



UNIVERSIDAD DEL SINÚ

Elías Bechara Zainúm

Seccional Cartagena

DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA IDENTIFICACIÓN DE
PLACAS VEHICULARES Y RECONOCIMIENTO DE CARACTERES EN TIEMPO
REAL IMPLEMENTADO EN RASPBERRY PI 3 PARA LA UNIVERSIDAD DEL
SINÚ SECCIONAL CARTAGENA SEDE SANTILLANA

Presentado por:

MOISES ARNEDO MENDOZA

ARNOLD CAICEDO PACHECO

MARIA ANTONIA FUENTES VILLERA

UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL CARTAGENA

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA-COLOMBIA

JUNIO-2018



DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA IDENTIFICACIÓN DE
PLACAS VEHICULARES Y RECONOCIMIENTO DE CARACTERES EN TIEMPO
REAL IMPLEMENTADO EN RASPBERRY PI 3 PARA LA UNIVERSIDAD DEL
SINÚ SECCIONAL CARTAGENA SEDE SANTILLANA

Trabajo de grado presentado como requisito para optar el título de
INGENIERO DE SISTEMAS

Asesor disciplinar

LUIS ALFREDO BLANQUICETT B.

LUIS MURILLO FERNÁNDEZ

Asesor metodológico

EUGENIA ARRIETA RODRÍGUEZ

UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL CARTAGENA
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA-COLOMBIA
JUNIO-2018

Acta de Calificación y aprobación

Notas de aceptación

Director de escuela

Director de investigaciones

Firma de jurado

Firma de jurado

Cartagena de indias, 2018

**EL DIRECTOR DE INVESTIGACIONES DE LA UNIVERSIDAD DEL SINU “ELIAS
BECHARA ZAINUM” SECCIONAL CARTAGENA**

HACE CONSTAR QUE:

En Cartagena, a los 02 días del mes de Junio del 2018, en la Oficina de la Dirección de Investigaciones de la Universidad, se aprobó por el jurado y se realizó la sustentación del Trabajo de Grado titulado DISEÑO Y DESARROLLO DE UN PROTOTIPO PARA IDENTIFICACIÓN DE PLACAS VEHICULARES Y RECONOCIMIENTO DE CARACTERES EN TIEMPO REAL IMPLEMENTADO EN RASPBERRY PI 3 PARA LA UNIVERSIDAD DEL SINÚ SECCIONAL CARTAGENA SEDE SANTILLANA que se desarrolló bajo la dirección de los Ingenieros LUIS ALFREDO BLANQUICETT y LUIS MURILLO presentado por los egresados MOISES ARNEADO MENDOZA, ARNOLD CAICEDO PACHECO Y MARIA ANTONIA FUENTES VILLERA.

Los jurados designados fueron los ingenieros MARIA CLAUDIA BONFANTE y AISNER MARRUGO.

Teniendo en cuenta la aprobación emitida, se encuentra que los estudiantes han cumplido con los requisitos de presentación y sustentación del trabajo de investigación, exigidos por el programa de INGENIERÍA DE SISTEMAS, Resolución 0178 de 15 de marzo de 2010.

Se expide esta constancia a los 06 días del mes de junio del 2018.

DIRECCIÓN DE INVESTIGACIONES
Universidad del Sinú

COORDINADOR DE INVESTIGACIONES
Escuela de Ingeniería de Sistemas

DEDICATORIAS

En primer lugar, agradecer a Dios quien me guía por el camino correcto, quien me ha llenado de fortaleza y sabiduría en momentos de dificultades en mi vida y que en todo momento está conmigo ayudándome aprender de los errores para no volver a cometerlos.

A mi padre Moises Arnedo Padilla y mi madre Nelcy Mendoza De Ávila quienes me han enseñado todo en la vida, siempre me han brindado su apoyo incondicional, gracias por haberme forjado como la persona que soy en la actualidad, todos mis logros se los debo a ustedes incluyendo este. Este es uno de los pagos por tantos esfuerzos que han hecho por mi y mis hermanos, a pesar de tantas dificultades ustedes jamás nos han desamparado, siempre haciendo hasta lo que no estaba en sus manos para brindarnos apoyo y por eso quiero dedicarles este logro, porque esto les pertenece más a ustedes que a mí, Los amo demasiado, Muchas Gracias.

A mis Hermanos Rafael Santos Arnedo Mendoza y Jesús Camilo Arnedo Mendoza, quienes hacen parte de mi fuente de motivación para realizar todas mis cosas bien y tengan un buen ejemplo a seguir, espero que sean mejor de lo que yo he podido ser y les den más alegrías a nuestros padres alcanzando sus logros y metas.

A mis familiares que siempre han estado apoyándome y preocupándose por verme alcanzar este logro, en especial mi primo Estiven Arnedo Reyes quien en muchas oportunidades dejo sus oficios de lado para brindarme ayuda, muchas gracias primo eres una excelente persona espero verte pronto logrando tus metas.

A mi novia Mileidis Monroy, quien me ha apoyado y me ha aconsejado con mucha sabiduría desde que éramos amigos, me has enseñado lo importante que es soñar con los pies en la tierra, gracias por estar siempre cuando te he necesitado.

Gracias al grupo de profesores de la universidad del Sinú, he aprendido mucho de cada uno de ellos y en especial a los de este ultimo semestre quienes nos dieron su apoyo hasta el ultimo momento para lograr culminar con este proyecto, Muchas gracias por todo.

Y por último a mis compañeros de lucha Arnol Caicedo Pacheco y María Fuentes Villera, fue un año muy difícil en los que pasamos momentos buenos y malos, en los que en alguna vez pensamos en tirar la toalla, pero gracias a Dios nos dio la fortaleza de seguir con este proyecto y terminarlo como no los habíamos propuesto. Y aquí estamos celebrando nuestra gran victoria a todos nuestros esfuerzos, Felicidades Ingenieros, gracias por todo.

MOISES ARNEDO MENDOZA

En primera instancia le doy gracias a Dios por permitirme cumplir con esta meta y darme fuerzas para seguir adelante, a mis padres Wilfrido Caicedo Rodriguez y Arleth Pacheco Guerrero, a mi Hermana Shirli Caicedo Pacheco, a mi novia Claudia Palacios Morelo que fueron las personas que más estuvieron presente en esta lucha.

Aparte al grupo de profesores de la universidad del Sinú, los cuales siempre estuvieron pendientes en el proceso y nos brindaron de su apoyo y tiempo para poder salir adelante con este proyecto.

Tambien mis dedicatorias van hacia el grupo de trabajo de Hardsoft Computers, Shirli Puello, David Tres Palacios, Oscar Garcia y Luis Gonzalez, los cuales fueron fundamentales y me brindaron el tiempo y el apoyo necesario en todo este proceso.

Y por ultimo compañeros de la universidad que siempre estuvieron presentes y mostrando su apoyo.

ARNOLD CAICEDO PACHECO

Dios, perfecto y misericordioso gracias por este logro alcanzado.

A mi madre y padre, autores de este triunfo, por nunca abandonarme y ayudarme a alcanzar esta meta. Este triunfo es de ustedes, amores de mi vida.

Hija, los limites no existen, eres mi todo, mi inspiración y fuerza para seguir adelante.

Esposo, eres mi mejor amigo, compañero, amante, tu eres mi guía, me ayudas a no desfallecer y me motivas a seguir, te admiro amor de mi vida.

Hermanos, somos el triunfo de nuestros padres y este es un escalón pero vamos por más.

Primas y tía mi segunda familia, han sido mi ayuda desde siempre, las amo.

Agradezco a cada persona que me encontré en este proceso, porque de cada uno aprendí algo.

Gracias.

MARIA ANTONIA FUENTES VILLERA

AGRADECIMIENTOS

Muy agradecidos con nuestros tutores Eugenia Arrieta, Luis blanquicet y Luis Murillo, quienes nos extendieron su mano para brindarnos apoyo y quienes forjaron nuestro camino con mucha sabiduría y dándonos fortaleza cuando sentíamos que todo estaba perdido, gracias a ustedes hemos logrado está meta muy importante en la vida de cada uno de nosotros y de nuestras familias, sus consejos nos convirtieron en mejores personas, en todos unos profesionales y al igual que ustedes aplicaremos en nuestras vida esa enseñanza que nos quedó...

Muchas gracias por todo y que Dios derrame muchas bendiciones sobre ustedes y sus familias

TABLA DE CONTENIDO

Pág.

INTRODUCCIÓN	20
1 DISEÑO METODOLÓGICO.....	21
1.1 PLANTEAMIENTO DEL PROBLEMA.....	21
1.1.1 Descripción del problema.....	21
1.1.2 Justificación	22
1.1.3 Formulación del problema.....	22
1.1.4 Objetivos	23
1.2 ESTADO DEL ARTE.....	24
1.3 MARCOS DE REFERENCIA	26
1.3.1 Marco teórico	26
1.3.2 Marco conceptual.....	35
1.4 DISEÑO METODOLÓGICO.....	37
1.4.1 Línea de investigación.....	37
1.4.2 Tipo de investigación.....	37
1.4.3 Definición de la metodología	38
2 ANÁLISIS DEL PROBLEMA.....	41
2.1 REQUISITOS ESPECÍFICOS.....	41
2.2 REQUISITOS COMUNES DE LAS INTERFACES	49
2.3 REQUISITOS FUNCIONALES	50
2.4 REQUISITOS NO FUNCIONALES.....	51
3 DISEÑO DE LA SOLUCIÓN	52
3.1 DISEÑO DE LA APLICACIÓN WEB	52
3.1.1 Iniciar Sesión.....	52
3.1.2 Gestionar Usuarios.....	53
3.1.3 Gestionar Cámaras.....	53
3.1.4 Detecciones en línea	54
3.1.5 Historial de las Detecciones.....	55
3.2 DIAGRAMAS.....	56

3.2.1	Diagrama de flujo del prototipo.....	56
3.2.2	Diagrama Entidad – Relación.....	57
3.2.3	Modelo Relacional.....	58
3.2.4	Diagrama de casos de uso	59
3.2.5	Diagramas de secuencia	60
3.3	DISEÑO DEL PROTOTIPO DE RECONOCIMIENTO DE PLACAS.....	64
3.3.1	Arquitectura de red.....	64
4	DESARROLLO	65
4.1	DESARROLLO DE APLICACIÓN WEB	65
4.2	RASPBERRY PI 3 MODEL B.....	73
4.3	IDENTIFICACIÓN Y RECONOCIMIENTO DE PLACAS VEHICULARES.....	78
4.4	INTEGRACIÓN DE LOS COMPONENTES.....	82
5.	PRUEBAS DEL PROTOTIPO.....	83
5.1	PRUEBAS A LA APLICACIÓN WEB.....	83
5.1.1	Pruebas de interfaz gráfica para requerimientos Web.....	83
5.1.2	Pruebas de funcionalidad para los requerimientos.....	85
5.2	PRUEBA DE RECONOCIMIENTO DE PLACAS VEHICULARES.....	91
6	RECOMENDACIONES	94
7	CONCLUSIONES.....	95
	BIBLIOGRAFÍA.....	96
	ANEXOS.....	100

LISTAS D E TABLAS

	Pág.
Tabla 1 Requerimiento - RF01, Fuente Autor _____	41
Tabla 2 Requerimiento - RF02, Fuente Autor _____	41
Tabla 3 Requerimiento - RF03, Fuente Autor _____	42
Tabla 4 Requerimiento - RF04, Fuente Autor _____	42
Tabla 5 Requerimiento - RF05, Fuente Autor _____	43
Tabla 6 Requerimiento - RF06, Fuente Autor _____	43
Tabla 7 Requerimiento - RF07, Fuente Autor _____	44
Tabla 8 Requerimiento - RF08, Fuente Autor _____	44
Tabla 9 Requerimiento - RF09, Fuente Autor _____	45
Tabla 10 Requerimiento - RF10, Fuente Autor _____	45
Tabla 11 Requerimiento - RF11, Fuente Autor _____	46
Tabla 12 Requerimiento - RF12, Fuente Autor _____	46
Tabla 13 Requerimiento - RF13, Fuente Autor _____	47
Tabla 14 Requerimiento -RNF01, Fuente Autor _____	47
Tabla 15 Requerimiento -RNF02, Fuente Autor _____	48
Tabla 16 Requerimiento -RNF03, Fuente Autor _____	48
Tabla 17 Modelos de Raspberry pi, Tomado de [33] _____	74
Tabla 18 fases de OpenALPR, Tomado de [34] _____	80
Tabla 19 Especificación de requisitos de software -Definiciones, acrónimos y abreviaturas, Fuente Autor _____	100

