamos la librería de PySerial
import serial
# Abrimos el puerto del arduino a 9600
PuertoSerie = serial.Serial('/dev/ttyACM0', 9600)
# Creamos un bucle sin fin
while True:
    # leemos hasta que encontremos el final de línea
    sArduino = PuertoSerie.readline()
    # Mostramos el valor leído y eliminamos el salto de línea del final
    print "Valor Arduino: " + sArduino.rstrip('\n')
```

ANEXO 5: Programación Arduino Uno

El siguiente código es para la obtención de la información de los dispositivos y armado de la cadena JSON en el lenguaje de programación IDE del Arduino.

```
#define DHT11_PIN 0
const int spu= 4;
int sve= 3;
int spr= 2;
int s1= 0;
int s2=0;
int s3=0;
int valor=0;
byte read_dht11_dat()
{
    byte i = 0;
    byte result=0;
    for(i=0; i< 8; i++)
    {
        while(!(PINC & _BV(DHT11_PIN))); // wait for 50us
        delayMicroseconds(30);
        if(PINC & _BV(DHT11_PIN))
            result |= (1<<(7-i));
        while((PINC & _BV(DHT11_PIN))); // wait '1' finish
    }
    return result;
}
```



```

}
void setup()
{
  DDRC |= _BV(DHT11_PIN);
  PORTC |= _BV(DHT11_PIN);
  Serial.begin(9600);
  pinMode(spu,INPUT);
  pinMode(sve,INPUT);
  pinMode(spr,INPUT);
}
void loop()
{
  byte dht11_dat[5];
  byte dht11_in;
  byte i;
  PORTC &= ~_BV(DHT11_PIN);
  delay(18);
  PORTC |= _BV(DHT11_PIN);
  delayMicroseconds(40);
  DDRC &= ~_BV(DHT11_PIN);
  delayMicroseconds(40);
  dht11_in = PINC & _BV(DHT11_PIN);
  if(dht11_in)
  {
    Serial.println("error en el sensor");
    delay(100);
    return;
  }
  delayMicroseconds(80);
  dht11_in = PINC & _BV(DHT11_PIN);
  if(!dht11_in)
  {
    Serial.println("dht11 start condition 2 not met");
    return;
  }
  delayMicroseconds(80);
  for (i=0; i<5; i++)
    dht11_dat[i] = read_dht11_dat();
  DDRC |= _BV(DHT11_PIN);
  PORTC |= _BV(DHT11_PIN);
  byte dht11_check_sum = dht11_dat[0]+dht11_dat[1]+dht11_dat[2]+dht11_dat[3];
  // check check_sum
  if(dht11_dat[4]!= dht11_check_sum)
  {
    Serial.println("DHT11 checksum error");
  }
  String temp1=String(dht11_dat[2], DEC);
  String temp2=String(dht11_dat[3], DEC);

```

```

String tempHu=String(""+temp1+"."+temp2+""); //Temperatura

s1= digitalRead(spu); //Puerta
s2= digitalRead(sve); //Ventana
s3= digitalRead(spr); //Presencia

String senpue= String(s1);
String senven= String(s2);
String senpre = String(s3);

String pre=String("{\"nombre':'MDG01movimiento','tipo_id':'1','estado':'1',
'valor':'"+senpre+"}");
String temp=String("{\"nombre':'MDG01temperatura','tipo_id':'2','estado':'1',
'valor':'"+tempHu+"}");
String pue=("{\"nombre':'MDG01Contacto','tipo_id':'3','estado':'1',
'valor':'"+senpue+"}");
String ven=("{\"nombre':'MDG01Contacto2','tipo_id':'3','estado':'1',
'valor':'"+senven+"}");
String jsonArduino=String("[{'idArduino':'MDGmonitoreo1','date':'14/11/2014',
'descripcion':'temperatura, contacto, movimiento', 'dispositivos':[]}");
Serial.print(jsonArduino);
Serial.print(pre+",");
Serial.print(temp+",");
Serial.print(pue+",");
Serial.print(ven);
Serial.println("]"]);
delay(1000);
}

```

ANEXO 6: Programación Python

El código siguiente de programación en Python se encarga de la lectura del puerto serie, obtención de la información y tratamiento de la misma para su respectivo uso, procesamiento y análisis.

```

#!/usr/bin/python
# -*- coding: utf-8 -*-
import re
import subprocess
import MySQLdb
import serial
import time
import json
import conexion
import metodos

def detectarDispositivos():

```

```

device_re =
re.compile("Bus\s+(?P<bus>\d+)\s+Device\s+(?P<device>\d+).+ID\s(?P<id>\w+:\w+)\s(?P<tag>.+)$", re.I)
df = subprocess.check_output("lsusb", shell=True)
devices = []
for i in df.split('\n'):
    if i:
        info = device_re.match(i)
        if info:
            dinfo = info.groupdict()
            dinfo['device'] = '/dev/bus/usb/%s/%s' % (dinfo.pop('bus'),
            dinfo.pop('device'))
            devices.append(dinfo)
return devices

def bscArduinos():
    aux = detectarDispositivos()
    arduinos = []
    for item in aux:
        if item['id'].find("2341:") >= 0:
            arduinos.append(item)
    return len(arduinos)

def datosArduinos():
    arduinos = bscArduinos()
    if arduinos > 0:
        cont = 0;
        jsonDispositivos= []
        ttys = subprocess.check_output("ls /dev/ttyACM*", shell=True)
        device_re2 = re.compile("/dev/", re.I)
        for tty in ttys.split('\n'):
            aux=0
            if tty:
                try:
                    PuertoSerie = serial.Serial(tty, 9600)
                    while True:
                        #PuertoSerie.write('1')
                        sArduino = PuertoSerie.readline()

                        if aux == 3:
                            jsonArduino = sArduino
                            break
                        aux = aux+1
                    print jsonArduino
                    jsonDispositivos.append(jsonArduino)
                except:
                    print '0 arduinos conectados'
        datosArduinos=[]

```

```

for jsonDispositivo in jsonDispositivos:
    jsoncorrecto = jsonDispositivo.replace('""', '')
    try:
        datos=json.loads(jsoncorrecto)
        datosArduinos.append(datos)
    except:
        print 'error Json'
return datosArduinos
#print datosArduinos
#print datos[0]['idArduino']
else:
    return False
def insertarDatosArduinos(dispositivosArduinos):
    for recorrerArduino in dispositivosArduinos:
        dispositivosDetectados = recorrerArduino[0]['dispositivos']
        for dispositivosDetectado in dispositivosDetectados:
            dispositivosInstancia=metodos.Metodos()
            nombreDispositivo=dispositivosInstancia.select(dispositivosDetectado['n
ombre'])
            if (nombreDispositivo[0][0]==0):

dispositivosInstancia.insert(dispositivosDetectado['nombre'],dispositivosDetectado['
tipo_id'],dispositivosDetectado['estado'])
            else:
idDispositivo=dispositivosInstancia.selectIdDisp(dispositivosDetectado['nombre'])
dispositivosInstancia.update(dispositivosDetectado['valor'],str(idDispositivo[0][0]))

```

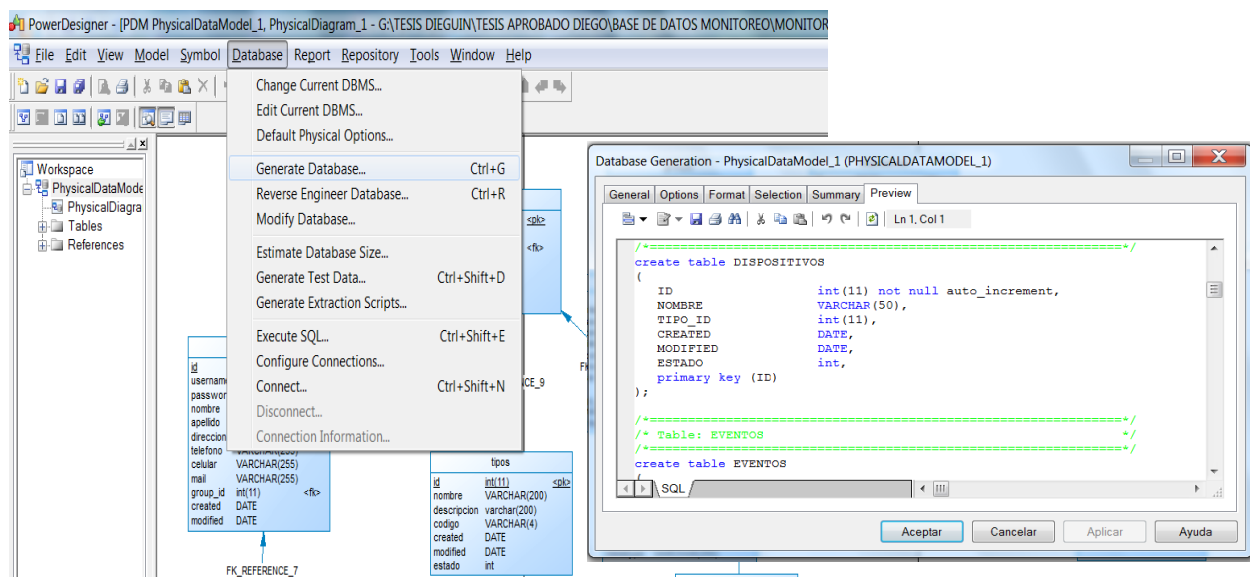
ANEXO 7: Implementación de la Base de Datos

A continuación se describe los pasos que se siguió para la implementación de la de base de datos.

PASO 1

Se procedió a generar el código sql desde la herramienta de diseño de base de datos Power Designer 12.5 en la que se diseñó la Base de datos de monitoreo.

Generación sql: Cargamos el diseño de la base de datos a Power Designer, vamos a la barra de herramientas opción Databases (Seleccionar Generate Databases ... - Ir a pestaña Preview), copiar todo el sql proporcionado.



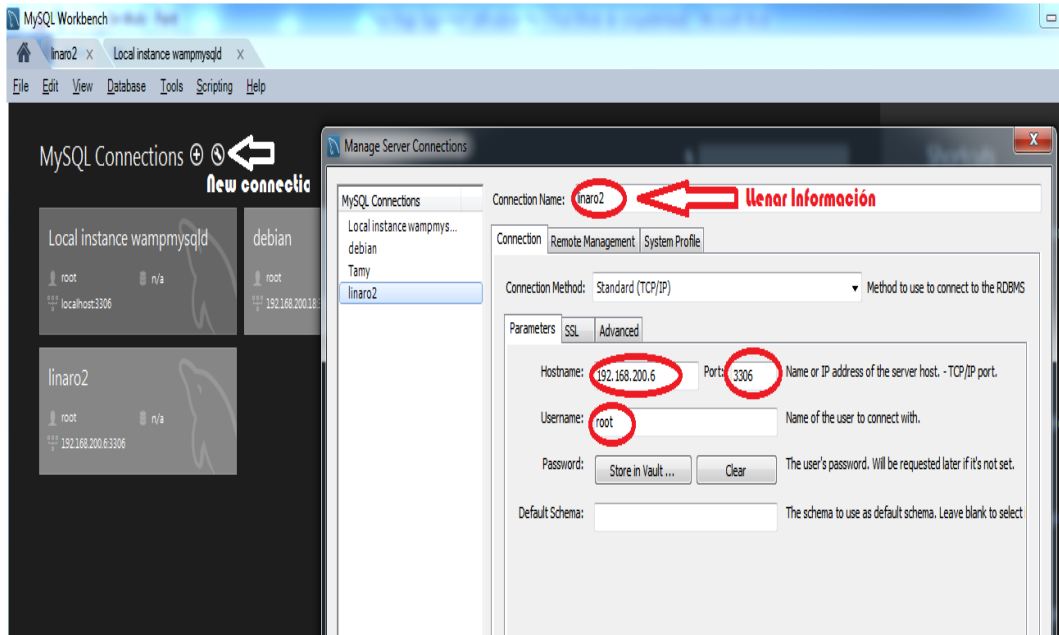
Pantalla 4: Generación Código SQL
Elaborado por: Diego Poaquiza - Investigador

PASO 2

Para la implementación del código sql generado, se instaló la herramienta visual de diseño de bases de datos llamado MySQL Wokbench versión 6.1 la cual es utilizada para una mejor administración de bases de datos a través de su interfaz de fácil manejo.

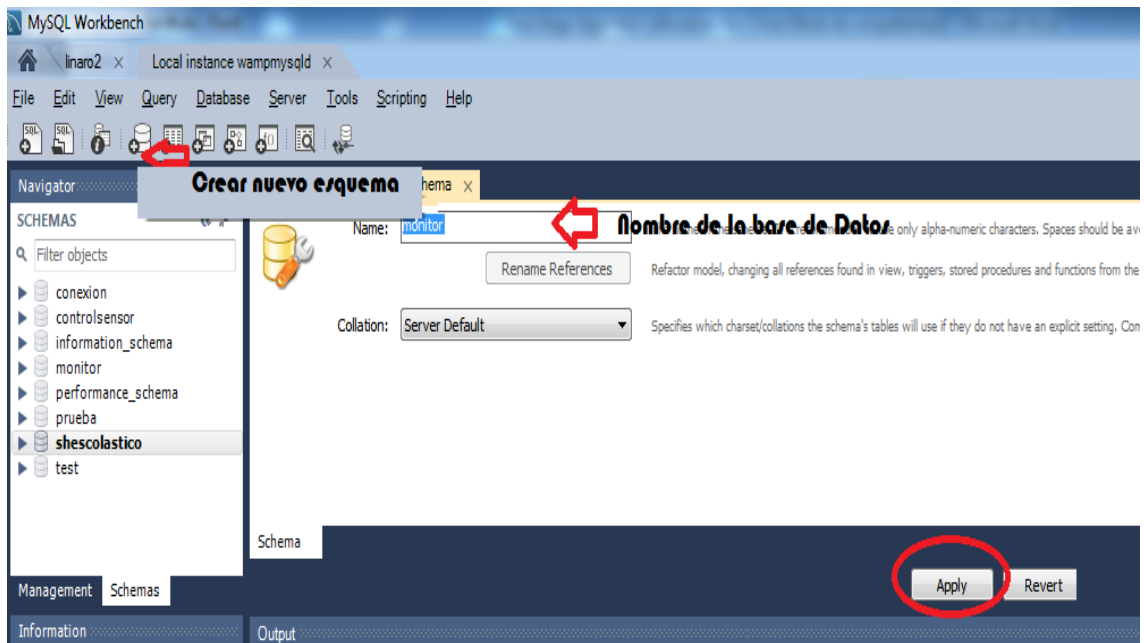
Una vez instalado se hizo una conexión remota hacia la base de datos instalada en el servidor para su administración.

Conexión remota: Vamos a New Connection y llenamos la información correcta del servidor para la conexión a mysql.



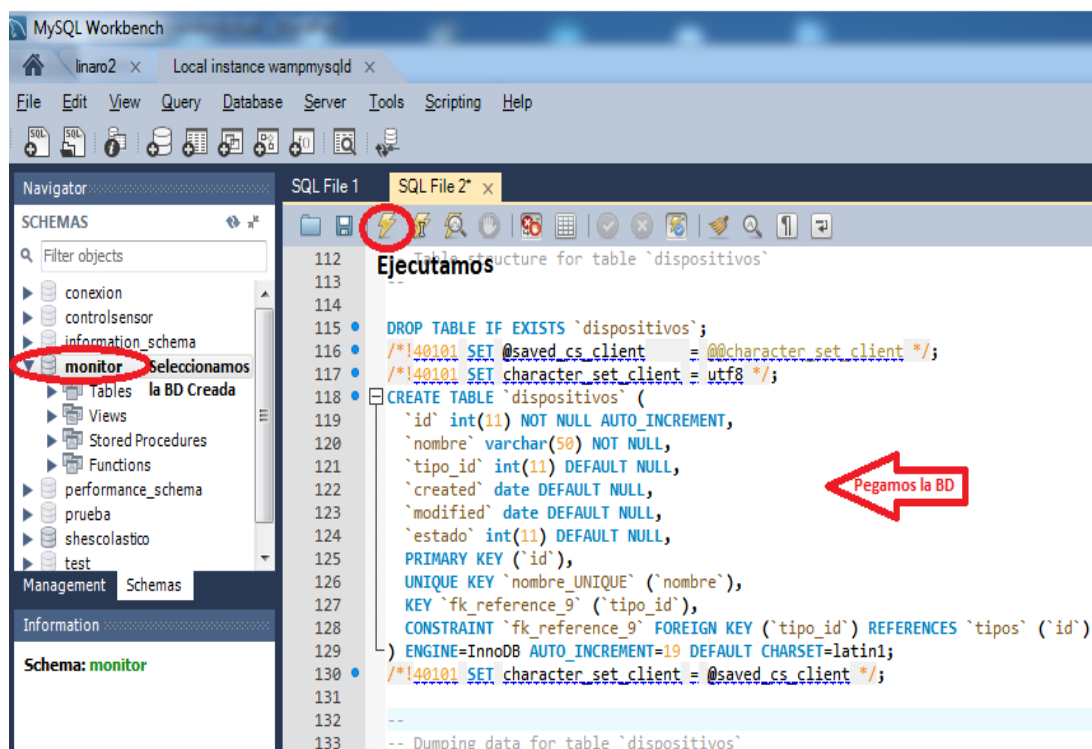
Pantalla 5: Conexión Remota Base de Datos Mysql
Elaborado por: Diego Poaquiza - Investigador

Una vez realizada la conexión se procedió a crear la base de datos (Para ello nos dirigimos a Crear un nuevo esquema – introducimos el nombre y Aceptamos).



Pantalla 6: Creación Base de la Base de Datos
Elaborado por: Diego Poaquiza - Investigador

Ingresamos a la base de datos creada y nos dirigimos a Crear un Nuevo Sql (procedemos a pegar el sql generado en Power Designer - ejecutamos todo), refrescamos la tabla de esquemas y verificamos la creación de la tablas de la base de datos.



Pantalla 7: Implementación Base de Datos
Elaborado por: Diego Poaquiza - Investigador

Por último se incluye el código SQL de la creación de la base de datos, con el objetivo de proporcionar una ayuda al administrador de la aplicación web, en la estructura y tratamiento de la información.