Tabla 20 Especificación de requisito de software- Referencias, Fuente Autor	100
Tabla 21 Usuario- Administrador, Fuente Autor	101
Tabla 22 Usuario-Vigilante, Fuente Autor	101
Tabla 23 Iniciar sesión, Fuente Autor	103
Tabla 24 Gestionar Usuarios, Fuente Autor	105
Tabla 25 Registrar Usuario, Fuente Autor	107
Tabla 26 Gestionar Cámaras, Fuente Autor	109
Tabla 27 Registrar Cámaras, Fuente Autor	110
Tabla 28 Detecciones en línea, Fuente Autor	110
Tabla 29 Historial de las Detecciones, Fuente Autor	111
Tabla 30 Prueba- Detecciones en línea, Fuente Autor	130
Tabla 31 Prueba-Historial detecciones, Fuente Autor	131
Tabla 32 Prueba-Gestionar empleados, Fuente Autor	133
Tabla 33 Prueba- Gestionar empleados, Fuente Autor	135
Tabla 34 Prueba- Autenticación de usuario, Fuente Autor	136
Tabla 35 Prueba- Gestionar usuario, Fuente Autor	137
Tabla 36 Prueba- Registrar usuario, Fuente Autor	138
Tabla 37 Prueba - Consultar usuario, Fuente Autor	139
Tabla 38 Prueba- Modificar usuario, Fuente Autor	140
Tabla 39 Prueba- Gestionar cámara, Fuente Autor	141
Tabla 40 Prueba- Consultar cámara, Fuente Autor	142
Tabla 41 Prueba- Modificar cámara, Fuente Autor	143
Tabla 42 Prueba - Registrar cámara, Fuente Autor	144

Tabla 43 Prueba - Consultar informe detecciones, Fuente Autor _____	145
Tabla 44 Prueba- Registrar entradas, Fuente Autor _____	146
Tabla 45 Prueba- Captura automática de la placa, Fuente Autor _____	147

LISTA DE FIGURAS

	Pág.
Figura 1 Raspberry-pi-3-model-b, Tomado de [9] _____	26
Figura 2 Pasos del algoritmo ANPR, Tomado de [15]. _____	30
Figura 3 Pasos algoritmo ANPR - reconocimiento de patrones, Tomado de [15]	31
Figura 4 Arquitectura de Tesseract, tomado de [21] _____	34
Figura 5 Modelo de Prototipos, Tomado de [31] _____	38
Figura 6 Iniciar sesión, Fuente Autor _____	52
Figura 7 Gestionar Usuarios, Fuente Autor _____	53
Figura 8 Gestionar Cámaras, Fuente Autor _____	53
Figura 9 Detecciones en línea, Fuente Autor _____	54
Figura 10 Historial de las Detecciones, Fuente Autor _____	55
Figura 11 Diagrama de flujo, Fuente Autor _____	56
Figura 12 Diagrama Entidad – Relación, Fuente Autor _____	57
Figura 13 Modelo Relacional, Fuente Autor _____	58
Figura 14 Casos de uso, Fuente Autor _____	59
Figura 15 Diagrama de secuencia- Iniciar sesión, Fuente Autor _____	60
Figura 16 Diagrama de secuencia-Registra Usuario, Fuente Autor _____	60
Figura 17 Diagrama de secuencia -Consultar y actualizar usuario, Fuente Autor	61
Figura 18 Diagrama de secuencia - Registrar cámara, Fuente Autor _____	61
Figura 19 Diagrama de secuencia -Consultar y actualizar cámara, Fuente Autor	62
Figura 20 Diagrama de secuencia -Registrar entrada, Fuente Autor _____	62

Figura 21 Diagrama de secuencia-Consultar detecciones, Fuente Autor _____	63
Figura 22 Arquitectura de red, Fuente Autor _____	64
Figura 23 Interfaz de inicio de sesión, Fuente Autor _____	65
Figura 24 Interfaz de Bienvenida, Fuente Autor _____	66
Figura 25 Interfaz de Detecciones en línea, Fuente Autor _____	67
Figura 26 Interfaz de Historial de Detecciones, Fuente Autor _____	68
Figura 27 Imagen de foto guardada en base de datos, Fuente Autor _____	69
Figura 28 Interfaz de listado de empleados, Fuente Autor _____	69
Figura 29 Interfaz Registrar empleados, Fuente Autor _____	70
Figura 30 Interfaz Actualizar empleados, Fuente Autor _____	71
Figura 31 Interfaz de Cámaras, Fuente Autor _____	71
Figura 32 Interfaz Actualizar Cámara, Fuente Autor _____	72
Figura 33 Interfaz Registrar Cámara, Fuente Autor _____	72
Figura 34 Raspberry PI 3 implementada, Fuente Autor _____	73
Figura 35 Ejecución de etcher, Fuente Autor _____	75
Figura 36 Instalar SO Raspbian a SD, Fuente Autor _____	75
Figura 37 Instalar SO Raspbian, Fuente Autor _____	76
Figura 38 SO Raspbian, Fuente Autor _____	76
Figura 39 cámara IP, Fuente Autor _____	82
Figura 40 Prueba-Detección en línea, Fuente Autor _____	83
Figura 41 Prueba-Historial de detecciones, Fuente Autor _____	84
Figura 42 Prueba- Gestionar empleados, Fuente Autor _____	84
Figura 43 Prueba- Gestionar cámaras, Fuente Autor _____	85

Figura 44 Prueba- Autenticación de usuario, Fuente Autor _____	85
Figura 45 Prueba-Gestionar usuario, Fuente Autor _____	86
Figura 46 Prueba-Registrar usuario, Fuente Autor _____	86
Figura 47 Prueba-Consultar usuario, Fuente Autor _____	87
Figura 48 Prueba-Modificar usuario, Fuente Autor _____	87
Figura 49 Prueba-Gestionar cámara, Fuente Autor _____	88
Figura 50 Prueba-Consultar cámara, Fuente Autor _____	88
Figura 51 Prueba-Modificar cámara, Fuente Autor _____	89
Figura 52 Prueba-Registrar cámara, Fuente Autor _____	89
Figura 53 Prueba-Consultar informe detecciones, Fuente Autor _____	90
Figura 54 Prueba-Registrar entradas, Fuente Autor _____	90
Figura 55 Prueba-Captura automática de la placa, Fuente Autor _____	91
Figura 56 Prueba de Reconocimiento, Fuente Autor _____	92

LISTA DE ANEXOS

	Pág.
ANEXO A. Especificación de requisitos de software _____	100
ANEXO B . Descripción de los campos de la aplicación _____	103
ANEXO C . Desarrollo página web _____	113
ANEXO D. Creación de las tablas en la base de datos _____	124
ANEXO E. Características de la cámara _____	127
ANEXO F. Pruebas a la aplicación web _____	128

RESUMEN

En la universidad del Sinú sede Santillana de la ciudad de Cartagena diariamente se presentan ingresos de vehículos, ya sea de estudiantes, administrativos o visitantes. La verificación de estos automotores se realiza normalmente por los vigilantes asignados, quienes realizan un registro de la placa del automóvil de forma manual, en las bitácoras de vigilancia.

Por estos motivos, se decidió desarrollar el prototipo para identificación de placas vehiculares y reconocimiento de caracteres en tiempo real, implementado en un dispositivo RASPBERRY PI 3. Este dispositivo se usa como controlador del programa, el cual permitirá realizar un video en tiempo real desde una cámara integrada. Desde el video captado se extraerá una imagen de la placa del vehículo, la cual permitirá reconocer los caracteres que la integran. Posteriormente se agregará la información de placa, fecha, hora, código de la cámara en un servidor al cual se accede desde una aplicación web que visualizará la información agregada.

Para el cumplimiento de los objetivos planteados se utilizó la metodología del modelo de Prototipos. Adicionalmente, para el desarrollo de la aplicación se utilizaron diversas tecnologías como el lenguaje de programación PHP, el sistema manejador de base de datos MySQL y el servidor web Wamp Server 2.2, para el algoritmo de reconocimiento se utilizó la librería OpenALPR con el lenguaje de programación Python.

Palabras claves: Raspberry pi 3, modelo de Prototipos, PHP, MySQL, Wamp Server 2.2, OpenALPR, Python.

INTRODUCCIÓN

La visión artificial (VA) es un campo de la inteligencia artificial (IA) que trata de crear sistemas tecnológicos de reconocimiento óptico a través de métodos que permiten adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina. Este tipo de tecnología permite automatizar el proceso de obtención de información de las propiedades físicas de los objetos a partir del análisis de las imágenes captadas por las cámaras.

Una aplicación de VA son los sistemas de reconocimiento automático de placas vehiculares (ANPR), estas se componen por hardware y software permitiendo la lectura de la placa del vehículo y el reconocimiento de los caracteres alfanuméricos por medio de técnicas de reconocimiento óptico de caracteres (OCR).

Estos sistemas tienen muchas aplicaciones, como lo es en peajes para verificar si los vehículos o las personas tienen requerimientos por parte de los organismos judiciales; en los conjuntos cerrados, para controlar el ingreso de los vehículos; en los parqueaderos, para saber la hora de entrada y salida de los automotores.

Este trabajo presenta el desarrollo de un prototipo para la identificación de placas vehiculares y reconocimiento de los caracteres que esta posee, mediante el dispositivo electrónico RASPBERRY Pi 3. Sus principales funcionalidades son; identificación automática de la placa, reconocimiento de caracteres por medio de OCR, generación de texto de la placa y envío de forma automática de la información al servidor. El usuario interactúa con el sistema por medio de una aplicación web que se visualiza la información registrada en base de datos, como lo es la fecha, hora, foto y placa del vehículo.

1 DISEÑO METODOLÓGICO

1.1 PLANTEAMIENTO DEL PROBLEMA

1.1.1 Descripción del problema

La seguridad se ha convertido en un factor de atención prioritaria para la sociedad colombiana debido a que en el país, en los últimos años, las estadísticas de delitos que afectan la seguridad ciudadana, en aspectos como hurto, lesiones personales y delitos sexuales han tenido un aumento del 12, 124 y 21.7%, respectivamente [1]. Aun cuando estos delitos afectan a la ciudadanía en general, es importante para las instituciones de educación superior poder contar con mecanismos que disminuyan el riesgo de materialización de estos delitos en su comunidad estudiantil, docente y administrativa.

Los riesgos de materialización de delitos que afectan a la ciudadanía, aumentan cuando se tienen deficiencias en los sistemas de seguridad de puntos críticos [2]. El acceso vehicular a las instalaciones de establecimientos públicos y privados es un punto crítico de seguridad y debe ser gestionado para su control [3].

Para el caso de la sede Santillana de la Universidad del Sinú, de acuerdo a información obtenida por entrevista al personal de seguridad, cuenta con capacidad para albergar 72 vehículos en sus espacios de estacionamiento en esta diariamente acceden 300 vehículos en promedio. El control de los accesos se realiza de manera manual a través de una bitácora. Esto evidencia deficiencias en el sistema de seguridad para control de accesos.

1.1.2 Justificación

La seguridad es un factor importante para la comunidad en general y de atención prioritaria. Esto se debe a los altos índices de inseguridad ciudadana que ha presentado el país durante los últimos años. Para garantizar la seguridad, se necesitan mecanismos que disminuyan el riesgo de materialización de delitos. En este sentido, el control de acceso vehicular es un factor relevante que deben cubrir las instituciones. En la actualidad, en la sede Santillana de la Universidad Del Sinú, se pueden evidenciar fácilmente algunas debilidades en los procedimientos de control de acceso, posiblemente por la inexistencia de sistemas que permitan automatizar el proceso, este se viene haciendo de forma manual a través de bitácoras físicas. Por lo anterior, este proyecto presenta el desarrollo de un prototipo para identificación automática de placas, que aporta en el mejoramiento del proceso de control de acceso vehicular a la sede Santillana.

Al controlar el acceso vehicular en la sede Santillana, la comunidad académica podrá gozar de mejores niveles de seguridad y este es un aspecto importante, ya que, influye directamente en el bienestar universitario. Este aporte es, en sí mismo, el beneficio que traería el desarrollo del proyecto para la sociedad académica de la Universidad del Sinú.

Adicionalmente el desarrollo del proyecto permitirá afianzar los conocimientos y competencias en la automatización de procesos, en el cual se incluye la identificación automática de las placas de los vehículos, reconocer los caracteres por medio de OCR, generar texto de la placa y enviar de forma automática la información al servidor y que de este se pueda acceder, con este desarrollo, que puede ser implementado en muchos establecimiento, se busca dar una solución rentable a la universidad y que los estudiantes tengan acceso a la tecnología a implementar.

1.1.3 Formulación del problema

¿Cómo diseñar y desarrollar un prototipo que permita controlar el acceso vehicular a las instalaciones de la sede Santillana de la Universidad del Sinú, seccional Cartagena, usando un dispositivo electrónico Raspberry pi 3?

1.1.4 Objetivos

Desarrollar un prototipo de identificación automática, reconocimiento y visualización de placas vehiculares, mediante la programación de un dispositivo RASPBERRY PI 3 y una aplicación web, para la Universidad del Sinú seccional Cartagena sede Santillana.

Para cumplir con este propósito se proponen los siguientes objetivos específicos

- Analizar y diseñar los módulos que integrará la aplicación Web y la estructura de red de los componentes de hardware que se implementarán, para obtener una representación del producto de software.
- Desarrollar una aplicación Web Responsive para la visualización de la información almacenada en el servidor.
- Implementar un algoritmo para la identificación de las placas vehiculares y reconocimiento de los caracteres que la integran en el dispositivo RASPBERRY PI 3.
- Realizar validación del prototipo a través de un proceso de pruebas que permita demostrar la correcta funcionalidad.