```
CREATE TABLE `dispositivos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) NOT NULL,
  `tipo_id` int(11) DEFAULT NULL,
  `created` date DEFAULT NULL,
  `modified` date DEFAULT NULL,
  `estado` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `nombre_UNIQUE` (`nombre`),
  KEY `fk_reference_9` (`tipo_id`),
  CONSTRAINT `fk_reference_9` FOREIGN KEY (`tipo_id`) REFERENCES `tipos` (`id`)
)
CREATE TABLE `eventos` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `zonificacion_id` int(11) DEFAULT NULL,
```

```

`created` timestamp NULL DEFAULT NULL,
`modified` timestamp NULL DEFAULT NULL,
`estado` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_reference_13` (`zonificacion_id`),
CONSTRAINT `fk_reference_13` FOREIGN KEY (`zonificacion_id`) REFERENCES `zonificaciones`
(`id`)
)
CREATE TABLE `eventousuarios` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`fechahora` datetime DEFAULT NULL,
`evento` int(11) DEFAULT NULL,
`user_id` int(11) DEFAULT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
PRIMARY KEY (`id`)
)
CREATE TABLE `fotosuarios` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`filename` varchar(255) DEFAULT NULL,
`dir` varchar(255) DEFAULT NULL,
`mimetype` varchar(255) DEFAULT NULL,
`filesize` int(11) DEFAULT NULL,
`user_id` int(11) DEFAULT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
`estado` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_reference_15` (`user_id`)
)
CREATE TABLE `groups` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`name` varchar(100) NOT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
PRIMARY KEY (`id`)
)
CREATE TABLE `particiones` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`nombre` varchar(200) DEFAULT NULL,
`descripcion` varchar(200) DEFAULT NULL,
`zona_id` int(11) DEFAULT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
`estado` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_reference_14` (`zona_id`),
CONSTRAINT `fk_reference_14` FOREIGN KEY (`zona_id`) REFERENCES `zonas` (`id`)
)
CREATE TABLE `sistemaestados` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`estado` int(11) DEFAULT NULL,
`nombre` varchar(45) DEFAULT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
PRIMARY KEY (`id`)
)
CREATE TABLE `tipos` (

```



```

`id` int(11) NOT NULL AUTO_INCREMENT,
`nombre` varchar(200) DEFAULT NULL,
`descripcion` varchar(200) DEFAULT NULL,
`codigo` varchar(4) NOT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
`estado` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `codigo_UNIQUE` (`codigo`)
)
CREATE TABLE `users` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`username` varchar(255) NOT NULL,
`password` char(40) NOT NULL,
`nombre` varchar(255) DEFAULT NULL,
`apellido` varchar(255) DEFAULT NULL,
`direccion` varchar(255) DEFAULT NULL,
`telefono` varchar(255) DEFAULT NULL,
`celular` varchar(255) DEFAULT NULL,
`mail` varchar(255) DEFAULT NULL,
`group_id` int(11) NOT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_reference_1` (`group_id`),
CONSTRAINT `fk_reference_1` FOREIGN KEY (`group_id`) REFERENCES `groups` (`id`)
)
CREATE TABLE `zonas` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`nombre` varchar(50) DEFAULT NULL,
`descripcion` varchar(100) DEFAULT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
`estado` int(11) DEFAULT NULL,
PRIMARY KEY (`id`)
)
CREATE TABLE `zonificaciones` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`dispositivo_id` int(11) DEFAULT NULL,
`particione_id` int(11) DEFAULT NULL,
`valor` varchar(5) DEFAULT NULL,
`created` date DEFAULT NULL,
`modified` date DEFAULT NULL,
`estado` int(11) DEFAULT NULL,
PRIMARY KEY (`id`),
KEY `fk_reference_12` (`particione_id`),
KEY `fk_reference_8` (`dispositivo_id`),
CONSTRAINT `fk_reference_12` FOREIGN KEY (`particione_id`) REFERENCES `particiones` (`id`),
CONSTRAINT `fk_reference_8` FOREIGN KEY (`dispositivo_id`) REFERENCES `dispositivos`
(`id`)
)