1.2 ESTADO DEL ARTE

Existen distintas investigaciones y proyectos relacionados con el procesamiento y análisis de imágenes aplicados al reconocimiento de matrículas vehiculares:

- Desarrollo de un sistema de visión artificial para el reconocimiento de placas en vehículos particulares: estudiantes de la universidad de San Buenaventura de Bogotá en el año 2008 desarrollaron un sistema de visión artificial para el reconocimiento automático de placas en vehículos particulares. El sistema permitió identificar los seis caracteres de la placa del vehículo a través del procesamiento de la imagen digital, y a su vez registra los datos del reconocimiento en un archivo de texto editable que contiene información adicional como hora y fecha de la identificación.

La adquisición de las imágenes se realizó a través de una cámara web. Se emplearon diversas técnicas de procesamiento de imágenes como filtros de color operaciones morfológicas, crecimiento de regiones entre otras con el fin de conseguir la localización de la placa y la segmentación de los caracteres de la misma. Posteriormente para el reconocimiento de los caracteres se implementaron redes neuronales artificiales perceptor multicapa, que facilitaron la identificación de los mismos [4].

- Sistema de visión artificial para la detección y lectura de matrículas: el trabajo se realizó en el departamento de Ingeniería de Sistemas y Automática de la Escuela de Ingenierías Industriales de la Universidad de Valladolid. Surge de la idea de desarrollar un sistema de detección y reconocimiento de placas de matrículas en automóviles. El objetivo principal es que pueda ser empleado en el control de acceso a parkings.

Se trabajó sobre coches que aparecen en una imagen tomada por una cámara al detectar el vehículo, es decir, en tiempo real.

Constará de tres bloques: localización de la placa de la matrícula, tratamiento de la misma y reconocimiento de los caracteres. Dicho algoritmo se creó para obtener resultados satisfactorios independientemente del tamaño y rotación que pueda tener la placa.

Para el desarrollo se usó la librería de código abierto OpenCV basada en el lenguaje de programación C++ [5].

- 3LPR es un sistema de reconocimiento de placas vehiculares (sistema LPR) utilizado para identificar y registrar los vehículos que acceden o salen de un parqueadero, consiguiendo un gran control de placas de vehículos, y por tanto mayor seguridad.

El sistema de reconocimiento automático de matrículas no sólo está enfocado a los parkings, sino que puede ser utilizado en todas aquellas instalaciones que necesiten controlar, vigilar y tener un registro de todos los vehículos que traspasan un determinado acceso. Ejemplo son los garajes privados de empresas, centros comerciales, peajes, hospitales, etc. [6].

1.3 MARCOS DE REFERENCIA

1.3.1 Marco teórico

1.3.1.1 Raspberry Pi 3

Raspberry Pi (RP) es un ordenador de bajo costo integrado en una placa del tamaño de una tarjeta de crédito que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal. Más formalmente, podemos definirla como una plataforma abierta que permite diferentes sistemas operativos (SO) y distribuciones ligeras para desarrollo. Las distribuciones más comunes son Linux, y la más utilizada Raspbian [7].

Desarrollado en Reino Unido por la Fundación Raspberry Pi, con el objetivo de estimular la enseñanza de ciencias de la computación en las escuelas, aunque no empezó su comercialización hasta el año 2012 [8].

Figura 1 Raspberry-pi-3-model-b, Tomado de [9]



El Raspberry Pi 3 es de la tercera generación. Reemplazó el modelo B de Raspberry Pi 2 en febrero de 2016 [9].

Características:

- CPU Quad Core 1.2GHz Broadcom BCM2837 de 64 bits
- 1GB de RAM
- BCM43438 LAN inalámbrica y Bluetooth de baja energía (BLE) a bordo
- GPIO extendido de 40 pines
- 4 puertos USB 2
- Salida estéreo de 4 polos y puerto de video compuesto
- HDMI de tamaño completo
- Puerto de cámara CSI para conectar una cámara Raspberry Pi
- Puerto de pantalla DSI para conectar una pantalla táctil Raspberry Pi
- Puerto Micro SD para cargar su sistema operativo y almacenar datos
- Fuente de alimentación micro USB conmutada actualizada de hasta 2.5 A

[9]

1.3.1.2 Python 3.6

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas [10].

Características del lenguaje:

Propósito general

Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas [11]

Multiplataforma

Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él [11]

Interpretado

Quiere decir que no se debe compilar el código antes de su ejecución. En realidad sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código [11]

Interactivo

Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente [11]

Orientado a Objetos

La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables [11].

Funciones y librerías

Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip [11].

Sintaxis clara

Por último, destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas

mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar [11].

1.3.1.3 Reconocimiento automático de matrículas (ANPR)

El reconocimiento automático de matrículas (ANPR) se inventó en 1976 en el Police Scientific Developm. Sucursal en el REINO UNIDO. Sin embargo, ganó mucha popularidad durante la última década junto con la mejora de la cámara y el aumento en el procesamiento [13].

ANPR es un método de vigilancia masiva que utiliza el reconocimiento óptico de caracteres en las imágenes para leer las matrículas de los vehículos siendo un conjunto especial de componentes de hardware y software, que procesan una señal de entrada que contiene una representación gráfica, tanto imágenes estáticas como secuencias de video y reconocen los caracteres de la placa. Transformando la información de esa imagen en un texto que el ordenador sea capaz de leer [14].

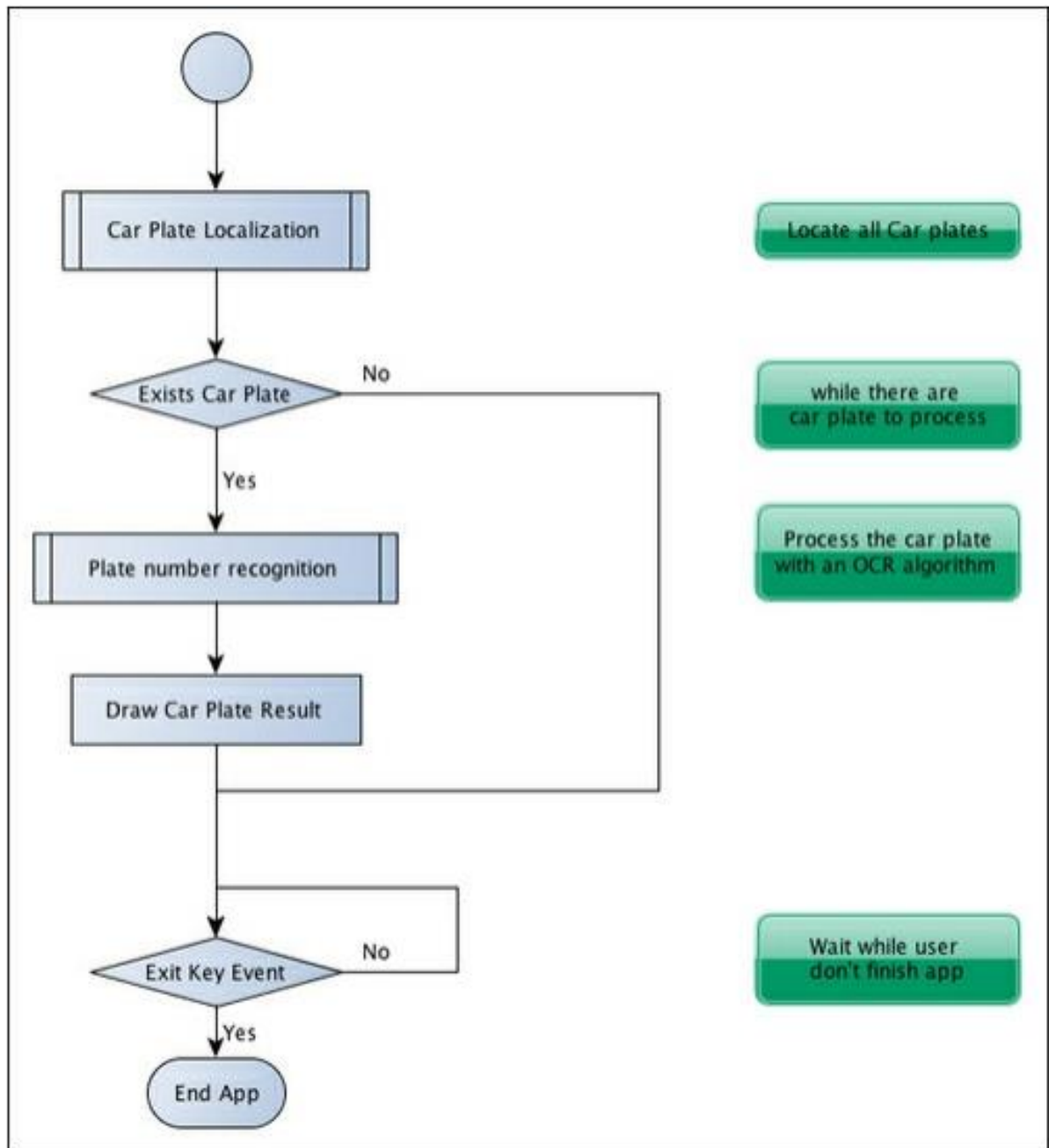
Algoritmo ANPR

ANPR se divide en dos pasos principales: detección de placa y reconocimiento de placa. La detección de placa tiene propósito de detectar la ubicación de la placa en todo el marco de la cámara. Cuando se detecta una placa en una imagen, el segmento de placa se pasa al segundo reconocimiento que usa un algoritmo de OCR para determinar los caracteres alfanuméricos en la placa [15].

En la siguiente figura podemos ver los dos pasos principales del algoritmo, detección de placa y reconocimiento de placa. Después estos pasos el programa dibuja sobre el marco de la cámara los caracteres de la placa que se han detectado.

Los algoritmos pueden devolver malos resultados o incluso ningún resultado:

Figura 2 Pasos del algoritmo ANPR, Tomado de [15].



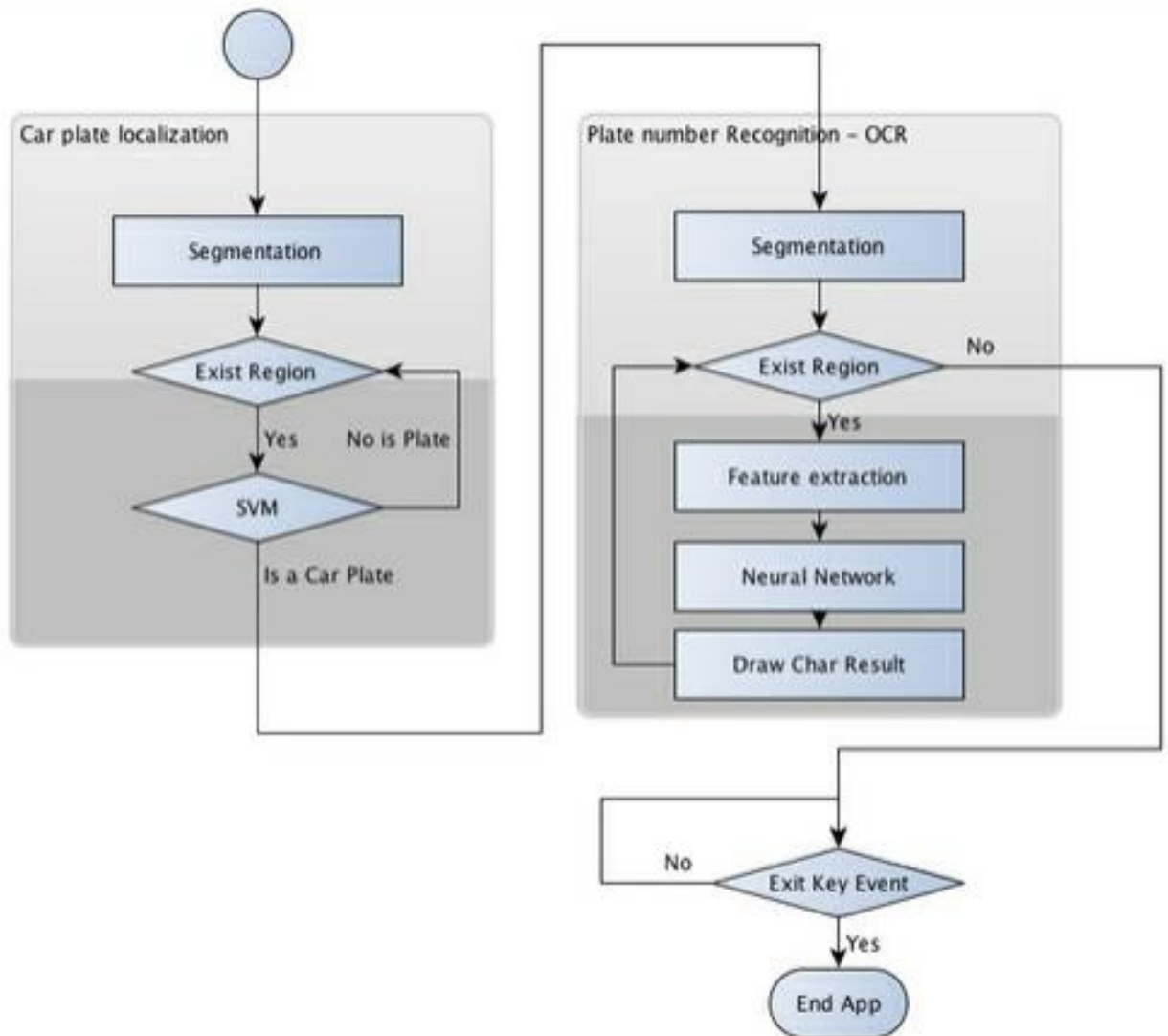
En cada paso que se muestra en la figura anterior, definiremos tres pasos adicionales que se utilizan comúnmente en algoritmos de reconocimiento de patrones:

- Segmentación: este paso detecta y elimina cada parche / región de interés en la imagen.
- Extracción de características: este paso extrae de un parche un conjunto de características.
- Clasificación: este paso extrae cada personaje del paso de reconocimiento de la placa o clasifica cada parche de imagen en "placa" o "sin placa" en el paso de detección de placa.

[15]

La siguiente figura nos muestra los pasos de reconocimiento de patrones en la aplicación de algoritmo completa:

Figura 3 Pasos algoritmo ANPR - reconocimiento de patrones, Tomado de [15]



1.3.1.4 OpenALPR

OpenALPR es una biblioteca de reconocimiento de matrículas automáticas de código abierto escrita en C ++ con enlaces en C #, Java, Node.js y Python. La biblioteca analiza imágenes y secuencias de video para identificar placas de matrícula. El resultado es la representación en texto de los caracteres de la matrícula [\[16\]](#).

OpenALPR se apoya en otras librerías para su funcionamiento, siendo estas:

- **OpenCV**: este un marco para el procesamiento de imágenes; modificación y aplicación de algoritmos en imágenes.
- **Tesseract OCR**: es un software de código abierto para el reconocimiento de caracteres que utiliza algoritmos de aprendizaje automático.

Los pasos para el procesamiento se inician en OpenCV que aplica algoritmos para detectar la matrícula, luego aplicará una segmentación para obtener caracteres en la matrícula, y luego los enviará a Tesseract para leerlos y reconocerlos (es decir, convertirlos a texto) [\[17\]](#).

Con OpenALPR, se puede:

- Reconocer placas de las transmisiones de la cámara. Los resultados son navegables, se pueden buscar y pueden activar alertas. El repositorio de datos puede estar en la nube o almacenado completamente dentro de su red en el sitio.
- Reconozca las placas de las transmisiones de la cámara y envíe los resultados a su propia aplicación.
- Procese un archivo de video y almacene los resultados de la matrícula en una base de datos CSV y SQLite.
- Analiza imágenes fijas desde la línea de comando.
- Integrar el reconocimiento de matrículas en su aplicación directamente en el código (C / C ++, C #, VB.NET, Java, Python, Node.js).

[\[16\]](#)

1.3.1.5 OpenCV (Open Source Computer Vision Library)

Se lanza bajo licencia BSD y, por lo tanto, es gratuito tanto para uso académico como comercial. Tiene interfaces C ++, Python y Java y es compatible con Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real [18].

OpenCV es una biblioteca de software de visión abierta y software de aprendizaje automático. Fue construido para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia de BSD, OpenCV facilita a las empresas utilizar y modificar el código [19].

La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de visión artificial y de aprendizaje automático tanto clásico como avanzado. Estos algoritmos se pueden usar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de la cámara y rastrear objetos en movimiento, etc. [19].

1.3.1.6 Tesseract OCR

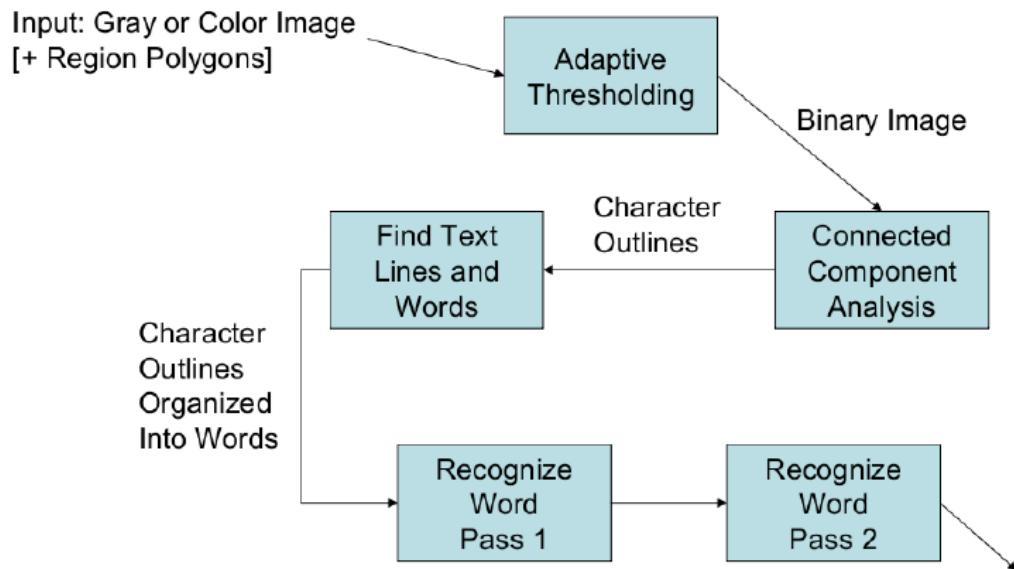
Tesseract es un motor OCR libre. Fue desarrollado originalmente por Hewlett Packard como software propietario entre 1985 y 1995. Tras diez años sin ningún desarrollo, fue liberado como código abierto en el año 2005 por Hewlett Packard y la Universidad de Nevada, Las Vegas. Tesseract es desarrollado actualmente por Google y distribuido bajo la licencia Apache, versión 2.0. [20].

Arquitectura

En la arquitectura de Tesseract, si la imagen de entrada es una imagen a color o en escala de grises se usa un filtro adaptativo para transformarla en una imagen binaria. Si la imagen es binaria en un principio, este paso se omite. Posteriormente, se realiza un análisis de los componentes en el cual se obtienen los contornos de los diferentes caracteres. Los contornos son anidados en

estructuras llamadas Blobs. Posteriormente estos Blobs organizados en líneas de texto (Text Lines), las cuales son divididas en palabras. Estas palabras son luego enviadas a reconocimiento y cuando son reconocidas se envían a un entrenamiento adaptativo (Adaptive Trainer, AT). Las nuevas lecciones aprendidas por el AT son usadas en una segunda pasada, buscando reconocer palabras no reconocidas. Al final el OCR busca resolver los espacios confusos y los textos en letra minúscula [21].

Figura 4 Arquitectura de Tesseract, tomado de [21]



Para cumplir estas etapas, Tesseract utiliza,

1. Algoritmos para la detección de líneas de texto.
2. Algoritmos para detectar palabras proporcionales y no-proporcionales. (En una palabra proporcional los caracteres tiene el mismo ancho)
3. Algoritmos que para separar caracteres unidos o asociar caracteres divididos.
4. Análisis lingüístico que busca identificar la palabra más parecida a un conjunto de caracteres.
5. Dos clasificadores de caracteres. Un clasificador estático y un clasificar adaptativo que es entrenado.

[21]

1.3.2 Marco conceptual

Reconocimiento óptico de caracteres (OCR): Es una tecnología que trata de emular la capacidad del ojo humano para reconocer objetos. Concretamente es un software que permite el reconocimiento óptico de los caracteres contenidos en una imagen (documento escaneado o fotografía), de forma que estos se vuelven comprensibles o reconocibles para un ordenador, obteniendo como resultado final un archivo en un formato de texto editable. El formato del archivo de salida (txt, pdf, etc.) dependerá de las posibilidades que ofrezca el software [22].

Blobs: La detección de blobs consiste en identificar y diferenciar cada objeto que se encuentra dentro de una imagen digital, siendo este objeto la región o la unión de regiones que serán extraídas de la imagen mediante métodos de segmentación [23].

Aprendizaje automático: El aprendizaje automático es un tipo de inteligencia artificial (AI) que proporciona a las computadoras la capacidad de aprender, sin ser programadas explícitamente. El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden cambiar cuando se exponen a nuevos datos [24]. Las máquinas aprenden por sí solas, son capaces de auto programarse aprendiendo de su propia experiencia combinando datos de entradas y situaciones del mundo real. Según Forbes, a través del uso de algoritmos pueden parsear datos, aprender de ellos y que máquina pueda predecir o sugerir algo. Desde sus inicios la IA ha creado algoritmos que han ido evolucionando para poder conseguir mejores resultados [25].

El aprendizaje automático emplea dos tipos de técnicas: el aprendizaje supervisado, que entrena un modelo con datos de entrada y salida conocidos para que pueda predecir salidas futuras, y el aprendizaje no supervisado, que encuentra patrones ocultos o estructuras intrínsecas en los datos de entrada [26].

Aprendizaje no supervisado: una técnica concreta dentro del Machine Learning, un subconjunto, el cual desarrolla la idea del aprendizaje profundo a través de modelos informáticos que funciona de forma similar al cerebro humano, un sistema de redes artificiales de neuronas que analiza los datos. La máquina evalúa ejemplos e instrucciones para modificar el modelo en el caso de que se produzcan errores. El sistema extrae patrones que facilita la solución de

problemas de una manera bastante precisa, en definitiva, toma decisiones a partir de datos [25].

Clasificador: Un clasificador es un método para determinar la posible clase de un objeto desconocido, o evento sobre la base de un número de casos de cada una de las clases, conocido como el conjunto de entrenamiento [27].

El primer paso en la clasificación es la extracción de características, que es donde cada instancia en el conjunto de entrenamiento se expresa como un vector de medidas. Cuando están siendo clasificadas las imágenes, este vector podría estar definido a partir de las intensidades de los píxeles, pero con frecuencia se aplica un paso de reducción de características. Las medidas se denominan habitualmente características, y pueden ser reales, enteros o categóricas. El espacio que abarca todas las combinaciones posibles de características se le denomina espacio de características. Para algunos problemas no todas las mediciones están disponibles y esto se conoce como datos faltantes [27].

Una vez las características se han extraído hay dos posibles casos, que en general se manejan de forma muy diferente. En el primer caso, la clase de cada instancia del conjunto de entrenamiento se pone a disposición del clasificador. En el segundo caso, la información no está disponible, y esto se llama clasificación sin supervisión o “clustering” [27].

Raspbian: es el sistema operativo recomendado para Raspberry Pi (al estar optimizado para su hardware) y se basa en una distribución de GNU/Linux llamada Debian [28].

1.4 DISEÑO METODOLÓGICO

1.4.1 Línea de investigación

La Universidad del Sinú seccional Cartagena cuenta con varios Grupos de Investigación que trabajan en una temática específica por medio de proyectos que permiten generar conocimientos. Las líneas de investigación de estos grupos se establecen de acuerdo a los temas que se aborden en los proyectos.

Para el prototipo de identificación y reconocimiento de placas vehiculares se escogió la línea de investigación de desarrollo de software, cuyo propósito es la implementación de un algoritmo, que permitan al dispositivo Raspberry PI realizar la tarea de reconocimiento y captura de placas, apoyado en la tecnología de visión artificial.

El semillero de investigación tendrá un impacto positivo debido a que se afianzarán los conocimientos en la disciplina de visión artificial. Esta permite la automatización del proceso de obtención de información de las propiedades físicas de objetos, a partir del análisis de imágenes captadas por cámara.

1.4.2 Tipo de investigación

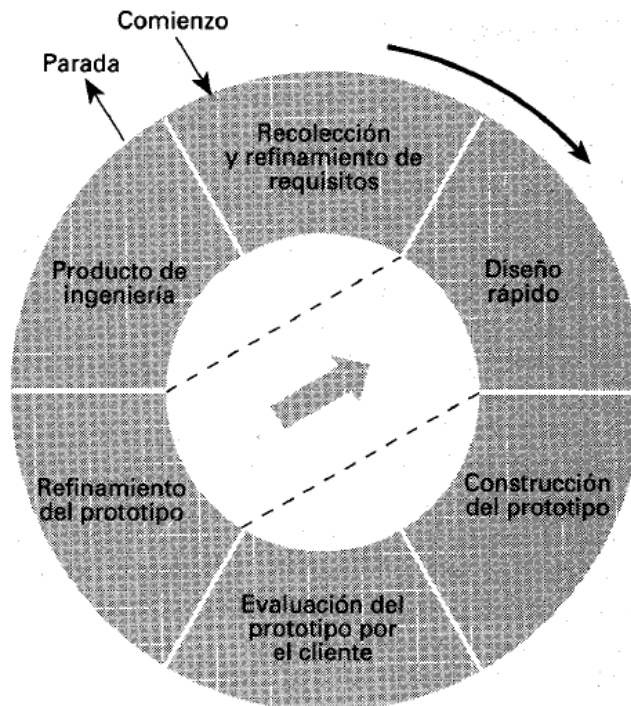
La investigación a utilizar es de tipo aplicada. En esta el problema está establecido y es conocido por el investigador. Por lo cual, utiliza la investigación para dar respuesta a preguntas específicas, En este tipo de investigación el énfasis del estudio está en la resolución práctica de problemas [29]. La característica más destacada de la investigación aplicada es su interés en la aplicación de los conocimientos que se han obtenido y en las consecuencias prácticas de este [29]. Por consiguiente, el investigar sobre el desarrollo de un software que automatiza el proceso de identificación de placas vehiculares, nos lleva a indagar sobre un tema no conocido para aplicarlo en la solución de la problemática planteada.