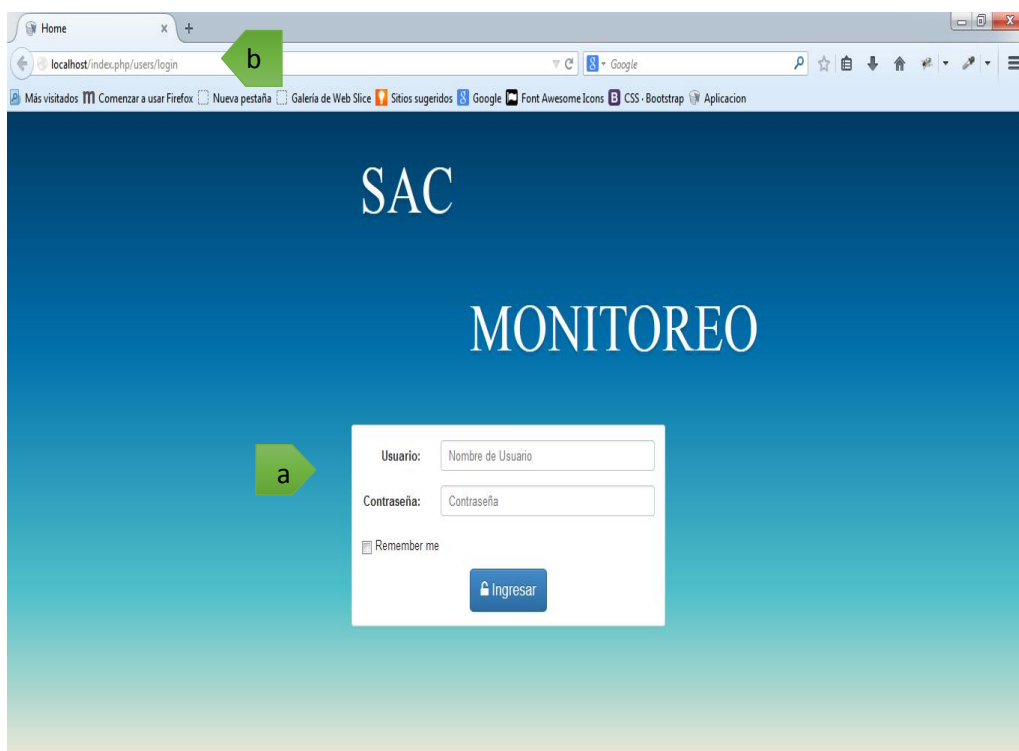
```

ANEXO 11: MANUAL DE USUARIO

El objetivo de este manual es proporcionar a los usuarios finales una guía para el correcto manejo de la aplicación web, explicando los controles y las funcionalidades de cada una de las páginas que componen dicha aplicación.

Pantalla de Inicio de Sesión.

Proporciona información, a la vez que posibilita el Inicio de Sesión a los diferentes usuarios de la aplicación web, con sus respectivos privilegios.



Pantalla 8: Inicio de Sesión
Fuente: Aplicación Web de Monitoreo

a) Inicio de Sesión: Consta de controles necesarios para el ingreso y validación de los datos del usuario de la aplicación.

b) Navegador: Muestra la url de navegación de la Aplicación Web de monitoreo.

Pantalla Principal.

Contiene el menú que permite realizar todas las tareas correspondientes a la aplicación web, de acuerdo a los privilegios del usuario que se validó.



Pantalla 9: Página Principal
Fuente: Aplicación Web de Monitoreo

a) Menú: Se muestra el menú de usuario administrador con sus respectivos iconos representativos respecto al menú.

b) Nombre del Formulario: En esta sección se muestra el nombre del formulario dependiendo de la opción del menú elegido.

c) Área de Contenido: Sección destinada para la gestión de la información. La información visualizada depende de las opciones del menú.

d) Usuario: Usuario: Se muestra la foto y el nombre de usuario logueado.

e) Menú de opciones: Consta de tres ítems:

➤ **Configurar cuenta:**

Permite dirigir a la página respectiva de configuración de la información personal del usuario logueado.

➤ **Acerca de:**

Muestra un cuadro modal sobre la información de la aplicación web.

Salir:

➤ Permite salir de la aplicación.

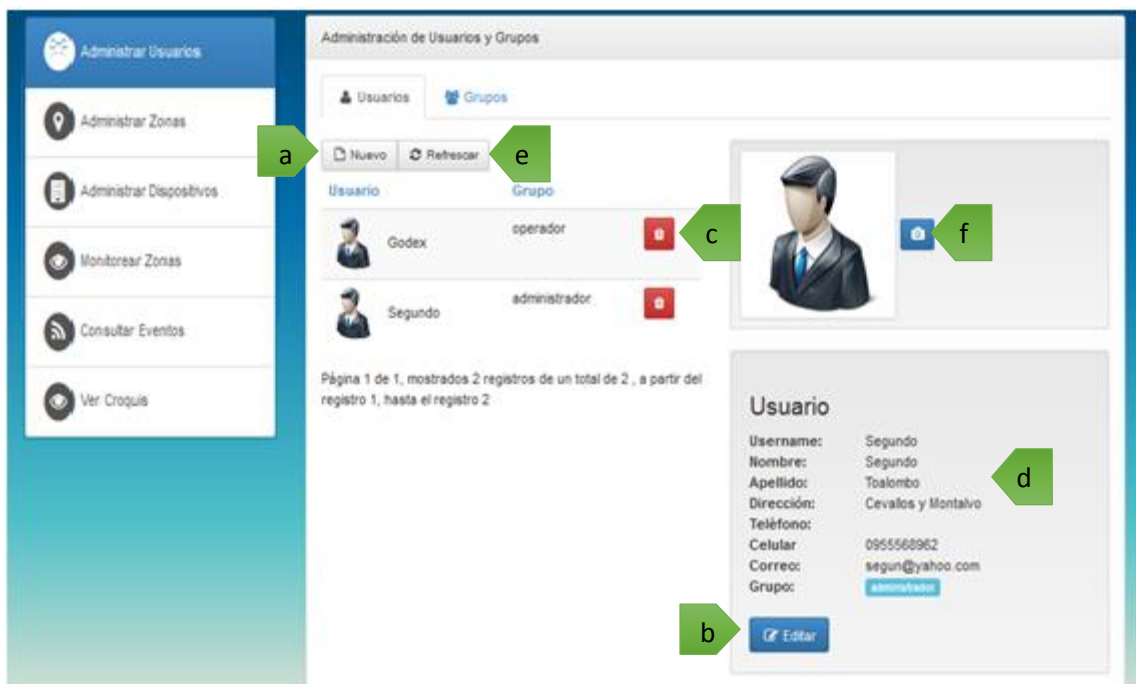
A continuación se detalla cada una de las páginas a las que se accede con cada opción del menú de usuario, en este caso la opción Administrador:

Pantalla Administración de Usuarios y Grupos.

Permite la inserción, consulta, modificación y eliminación de los usuarios y grupos, asignándoles los respectivos privilegios a los usuarios para la interacción con la aplicación web.

Pestaña Usuarios.

Nota: Para la Pestaña Grupos será el mismo proceso de administración que la de los usuarios.



Pantalla 10: Administración de usuarios
Fuente: Aplicación Web de Monitoreo

a) Botón Nuevo: Contiene los controles necesarios para la inserción de un nuevo usuario y la asignación de un rol.

b) Botón Editar: Muestra un cuadro modal para la modificación de la información de los usuarios.

c) Botón Eliminar: Elimina a un usuario determinado de la base de datos.

- d) **Lista de información:** Muestra la información detallada del usuario seleccionado.
- e) **Botón Refrescar:** Actualiza la tabla total de la información de usuarios.
- f) **Botón cambio de foto:** Permite cambiar la foto del usuario seleccionado.

Cuadro Modal de Ingreso de Usuarios.

Se visualiza al dar clic en el botón nuevo y posibilita el ingreso de un nuevo usuario.

The image shows a modal window titled "Nuevo Usuario". It contains the following fields:

- Username: Ingrese Nombre de Usuario
- Contraseña: Ingrese su contraseña
- Nombres: Ingrese sus Nombres
- Apellidos: Ingrese sus Apellidos
- Dirección: Ingrese su Dirección
- Teléfono: Ingrese su número telefonico
- Celular: Ingrese su número de celular