1.4.3 Definición de la metodología

Las metodologías destinadas al desarrollo de software se consideran como una estructura utilizada para planificar y controlar el proceso de creación de un sistema de información eficiente y de mayor calidad, con la documentación necesaria, en perfecto orden y en el tiempo requerido [30].

El Modelo de Prototipos. También conocido como desarrollo con prototipación o modelo de desarrollo evolutivo, se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Este modelo se utiliza para dar al usuario una vista preliminar de parte del software. Este modelo es básicamente prueba y error ya que si al usuario no le gusta una parte del prototipo significa que la prueba fallo por lo cual se debe corregir el error que se tenga hasta que el usuario quede satisfecho. Además el prototipo debe ser construido en poco tiempo, usando los programas adecuados y no se debe utilizar mucho dinero pues a partir de que este sea aprobado se puede iniciar el verdadero desarrollo del software [31].

Figura 5 Modelo de Prototipos, Tomado de [31]



Las fases que comprende el método de prototipos evolutivos y las actividades establecidas para cada una de estas, son:

Recolección y refinamiento de requisitos: Esta fase equivale a la de recolección de requisitos del ciclo de vida clásico e incluye además la planificación de las actividades a realizar en cada iteración [32].

En esta fase las actividades establecidas para alcanzar el objetivo en el que se analiza los módulos de la aplicación Web y la estructura de red implementada, se contempla, el levantamiento de los requisitos funcionales y no funcionales del sistema de información y la descripción de las características del dispositivo RASPBERRY PI 3 siendo este el componente principal de la arquitectura de red.

Modelado, diseño rápido: Se tratará de un diseño rápido, centrado sobre todo en la arquitectura del sistema y la definición de la estructura de las interfaces más que en aspectos procedimentales de los programas [32].

El objetivo de diseñar los módulos de la aplicación web y de la estructura de red se desarrolló en esta etapa con las siguientes actividades:

- Elaboración de los casos de usos
- Elaboración del diagrama Entidad – Relación de la base de datos
- Elaboración del modelo de la base de datos
- Elaboración del diagrama de flujo
- Diseño de los mockups
- Diseño de la arquitectura de red del prototipo de reconocimiento de placas

Construcción del Prototipo: A partir del diseño se construye el prototipo, el objetivo es que la codificación sea rápida [32].

Para esta etapa, el desarrollo de una página Web Responsive y el algoritmo para reconocimiento de las placas vehiculares son los objetivos que se establecen, en este se realizan las siguientes actividades:

- El desarrollo de la aplicación Web Responsive
- La implementación del algoritmo para la identificación y reconocimiento de placas vehiculares en el dispositivo Raspberry Pi 3
- Instalación del sistema operativo y Configuración del Raspberry PI 3

Evaluación del prototipo por el cliente: Una vez listo el prototipo, hay que presentarlo al cliente para que lo pruebe y sugiera modificaciones. En este punto el cliente puede ver una implementación de los requisitos que ha definido inicialmente y sugerir las modificaciones necesarias en las especificaciones para que satisfagan mejor sus necesidades [32].

Para la verificación de la funcionalidad del prototipo se determinó el objetivo en el cual se realizan las validaciones de este a través de esta etapa, con las siguientes actividades:

- Prueba del prototipo funcional

Refinamiento del prototipo: Ocurre la retroalimentación del usuario luego de usar el prototipo, es seguido por el diseño rápido o por producto de ingeniería.

2 ANÁLISIS DEL PROBLEMA

2.1 REQUISITOS ESPECÍFICOS

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autenticación de Usuario
Características:	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento:	Permite o deniega el ingreso al sistema de información de acuerdo a las credenciales de los usuarios, ver Anexo B para la descripción de los campos.
Requerimiento NO funcional:	RNF01 RNF03
Prioridad del requerimiento:	Alta

Tabla 1 Requerimiento - RF01, Fuente Autor

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Gestionar Usuario
Características:	Permite gestionar información referente a los usuarios del sistema.
Descripción del requerimiento:	En este requerimiento se realizarán operaciones sobre el módulo de los usuarios, permitiendo el ingreso, la consulta y la actualización de los registros en la base de datos. Ver Anexo B, en el se detallan las acciones para ejecutar cada proceso.
Requerimiento NO funcional:	RNF03
Prioridad del requerimiento:	Media

Tabla 2 Requerimiento - RF02, Fuente Autor

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Registrar Usuario
Características:	Registra la datos de los usuarios que tendrán acceso al modulo de consultas de entrada de los vehiculos al parqueadero.
Descripción del requerimiento:	Ingresa los datos de cedula, nombre, apellidos, telefono, direccion , email y clave de acceso del sistema de información, para conocer las características de cada campo ver Anexo B.
Requerimiento NO funcional:	RNF01 RNF02
Prioridad del requerimiento:	Media

Tabla 3 Requerimiento - RF03, Fuente Autor

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Consultar Usuario
Características:	Consulta los datos de los usuarios registrados en el sistemas.
Descripción del requerimiento:	El sistema permitira al administrador consultar los datos de los usuarios registrados, de acuerdo a los filtros realizados. En el Anexo B en la tabla de gestionar usuario tiene incluido este requerimiento con el campo mostrar registros.
Requerimiento NO funcional:	RNF01
Prioridad del requerimiento:	Baja

Tabla 4 Requerimiento - RF04, Fuente Autor

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Modificar Usuario
Características:	Realizará las modificaciones de acuerdo a solicitud del usuarios.
Descripción del requerimiento:	Permite la actualización de datos básicos de los usuarios registrados como lo es teléfono, dirección, email y clave de acceso al sistema de información. En el Anexo B en la tabla de gestionar usuario se incluye este requerimiento con el botón editar empleado.
Requerimiento NO funcional:	RNF01
Prioridad del requerimiento:	Baja

Tabla 5 Requerimiento - RF05, Fuente Autor

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Gestionar Camara
Características:	Permite gestionar la información de las cámaras.
Descripción del requerimiento:	Realizará operaciones sobre el módulo de las cámaras, como lo es, el registro, la consulta y actualización de los datos, para ver adonde realizar cada proceso ir al Anexo B.
Requerimiento NO funcional:	RNF03
Prioridad del requerimiento:	Media

Tabla 6 Requerimiento - RF06, Fuente Autor

Identificación del requerimiento: RF07

Nombre del Requerimiento: Consultar Camara

Características: Consulta las camaras registradas en el sistemas.

Descripción del requerimiento: El sistema permitira al administrador consultar los datos de las camaras registradas, de acuerdo a los filtros realizados. En el Anexo B en la tabla de gestionar camara se incluye este requerimiento en el campo de grilla Lista de cámaras.

Requerimiento NO funcional: RNF01

Prioridad del requerimiento: Baja

Tabla 7 Requerimiento - RF07, Fuente Autor

Identificación del requerimiento: RF08

Nombre del Requerimiento: Modificar Camara

Características: Realizará las modificaciones de los datos de la camara.

Descripción del requerimiento: Permite la actualizacion de los datos de la camara como lo es la ubicacion de esta. En el Anexo B en la tabla de gestionar camara se incluye este requerimiento con el boton de editar camara.

Requerimiento NO funcional: RNF01

Prioridad del requerimiento: Baja

Tabla 8 Requerimiento - RF08, Fuente Autor

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Registrar Camara
Características:	Registra la información de las camaras.
Descripción del requerimiento:	Ingresar los datos de ubicación y características de las camaras en la base de datos, ver Anexo B para conocer las condiciones de cada campo.
Requerimiento NO funcional:	RNF01 RNF02
Prioridad del requerimiento:	Media

Tabla 9 Requerimiento - RF09, Fuente Autor

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Historial de las detecciones
Características:	El sistema ofrecerá a los usuarios información general acerca de las entradas realizadas de los vehiculos.
Descripción del requerimiento:	Muestra información general sobre las entradas realizadas de acuerdo al filtro seleccionado, en este se detallaran fecha y hora de ingreso, placa y fotografia del vehiculo. Permite la generación de informes, para mayor detalle ver Anexo B.
Requerimiento NO funcional:	RNF01 RNF03
Prioridad del requerimiento:	Alta

Tabla 10 Requerimiento - RF10, Fuente Autor

Identificación del requerimiento:	RF11
Nombre del Requerimiento:	Detecciones en línea
Características:	El sistema ofrecerá a los usuarios información general de los ingresos realizados en el día siendo este actualizado cada 2 segundos.
Descripción del requerimiento:	Muestra información general sobre las entradas realizadas en el día, en este se detallaran fecha y hora de ingreso, placa y fotografía del vehículo. Permite la generación de informes, para mayor detalle ver Anexo B.
Requerimiento NO funcional:	RNF01
	RNF03
Prioridad del requerimiento:	Alta

Tabla 11 Requerimiento - RF11, Fuente Autor

Identificación del requerimiento:	RF12
Nombre del Requerimiento:	Registrar Entradas
Características:	Registra los datos de los ingresos al parqueadero de vehiculos.
Descripción del requerimiento:	Ingresa los datos de placa, fecha y hora de entrada en la base de datos, la fotografía del vehiculo se almacena en el servidor la cual es tomada automaticamente desde la Raspberry.
Requerimiento NO funcional:	RNF03
	RNF02
Prioridad del requerimiento:	Alta

Tabla 12 Requerimiento - RF12, Fuente Autor

Identificación del requerimiento:	RF13
Nombre del Requerimiento:	Captura automática de la Placa
Características:	Identifica y captura la placa automáticamente del vehículo
Descripción del requerimiento:	Identifica en tiempo real la placa de los vehículos que van a ingresar en el parqueadero, capturando una imagen de esta y reconociendo los caracteres que en esta se presentan. Este requerimiento no tiene interfaz que interactúe con el usuario.
Requerimiento NO funcional:	RNF03
Prioridad del requerimiento:	Alta

Tabla 13 Requerimiento - RF13, Fuente Autor

Requerimientos No Funcionales

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento:	Alta

Tabla 14 Requerimiento -RNF01, Fuente Autor

Identificación del requerimiento: RNF02

Nombre del Requerimiento: Seguridad en información

Características: Se encarga de proteger la integridad y la privacidad de la información almacenada en la base de datos.

Descripción del requerimiento: Garantizar la seguridad del sistema con respecto a la información y datos que se manejan, realizando copia de seguridad de dicha información.

Prioridad del requerimiento: Alta

Tabla 15 Requerimiento -RNF02, Fuente Autor

Identificación del requerimiento: RNF03

Nombre del Requerimiento: Eficiencia

Características: Permite realizar el proceso con el mínimo de esfuerzos y recursos del sistema

Descripción del requerimiento: El sistema debe ser capaz de tomar los datos de la placa del vehículo en tiempo real para realizar su respectivo procesamiento y actualizar los datos de la base de datos en segundos.

Prioridad del requerimiento: Alta

Tabla 16 Requerimiento -RNF03, Fuente Autor

2.2 REQUISITOS COMUNES DE LAS INTERFACES

Interfaces de usuario

La interfaz con el usuario será a través de una página web diseñada con la técnica responsive o adaptativa que busca la correcta visualización de una misma página en distintos dispositivos. Desde ordenadores de escritorio a tablets y móviles. Ésta deberá ser construida específicamente para el sistema propuesto.

Interfaces de hardware

Será necesario disponer de equipos de cómputos en perfecto estado con las siguientes características:

- Adaptador de red con acceso a internet
- Memoria mínima de 256Mb.
- Mouse.
- Teclado.
- Pantalla

Interfaces de software

- Explorador: Mozilla o Chrome.

Interfaces de comunicación

El servidores, los clientes y la aplicación se comunicarán entre sí, mediante protocolos estándares en internet, siempre que sea posible. Por ejemplo, para transferir archivos o documentos deberán utilizarse protocolos existentes (FTP u otros convenientes).

2.3 REQUISITOS FUNCIONALES

Requisito funcional 1

- **Autenticación De usuario:** Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
 - ✓ Permite o deniega el ingreso al sistema de información de acuerdo a las credenciales de los usuarios.

Requisito funcional 2

- **Gestionar usuario:** Permite gestionar información referente a los usuarios del sistema.
 - ✓ **Registrar usuario:** Registra la información de los usuarios que tendrán acceso al modulo de consultas de entrada de los vehiculos al parqueadero.
 - ✓ **Consultar usuario:** Consulta los datos de los usuarios registrados en el sistemas.
 - ✓ **Modificar usuario:** Realizará las modificaciones de acuerdo a solicitud del usuarios.

Requisito funcional 3

- **Gestionar Camara:** Permite gestionar la información de las camaras.
 - ✓ **Registrar Camara:** Registra la información de las camaras.
 - ✓ **Consultar Camara:** Consulta las camaras registradas en el sistemas.
 - ✓ **Modificar Camara:** Realizará las modificaciones de los datos de la camara.

Requisito funcional 4

- **Historial de las detecciones:** El sistema ofrecerá a los usuarios información general acerca de las entradas realizadas, permite realizar filtros de acuerdo a la solicitud del usuario y exporta la misma.
- **Detecciones en línea:** La aplicación Web será actualizada cada 2 segundos, en esta se visualizará la información únicamente de los vehículos ingresados en el día, permite realizar filtros y exportar los datos de la misma.