- Email: Ingrese su correo electrónico
- Rol: root (dropdown menu)

At the bottom, there are two buttons: "Guardar" (blue) and "Cancelar" (grey). A close button "x" is in the top right corner. Green callout boxes 'a', 'b', and 'c' are placed around the form: 'a' points to the input fields, 'b' points to the "Guardar" button, and 'c' points to the "Cancelar" button.

Pantalla 11: Ingreso de usuario
Fuente: Aplicación Web de Monitoreo

- a) **Datos a Ingresar:** Permite la digitación de información del usuario a ingresar con sus respectivos controles necesarios.
- b) **Botón Guardar:** Guarda la información digitada del usuario.
- c) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Modificación de Usuarios.

Se visualiza al dar clic en el botón editar y posibilita la edición de un usuario registrado.

Pantalla 12: Cuadro Modal Modificación de usuarios
Fuente: Aplicación Web de Monitoreo

- a) **Datos a Modificar:** Permite la modificar la información del usuario con sus respectivos controles necesarios.
- b) **Botón Guardar:** Guarda la información editada del usuario.
- c) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Eliminar Usuario.

Se visualiza al dar clic en el botón eliminar y posibilita la eliminación de un usuario registrado.

Pantalla 13: Cuadro Modal Eliminar usuario
Fuente: Aplicación Web de Monitoreo

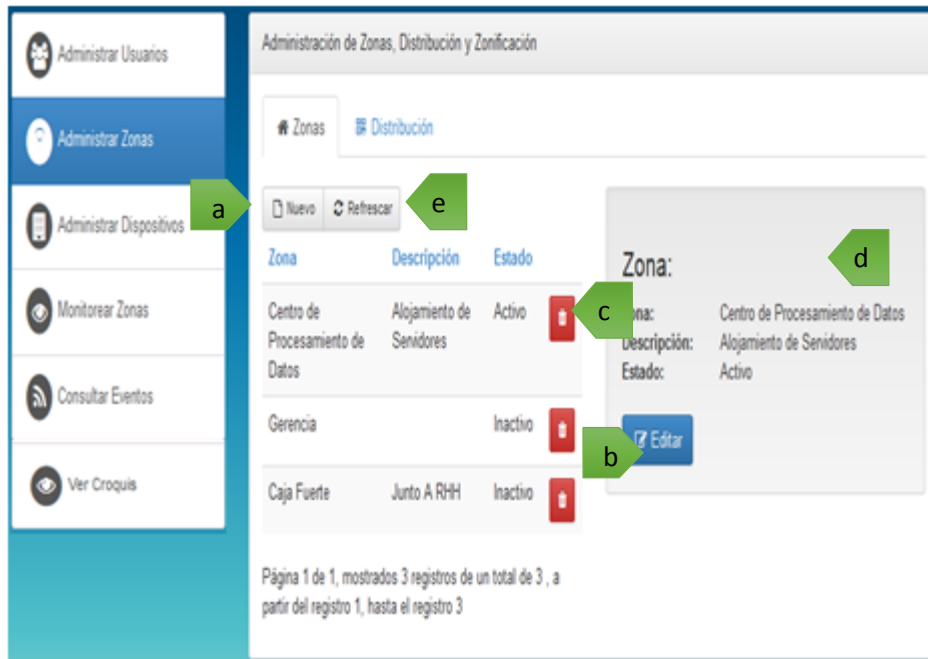
- a) **Botón Eliminar:** Elimina el usuario seleccionado de la base de datos.
- b) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Pantalla Administración Zonas y Distribución.

Permite la inserción, consulta, modificación y eliminación de las zonas con sus respectivas distribuciones para cada zona.

Pestaña Zonas.

Nota: Para la pestaña Distribución será el mismo proceso de administración que la de las Zonas.



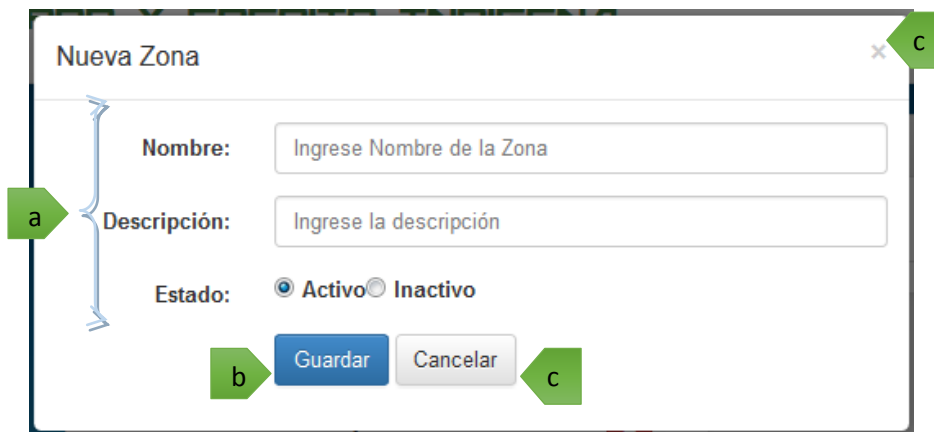
Pantalla 14: Administración Zonas y Distribución

Fuente: Aplicación Web de Monitoreo

- a) **Bono Nuevo:** Contiene los controles necesarios para la inserción de una nueva zona.
- b) **Botón Editar:** Muestra un cuadro modal para la modificación de la información de la zona seleccionada.
- c) **Botón Eliminar:** Elimina a una zona determinada de la base de datos.
- d) **Lista de información:** Muestra la información detallada de la zona seleccionada.
- e) **Botón Refrescar:** Actualiza la tabla total de la información de zonas.

Cuadro Modal de Ingreso de Zonas.

Se visualiza al dar clic en el botón nuevo y posibilita el ingreso de una nueva zona.

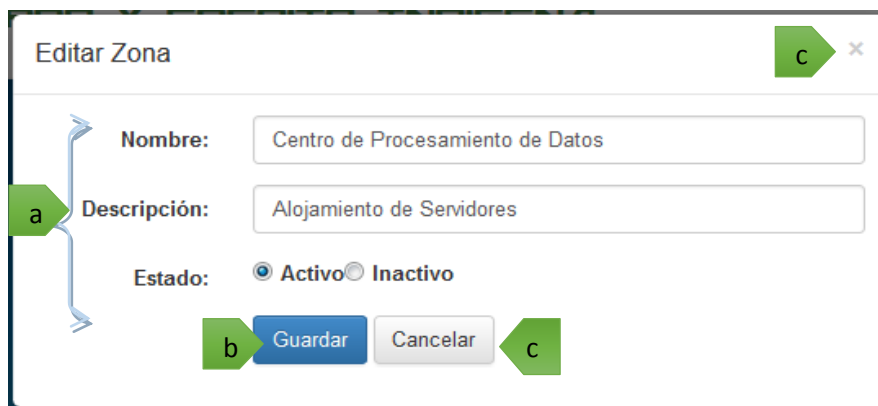


Pantalla 15: Cuadro Modal ingreso de zonas
Fuente: Aplicación Web de Monitoreo

- a) **Datos a Ingresar:** Permite la digitación de información de la zona a ingresar con sus respectivos controles necesarios.
- b) **Botón Guardar:** Guarda la información digitada de la zona.
- c) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Modificación de Zonas.

Se visualiza al dar clic en el botón editar y posibilita la edición de una zona registrada en la base de datos.

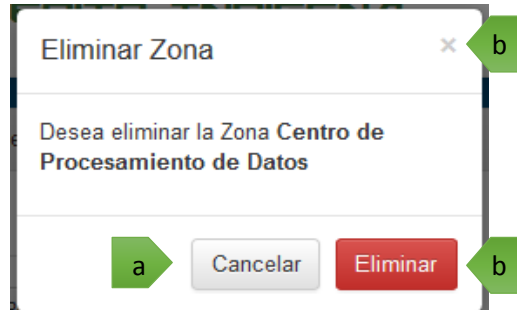


Pantalla 16: Cuadro Modal Modificación de Zonas
Fuente: Aplicación Web de Monitoreo

- a) **Datos a Modificar:** Permite la modificar la información de la zona con sus respectivos controles necesarios.
- b) **Botón Guardar:** Guarda la información editada de la zona.
- c) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Eliminar Zona.

Se visualiza al dar clic en el botón eliminar y posibilita la eliminación de una zona registrada.



Pantalla 17: Cuadro Modal Eliminar Zona
Fuente: Aplicación Web de Monitoreo

a) **Botón Eliminar:** Elimina la zona seleccionada de la base de datos.

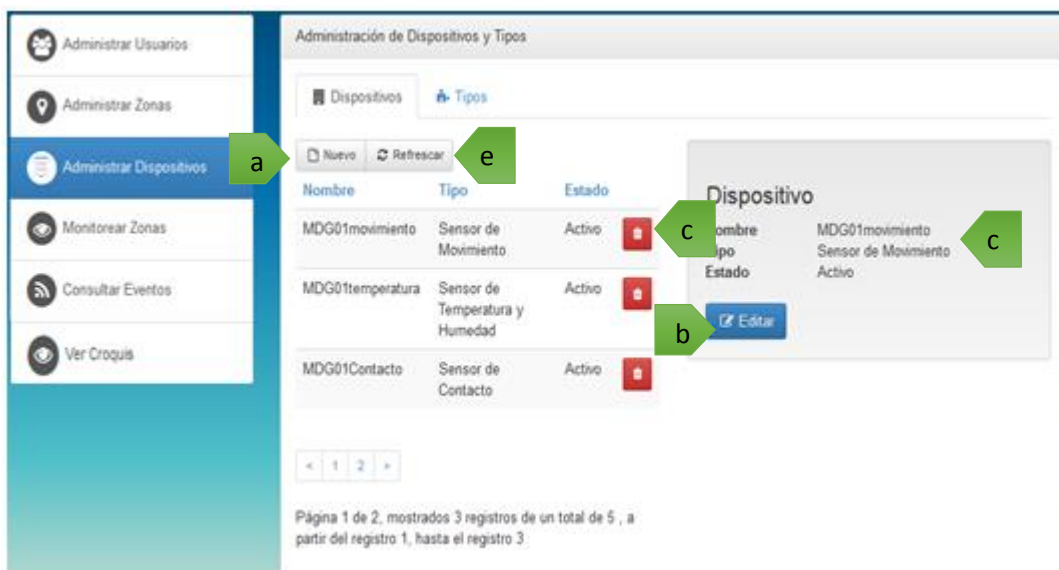
b) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Pantalla Administración de Dispositivos y Tipos.

Permite la inserción, consulta, modificación y eliminación de los dispositivos con sus respectivas tipos por cada dispositivo.

Pestaña Dispositivos.

Nota: Para la pestaña Tipos será el mismo proceso de administración que la de los Dispositivos.



Pantalla 18: Administración de Dispositivos y Tipos
Fuente: Aplicación Web de Monitoreo

- a) **Bono Nuevo:** Contiene los controles necesarios para la inserción de un nuevo dispositivo y la asignación de un tipo.
- b) **Botón Editar:** Muestra un cuadro modal para la modificación de la información de los dispositivos.
- c) **Botón Eliminar:** Elimina a un dispositivo determinado de la base de datos.
- d) **Lista de información:** Muestra la información detallada del dispositivo seleccionado.
- e) **Botón Refrescar:** Actualiza la tabla total de la información de dispositivos.

Cuadro Modal de Ingreso de Dispositivos.

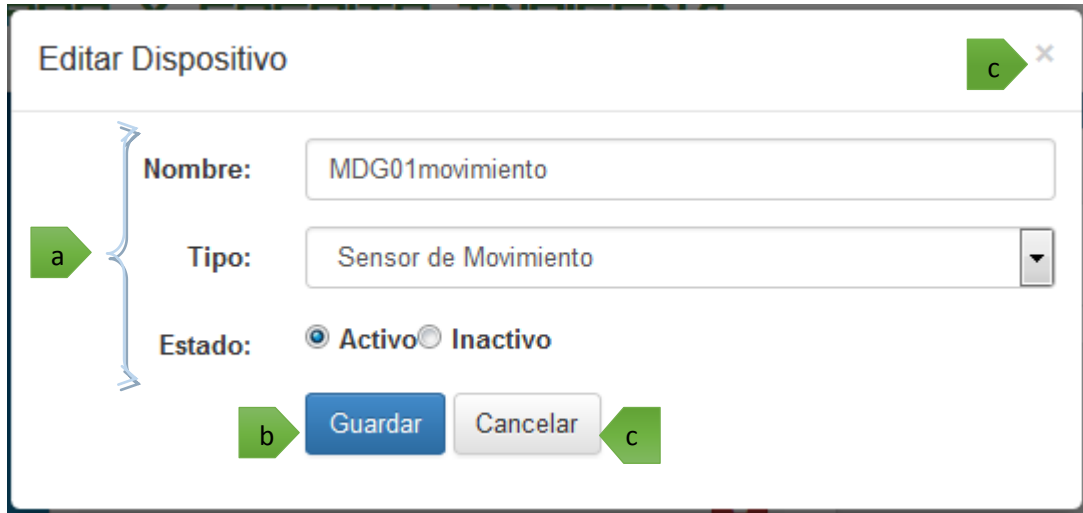
Se visualiza al dar clic en el botón nuevo y posibilita el ingreso de un nuevo dispositivo.

Pantalla 19: Cuadro Modal de Ingreso de Dispositivos
Fuente: Aplicación Web de Monitoreo

- a) **Datos a Ingresar:** Permite la digitación de información del dispositivo a ingresar con sus respectivos controles necesarios.
- b) **Botón Guardar:** Guarda la información digitada del dispositivo.
- c) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Modificación de Dispositivo.

Se visualiza al dar clic en el botón editar y posibilita la edición de un dispositivo registrado.

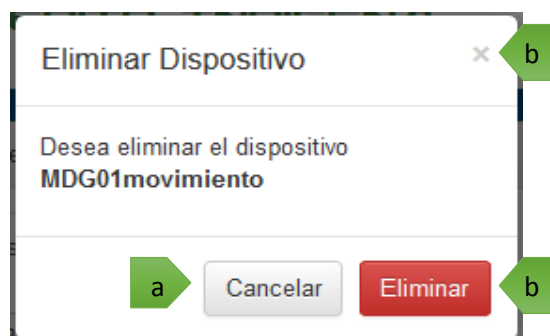


Pantalla 20: Cuadro Modal Modificación de Dispositivo
Fuente: Aplicación Web de Monitoreo

- a) **Datos a Modificar:** Permite la modificar la información del dispositivo con sus respectivos controles necesarios.
- b) **Botón Guardar:** Guarda la información editada del dispositivo.
- c) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Eliminar Dispositivo.

Se visualiza al dar clic en el botón eliminar y posibilita la eliminación de un dispositivo registrado.



Pantalla 21: Cuadro Modal Eliminar Dispositivo
Fuente: Aplicación Web de Monitoreo

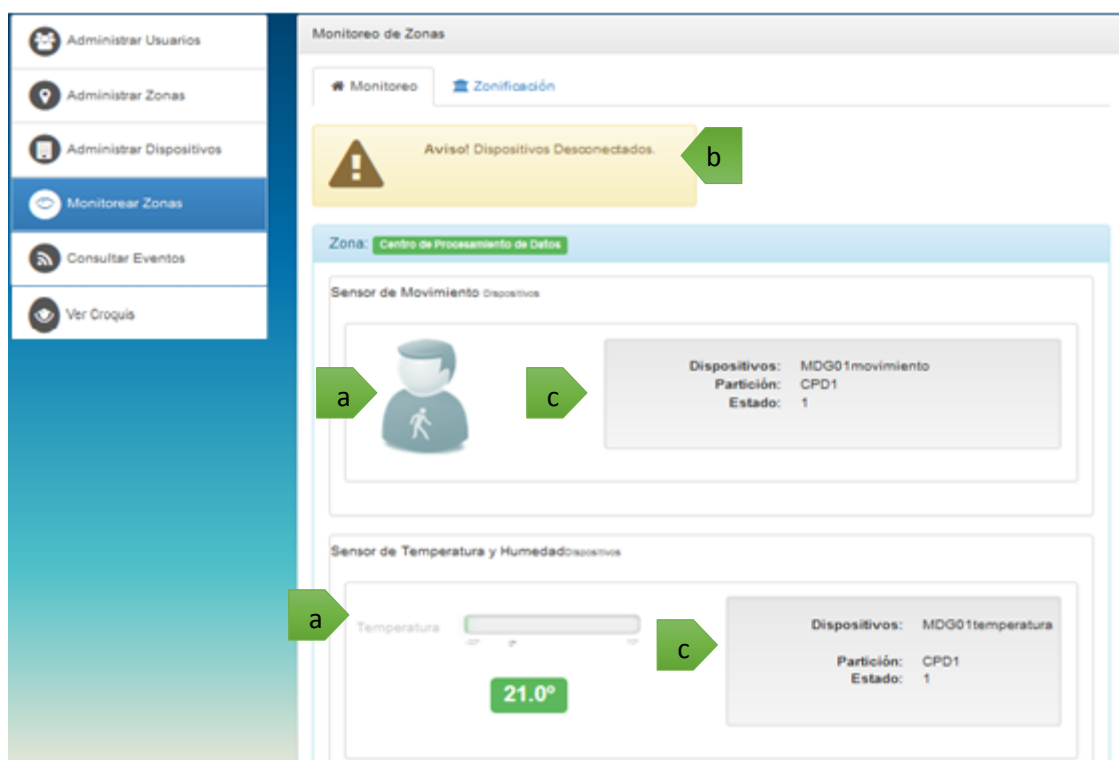
Botón Eliminar: Elimina el dispositivo seleccionada de la base de datos.

b) **Botón Cancelar y x:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Pantalla Monitoreo de Zonas y Zonificación.

Pestaña Monitoreo:

Permite al usuario activar o desactivar el sistema de alarma y realizar el monitoreo del Centro de Procesamiento de Datos mediante los sensores instalados.



Pantalla 22: Monitoreo de Zonas
Fuente: Aplicación Web de Monitoreo

a) **Área Imagen Dispositivo:** Visualiza la imagen de los dispositivos instalados en el CPD, la imagen cambia dependiendo del estado y tipo.

b) **Estados:** Muestra el estado actual del Sistema de Alarma, además muestra un botón que permite activar o desactivar el sistema siempre y cuando los dispositivos estén conectados al sistema.