Requisito funcional 5

- **Registrar Entradas:** Registra los datos de fecha, hora y placa de las entradas de los vehículos al parqueadero.

Requisito funcional 6

- **Captura automática de la Placa:** Identifica en tiempo real la placa de los vehículos que van a ingresar en el parqueadero, capturando una imagen de esta y reconociendo los caracteres que la integran.

2.4 REQUISITOS NO FUNCIONALES

➤ Eficiencia

- ✓ El sistema debe ser capaz de tomar los datos de la placa del vehículo en tiempo real para realizar su respectivo procesamiento.
- ✓ Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

➤ Seguridad de los datos

El sistema debe ser capaz de realizar copia de seguridad de las bases de datos, por si algo falla en el servidor tener un backup de dicha información.

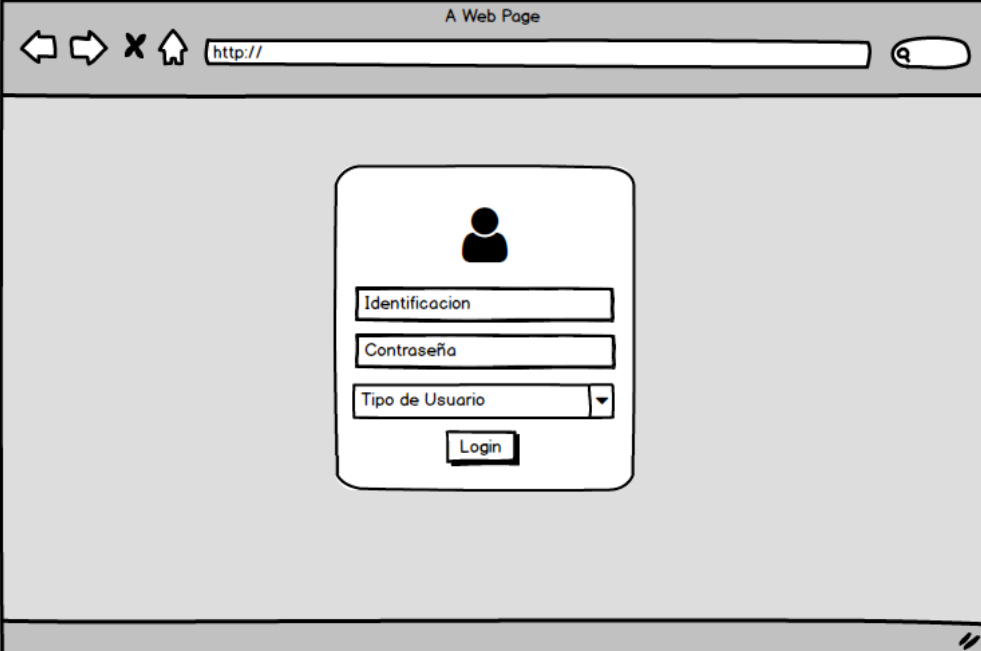
3 DISEÑO DE LA SOLUCIÓN

3.1 DISEÑO DE LA APLICACIÓN WEB

Las interfaces de la aplicación web constan de 3 módulos, cada uno de ellos tiene diferentes funciones, los cuales se especificarán a continuación:

3.1.1 Iniciar Sesión

Figura 6 Iniciar sesión, Fuente Autor



El diagrama muestra una ventana de navegador web con el título "A Web Page". La barra de direcciones contiene "http://". El contenido principal de la página es un formulario de inicio de sesión centrado, que incluye un ícono de usuario, tres campos de entrada: "Identificación", "Contraseña" y "Tipo de Usuario" (con un menú desplegable), y un botón "Login".

Esta interfaz es la principal de la aplicación web, aquí se validará el rol que tiene el usuario que intenta acceder al perfil designado; para el ingreso el usuario debe ingresar su número de identificación y contraseña para más detalle ver anexo B.

3.1.2 Gestionar Usuarios

Figura 7 Gestionar Usuarios, Fuente Autor

Nombre de aplicacion

Home/ Listado de empleados

Agregar Empleado

Mostrar 10 registros

filtrar

Identificacion	Nombres	Primer apellido	Segund apellido	Direccion	E-Mail	Telefono	Rol	Edicion
1047XXXXXX	MOISES	ARNEDO	MENDOZA	PASEO BOLIVAR xxxxx	MOISESXX@XXX.COM	304XXXXXX	Administrador	Editar

En la pestaña, listado de empleados se gestiona el registro, la consulta y la actualización de la información de los usuarios del sistema los cuales son empleados, esta interfaz es visualizada solo por el administrador del sistema, véase detalle en anexo B.

3.1.3 Gestionar Cámaras

Figura 8 Gestionar Cámaras, Fuente Autor

Nombre de aplicacion

Home/ Gestionar Camaras

Agregar Camara IP

Mostrar 10 registros

filtrar

Id	Lugar	Caracteristicas	Opciones
Cam_EntPpal	Entrada Principal	Direccion IP: 192.XXX.XXX.XXX	Boton Editor

En la pestaña gestionar cámara se permitirá el ingreso de los datos de estas, se realizarán las consultas y actualizaciones de las mismas. Con el fin de conectar diferentes cámaras al sistema de información, véase detalle en anexo B.

3.1.4 Detecciones en línea

Figura 9 Detecciones en línea, Fuente Autor

Nombre de la aplicación

Reportes

- >Detecciones en línea
- >Historial de detecciones

Empleados

- >Listado de empleados

Cameras

- >Gestionar Camaras

Home/ Detecciones en línea

Filtrar

id	Camara	Fecha y Hora	Placa	Fotografia
1	Cam_Ppal_IP	ABC123	Verfoto	
2	Cam_Ppal_IP	ABC123	Verfoto	

Esta interfaz permite visualizar los vehículos que han ingresado en el día, actualizando la información de la grilla cada 2 segundos, en esta se muestra los datos de la placa, fecha y hora de ingreso y proporciona una imagen del vehículo, véase detalle en anexo B.

Permite generar reportes los cuales se pueden copiar o imprimir, de igual manera exportar en PDF Y EXCEL.

3.1.5 Historial de las Detecciones

Figura 10 Historial de las Detecciones, Fuente Autor

Id	Camara	Fecha y Hora	Placa	Fotografia
1	Cam_Ppal_IP	01-Abr-2018 10:45 AM	ABC123	VerFoto
2	Cam_Ppal_IP	01-Abr-2018 10:59 AM	ABC456	VerFoto

Muestra una grilla con el historial de los ingresos realizados visualizando la información de placa, fecha y hora de ingreso y proporciona una imagen del vehículo, véase detalle en anexo B.

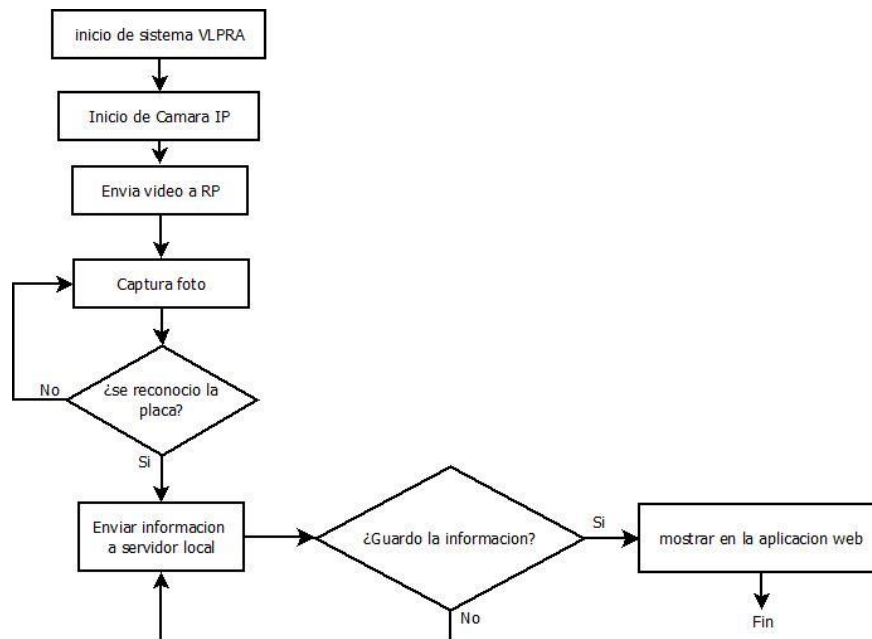
Permite generar reportes los cuales se pueden copiar o imprimir, de igual manera exportarlos en PDF Y EXCEL.

3.2 DIAGRAMAS

3.2.1 Diagrama de flujo del prototipo

En el diagrama se representan las acciones que se ejecutarán en el prototipo de identificación y reconocimiento de placas.

Figura 11 Diagrama de flujo, Fuente Autor

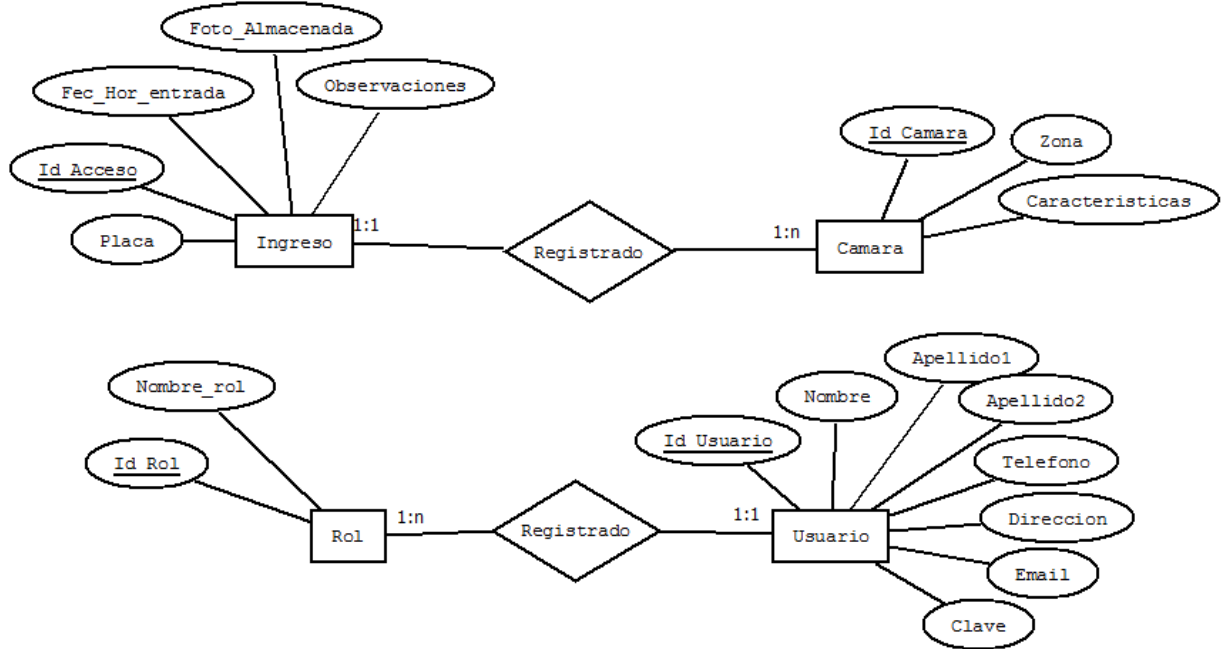


La cámara IP estará encendida para el envío del video a la Raspberry Pi, esta captura una foto del vehículo y realiza el reconocimiento de caracteres (OCR), si este procedimiento no detecta carácter se toma otra fotografía, si vuelve a fallar, el vigilante deberá solicitar al administrador registrar en la base de datos el ingreso del vehículo. El vigilante para determinar que no hubo reconocimiento verificará en la aplicación Web en el módulo de detecciones en línea si se encuentra la placa del vehículo.

Si el reconocimiento es exitoso se envía a la base de datos la información de placa, fecha y hora de ingreso, al igual que la fotografía del vehículo se enviará al servidor, si ocurre algún error en el envío se debe verificar que las conexiones a la red sean correctas ya que el envío se realiza en este medio. Todo vehículo que ingresa debe de estar en el módulo de detecciones en línea.

3.2.2 Diagrama Entidad – Relación

Figura 12 Diagrama Entidad – Relación, Fuente Autor



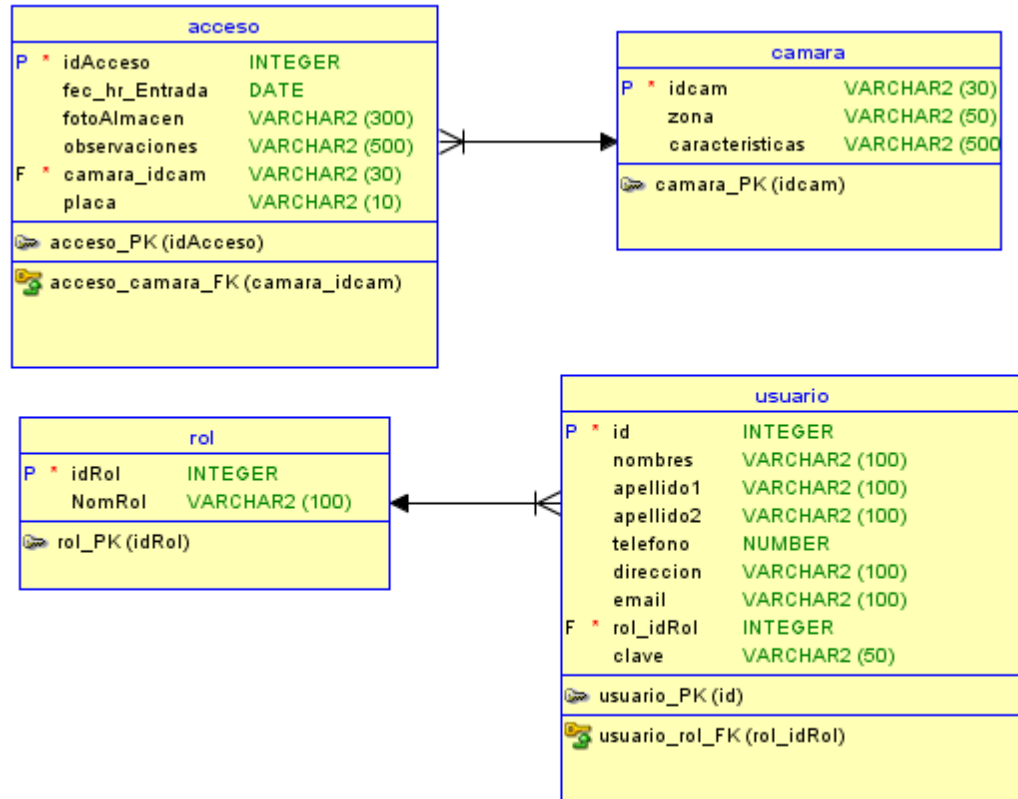
El Diagrama de entidad-relación, es una técnica de modelización que muestra los datos del sistema, así como las relaciones entre estos. Para el prototipo de reconocimiento de placas vehiculares se diseñaron cuatro entidades con sus atributos, estas se dividieron en dos partes:

- La primera está compuesta para el registro de las entradas de los vehículos relacionando la cámara que realiza el proceso de captura.
- En la segunda parte se relacionan los usuarios del sistema con sus respectivos roles, estos se dividen de acuerdo a la función que realizan.

La división se realiza al establecer que las entidades de ingresos y cámara no tendrán usuario asociado al ser un proceso automatizado sin la intervención de una persona. La entidad de los usuarios se limita al ingreso de los datos de los empleados, usuarios del prototipo.

3.2.3 Modelo Relacional

Figura 13 Modelo Relacional, Fuente Autor



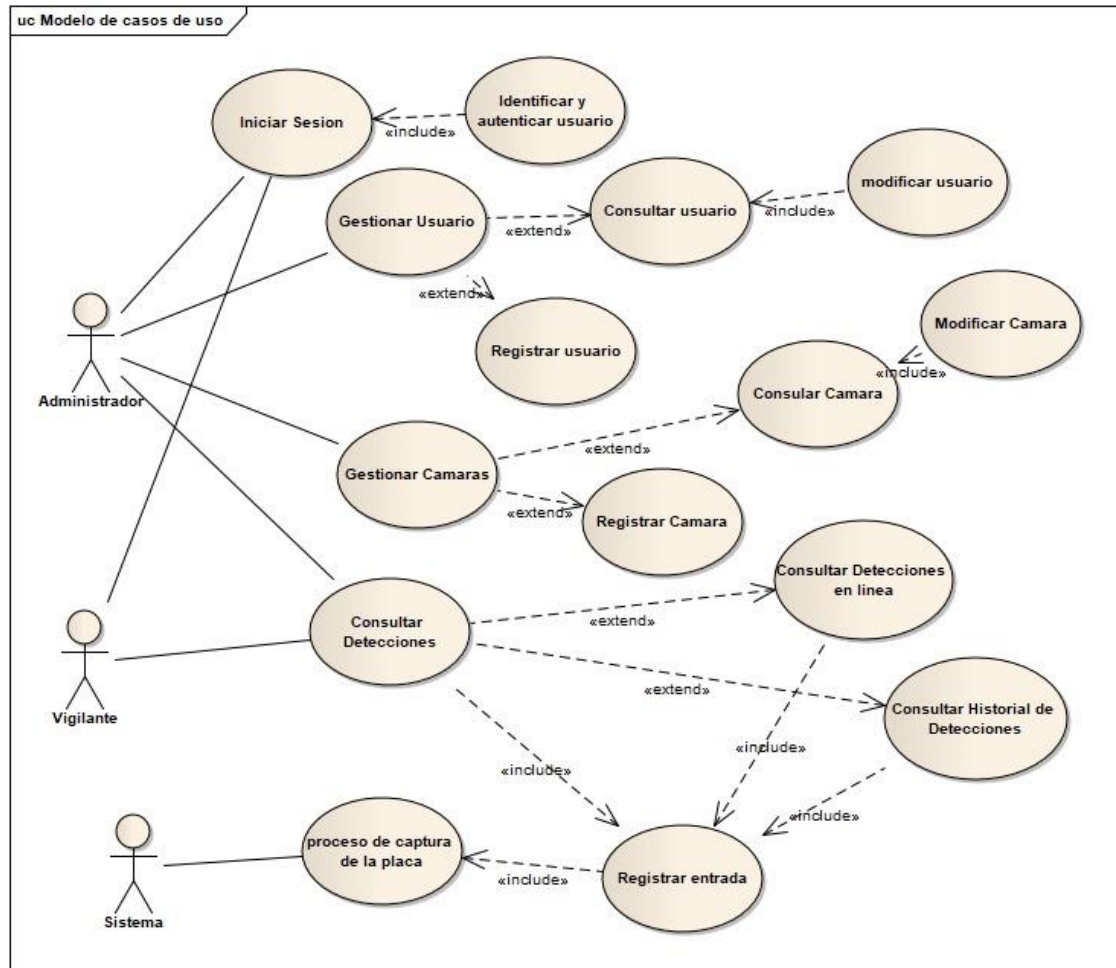
A partir del diseño del diagrama entidad-relación (ER) se representa el modelo Relacional, esto se debe a que del conjunto de entidades y relaciones establecidas en el diagrama se extraen la estructura general de la base de datos como lo son las tablas.

El modelo se divide en dos partes de acuerdo al diseño y la explicación de la funcionalidad que se realizó en el diagrama.

Ver en Anexo D, creación de las tablas.

3.2.4 Diagrama de casos de uso

Figura 14 Casos de uso, Fuente Autor



En los Casos de usos se representa la lista de tareas que realizan los actores y está directamente relacionada con los requisitos del sistema de información detallados en el análisis del problema, los cuales se deben alcanzar en el proyecto.

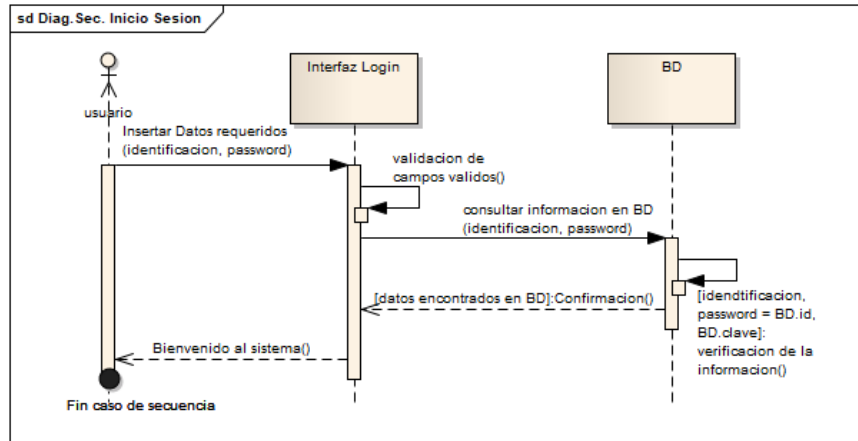
Para el prototipo se definieron 3 actores:

- Administrador
- Vigilante
- El sistema tendrá el proceso de la captura de la placa esto se hará automáticamente y lo enviará a la base de datos registrando la entrada.

3.2.5 Diagramas de secuencia

Autenticación de sesión

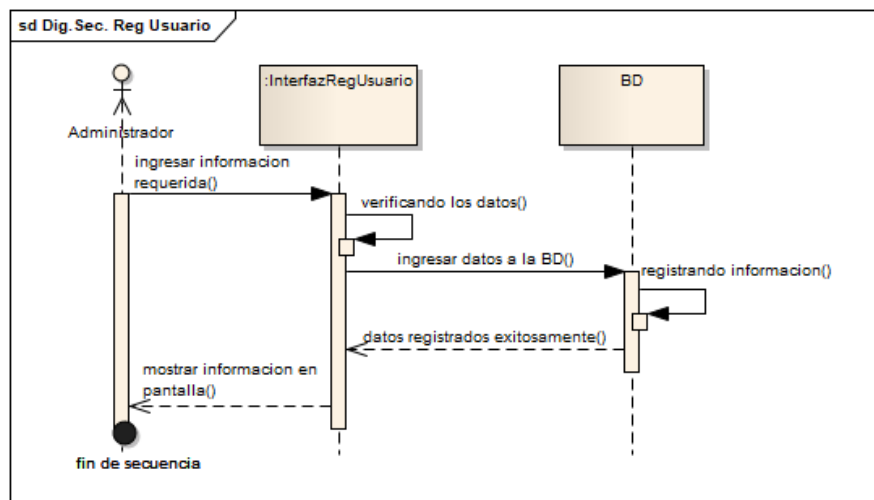
Figura 15 Diagrama de secuencia- Iniciar sesión, Fuente Autor



Para iniciar sesión es necesario estar previamente registrado en la base de datos como usuario, para que el ingreso sea exitoso.

Registra Usuario

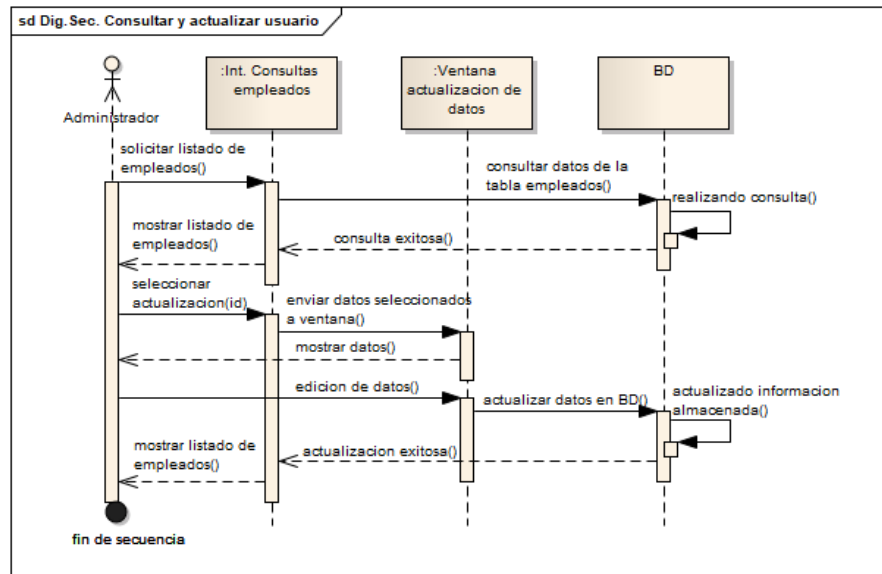
Figura 16 Diagrama de secuencia-Registra Usuario, Fuente Autor



El administrador es la persona permitida para realizar el ingreso a la base de datos de un nuevo un usuario, debe diligenciar los datos para tener un mensaje exitoso.

Consultar y actualizar usuario

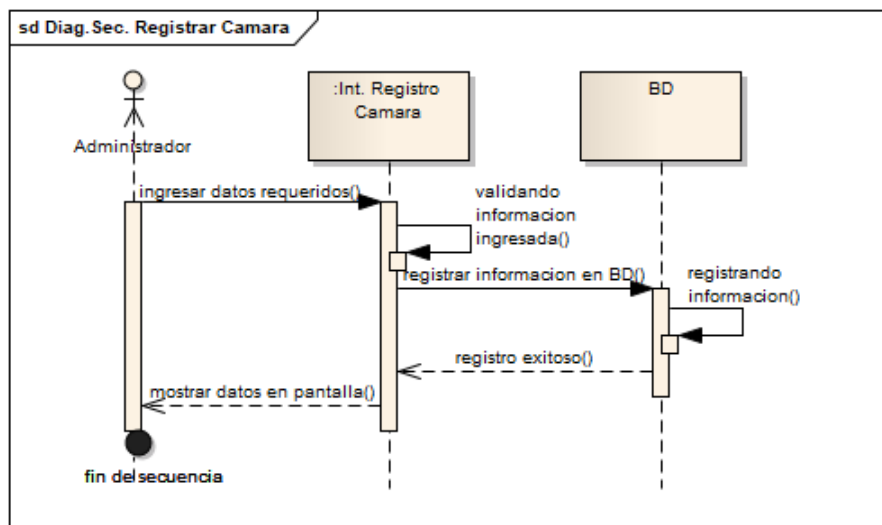
Figura 17 Diagrama de secuencia -Consultar y actualizar usuario, Fuente Autor



El administrador podrá visualizar a los usuarios a través de una lista y filtrar si desea. Para actualizar los datos debe seleccionar la opción y realizar los cambios.