c) **Área de información:** Contiene la información de la zonificación de los dispositivos instalados en el CPD, mostrando la información de su distribución y estado, la información depende del tipo de dispositivo.

Pestaña Zonificación:

En esta sección se muestra la información de la zonificación y permita al usuario la gestión de la misma de los dispositivos a instalarse en el CPD.

Dispositivo	Distribución	Estado
MDG01Contacto	Puerta A1	Activo
MDG01Contacto2	Ventana V1	Activo
MDG01temperatura	CPD1	Activo
MDG01movimiento	CPD1	Activo
MDG01alarma	CPD1	Activo

Página 1 de 1, mostrados 5 registros de un total de 5 , a partir del registro 1, hasta el registro 5

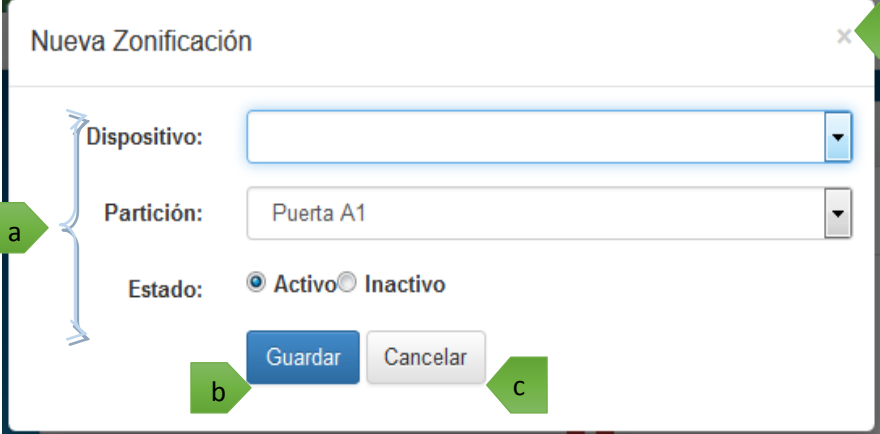
Pantalla 23: Administración de zonificación

Fuente: Aplicación Web de Monitoreo

- a) **Bono Nuevo:** Contiene los controles necesarios para la inserción de una nueva zonificación.
- b) **Botón Editar:** Muestra un cuadro modal para la modificación de la información de la zonificación seleccionada.
- c) **Botón Eliminar:** Elimina a una zonificación determinada de la base de datos.
- d) **Lista de información:** Muestra la información detallada de la zonificación seleccionada.
- e) **Botón Refrescar:** Actualiza la tabla total de la información de zonificación.

Cuadro Modal de Ingreso de Zonificación.

Se visualiza al dar clic en el botón nuevo y posibilita el ingreso de una nueva zonificación.



Pantalla 24: Cuadro Modal de Ingreso de Zonificación
Fuente: Aplicación Web de Monitoreo

a) Datos a Ingresar: Permite la digitación de información de la zonificación a ingresar con sus respectivos controles necesarios.

b) Botón Guardar: Guarda la información digitada de la zonificación.

c) Botón Cancelar y x: Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Modificación de Zonificación.

Se visualiza al dar clic en el botón editar y posibilita la edición de una zonificación registrada en la base de datos.



Pantalla 25: Cuadro Modal Modificación de Zonificación
Fuente: Aplicación Web de Monitoreo

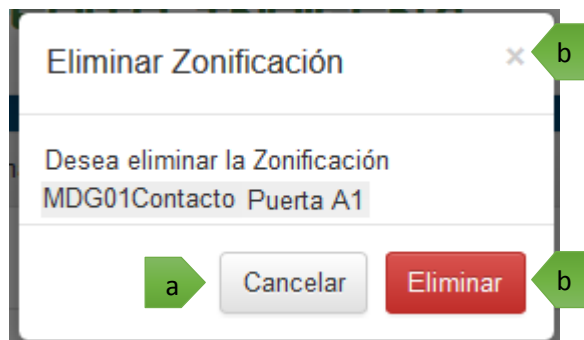
a) Datos a Modificar: Permite la modificar la información de la zonificación con sus respectivos controles necesarios.

b) Botón Guardar: Guarda la información editada de la zonificación.

c) Botón Cancelar y x: Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Cuadro Modal Eliminar Zonificación.

Se visualiza al dar clic en el botón eliminar y posibilita la eliminación de una zonificación registrada.



Pantalla 26: Cuadro Modal Eliminar Zonificación
Fuente: Aplicación Web de Monitoreo

a) Botón Eliminar: Elimina la zonificación seleccionada de la base de datos.

b) Botón Cancelar y x: Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Pantalla configuración de la cuenta de usuario

Nos permite configurar la información de la cuenta del usuario logueado.



Pantalla 27: Configuración de la cuenta de usuario
Fuente: Aplicación Web de Monitoreo

- a) **Usuario:** Se visualiza el nombre y la foto del usuario logueado.
- b) **Área de datos:** Se muestra una la lista de la información personal del usuario logueado.
- c) **Área actualización de la información:** Se visualiza la información detalla de la fila seleccionada por el usuario para su respectiva actualización.
- d) **Botón Guardar:** Verifica la validación de datos y procede al guardado de los mismos.
- e) **Botón Cancelar:** Cierra el cuadro modal sin realizar ningún cambio sobre la base de datos.

Pantalla Consultar Eventos

Nos permite configurar la información de la cuenta del usuario logueado.

Permite la consulta de los eventos suscitados en el CPD de los dispositivos de seguridad instalados en el lugar.

Pestaña Eventos Dispositivos.

Consultar Eventos

← Eventos Dispositivos ← Eventos Usuarios

Reporte Refrescar Buscar por fecha ... q

Zonificación	Fecha	Estado
MDG01movimiento	2015-01-15 15:23:01	Activado
MDG01temperatura	2015-02-24 12:18:46	27 Grados
MDG01temperatura	2015-04-15 22:02:46	22 Grados
MDG01temperatura	2015-04-15 22:02:49	21 Grados

Página 1 de 1, mostrados 4 registros de un total de 4 , a partir del registro 1, hasta el registro 4

Pantalla 28: Eventos Dispositivos
Fuente: Aplicación Web de Monitoreo

- a) **Botón Refrescar:** Actualiza la tabla total de la información de los eventos.
- b) **Buscar:** Permite realizar búsqueda personalizada por fecha de la información sobre los eventos suscitados.
- b) **Reporte:** Permite generar reportes personalizados de los eventos suscitados depende de la búsqueda realizada.

Pestaña Eventos Usuarios.

Consultar Eventos

Eventos Dispositivos | Eventos Usuarios

Reporte | Refrescar | Buscar por usuario ...

Fecha-Hora	Acción	Usuario
2014-12-02 18:01:04	Ingreso Sistema	Diego Poaquiza
2010-01-01 09:11:04	Ingreso Sistema	Segundos Toalombo
2010-01-01 09:11:26	Salio Sistema	Segundos Toalombo
2015-04-16 01:51:03	Ingreso Sistema	Diego Poaquiza

Página 1 de 1, mostrados 4 registros de un total de 4 , a partir del registro 1, hasta el registro 4

Pantalla 29: Eventos Dispositivos
Fuente: Aplicación Web de Monitoreo

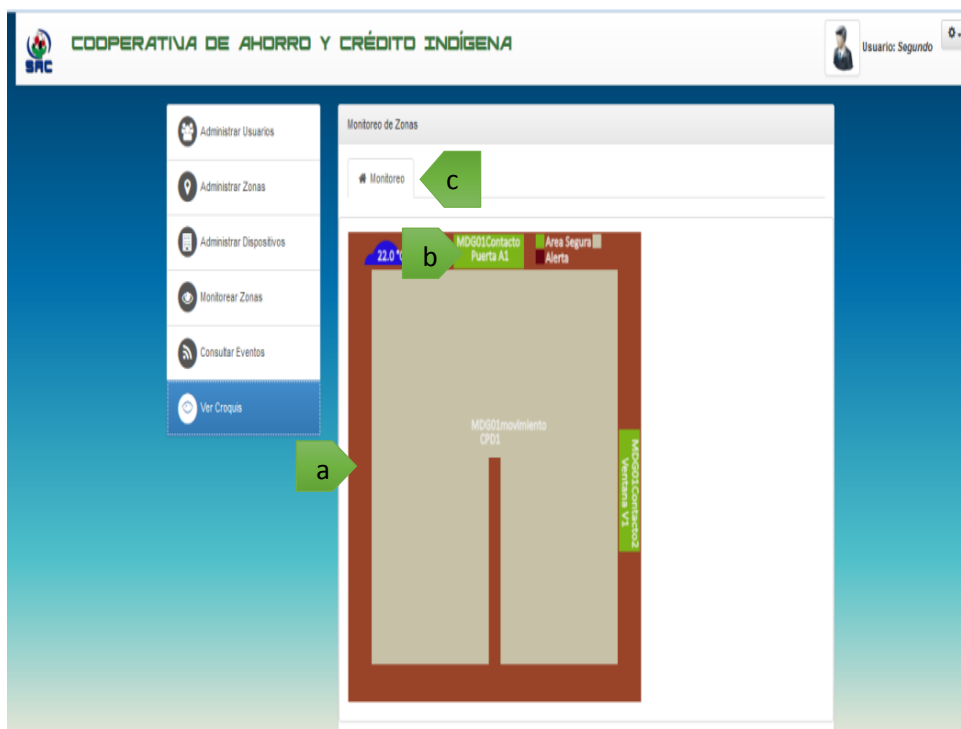
a) Botón Refrescar: Actualiza la tabla total de la información de los eventos realizados por los usuarios.

b) Buscar: Permite realizar búsqueda personalizada por fecha de la información sobre los eventos de los usuarios.

b) Reporte: Permite generar reportes personalizados de los eventos realizados por los usuarios, depende de la búsqueda realizada.

Pestaña Ver Croquis.

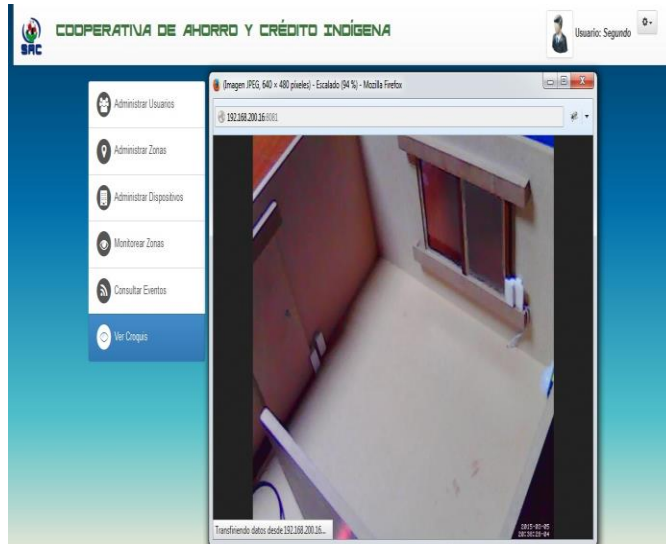
Permite al usuario realizar el monitoreo del CPD respectivo por medio de una cámara y un croquis el cual muestra el estado de mismo de acuerdo a la información arrojada por los sensores.



Pantalla 30: Monitoreo mediante Croquis y Cámara Web
Fuente: Aplicación Web de Monitoreo

- a) **Croquis:** Visualiza al usuario el croquis del CPD con la respectiva información de los dispositivos instalados.
- b) **Área de Información:** Se muestra la lista de información con los estados de la información emitida por los respectivos sensores.
- c) **Pestaña Módulo:** Se muestra el nombre de la pestaña a interactuar, además que al dar click sobre el mismo permite abrir una ventana emergente con la visualización de la imagen ofrecida por la cámara instalada en el CPD.

Ventana emergente con la información emitida por la Cámara web.



Pantalla 31: Ventana emergente cámara web
Fuente: Aplicación Web de Monitoreo

ANEXO 10: Infraestructura del Centro de Procesamiento de Datos Cooperativa Indígena “SAC”

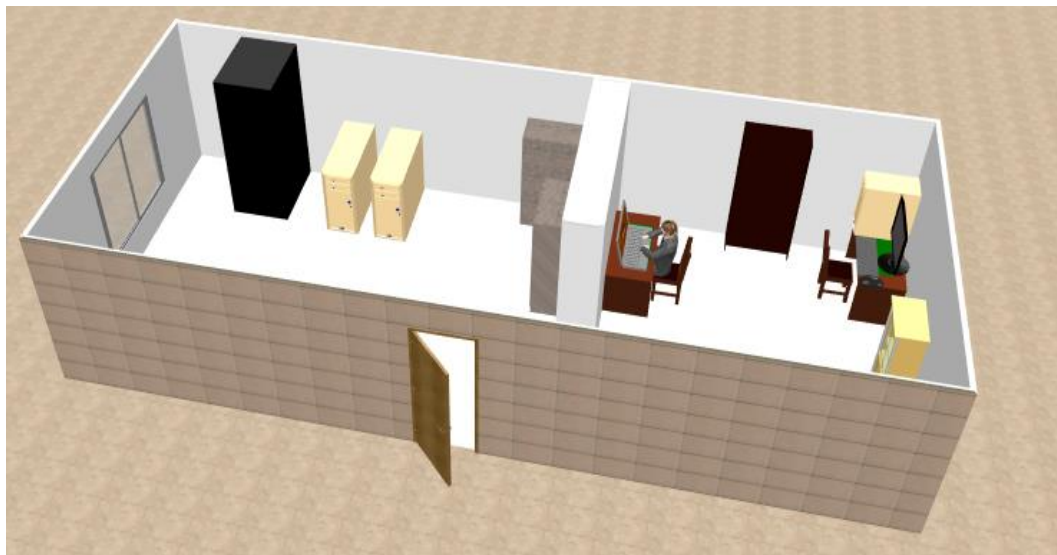


Fig. 79: Infraestructura CPD Cooperativa SAC
Fuente: Diego Poaquiza - Investigador

ANEXO 9: Fotos



Foto 1: Armado Arduino y Sensores
Fuente: Diego Poaquiza - Investigador



Foto 2: Armado Arduino y Sensores
Fuente: Diego Poaquiza - Investigador



Foto 3: Armado Arduino y Sensores
Fuente: Diego Poaquiza - Investigador