Registrar cámara

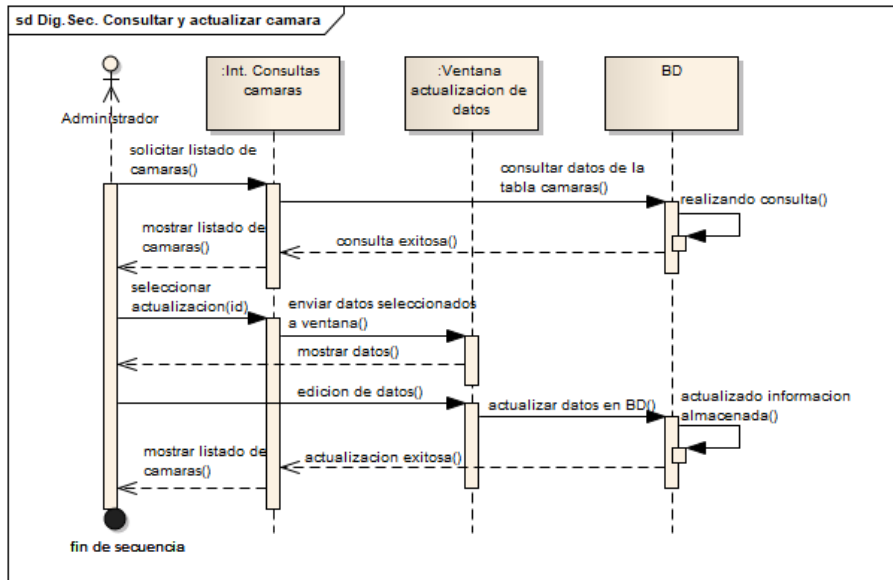
Figura 18 Diagrama de secuencia - Registrar cámara, Fuente Autor



El administrador es la persona permitida para realizar el ingreso a la base de datos de una nueva cámara, debe diligenciar los datos para tener un mensaje exitoso.

Consultar y actualizar cámara

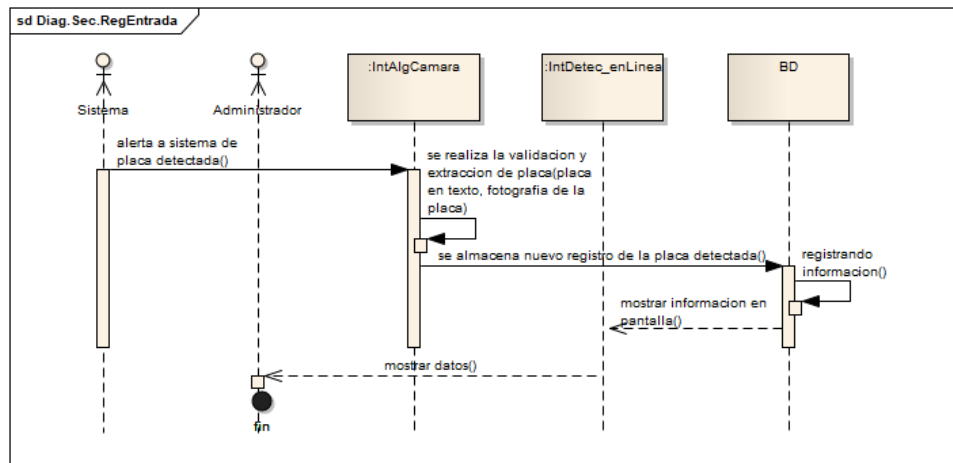
Figura 19 Diagrama de secuencia -Consultar y actualizar cámara, Fuente Autor



El administrador podrá visualizar las cámaras registradas a través de una lista y filtrar si desea. Para actualizar los datos debe seleccionar la opción y realizar los cambios.

Registrar entrada

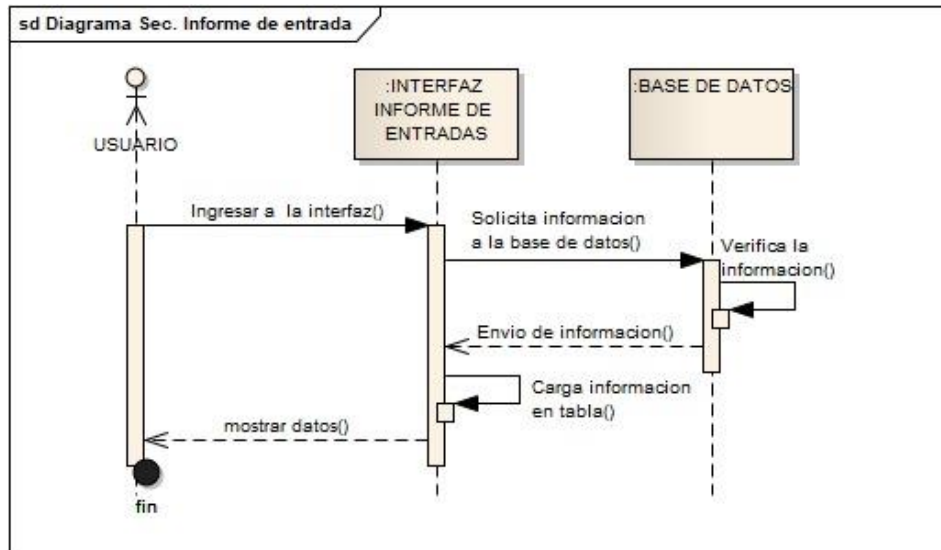
Figura 20 Diagrama de secuencia -Registrar entrada, Fuente Autor



El sistema realizará el ingreso de forma automática, reconociendo por medio de la RASPBERRY la placa y enviando a la base de datos la información de la entrada. El usuario podrá ver la entrada a través de la aplicación Web.

Consultar detecciones

Figura 21 Diagrama de secuencia-Consultar detecciones, Fuente Autor

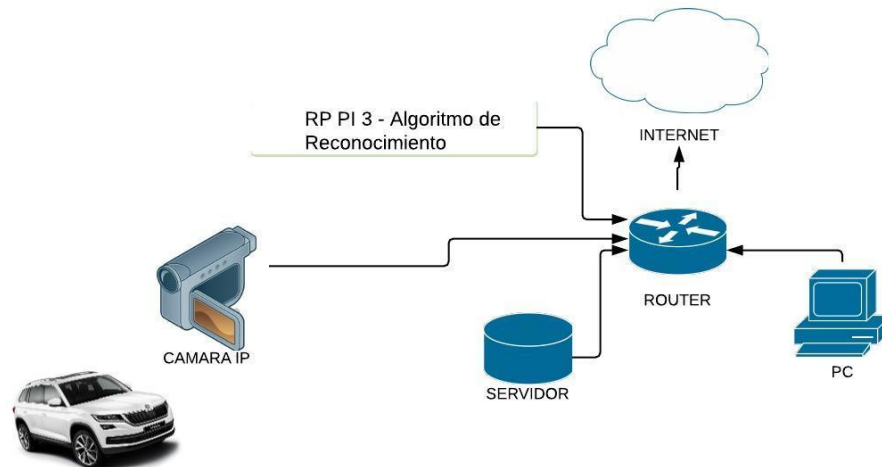


Los usuarios podrán consultar las entradas realizadas en el día, igualmente el histórico de las mismas, esto atreves de una lista que puede ser filtrada con la información requerida.

3.3 DISEÑO DEL PROTOTIPO DE RECONOCIMIENTO DE PLACAS

3.3.1 Arquitectura de red

Figura 22 Arquitectura de red, Fuente Autor



A continuación se nombran las tecnologías usadas en el proyecto. Estas se componen de los dispositivos de hardware:

- **Cámara IP**, ver Anexo E.
- **Raspberry PI 3 (RP).**
- **Servidor:** se configuró como servidor de Protocolo de transferencia de archivos (FTP) para acceder a las carpetas de este desde la RP, debido a que las imágenes capturadas de los vehículos se envían a dichas carpetas para su posterior visualización en la aplicación Web, de igual manera se configuró la base de datos y la aplicación web, la cual puede ser accedida desde cualquier computador que se encuentre conectada a la misma red.
- **Router**

Todos los dispositivos se conectan al Router, el cual les da salida a Internet, este será el medio de comunicación entre los mismos. El servidor tendrá una IP fija al igual que la cámara.

4 DESARROLLO

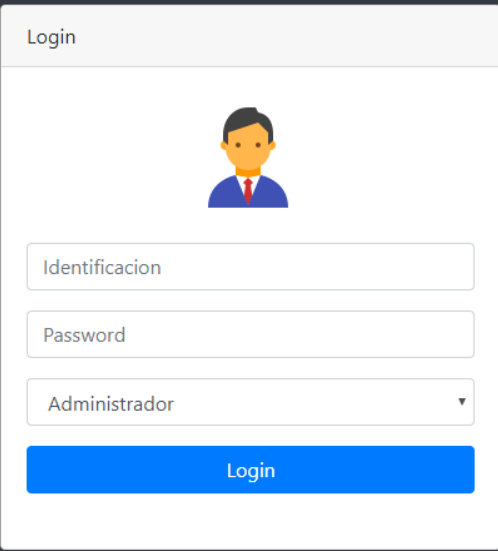
4.1 DESARROLLO DE APLICACIÓN WEB

La elaboración de la aplicación web esta creada bajo el lenguaje de programación PHP, HTML, JavaScript, JQuery, Ajax y el framework de Bootstrap 3. Se crearon tres módulos que permiten manejar la información almacenada en la base datos desde el servidor de MySQL mediante una cadena de conexión especifica (véase *anexo C*)

Interfaz de inicio de sesión

Esta interfaz permite identificar el tipo de sesión que se mantendrá durante la ejecución de la aplicación que podría variar entre “*administrador*” y “*vigilante*” teniendo en cuenta los permisos que puede tener cada uno de ellos, el usuario deberá ingresar su *número de identificación, contraseña y tipo de usuario*, se verificará la existencia del usuario mediante una función Ajax que realiza un llamado a la clase *logueame.php* que se conecta directamente a la base de datos (véase *anexo C*), después de verificar la existencia del registro la ventana lo llevara a la interfaz correspondiente.

Figura 23 Interfaz de inicio de sesión, Fuente Autor



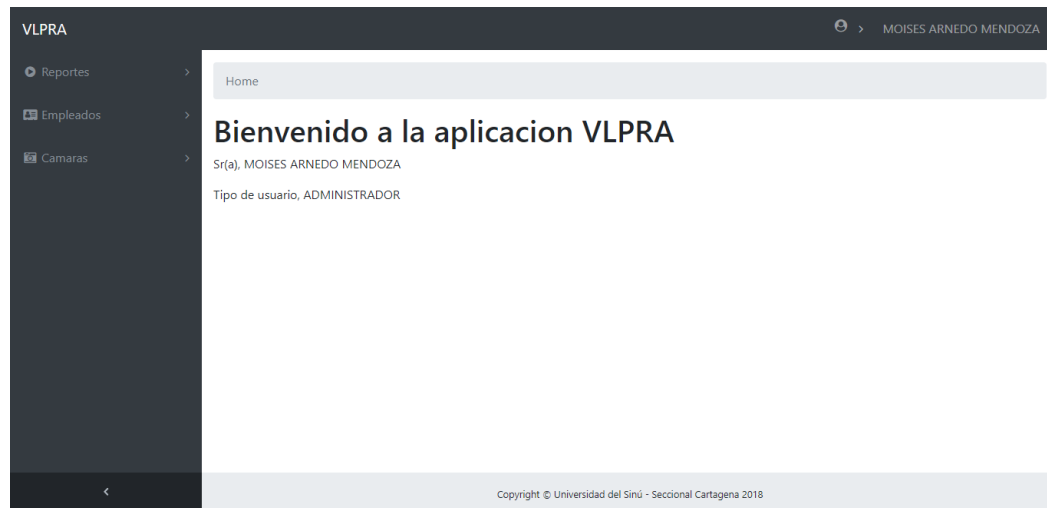
The image shows a login form with the following elements:

- Title: Login
- Icon: A stylized user icon in a blue suit.
- Input fields:
 - Identificacion
 - Password
 - Administrador (with a dropdown arrow)
- Button: A blue button labeled Login.

Interfaz de Bienvenida

La ventana principal de la aplicación (figura 23) tendrá en la parte izquierda el menú para acceder a los diferentes módulos creados para el manejo de la aplicación, todas las interfaces están elaboradas con Bootstrap 3 y son responsive, en la parte superior derecha tendremos el nombre del usuario que inicio la sesión y un Dropdown con la opción Logout que hace llamado a la clase *finalizar.php* que se encargará de destruir la sesión iniciada. Puede ver el documento del Anexo C para ver el código fuente.

Figura 24 Interfaz de Bienvenida, Fuente Autor



Los tres módulos creados fueron:

Reportes: es el modulo donde se tendrá acceso a las detecciones que se ha realizado, tiene dos opciones de interfaces que son “*Detecciones en línea*” e “*Historial de detecciones*”.

Empleados: en este módulo se visualizará una interfaz con el listado de empleados, permite registrar y actualizar los datos de los mismos.

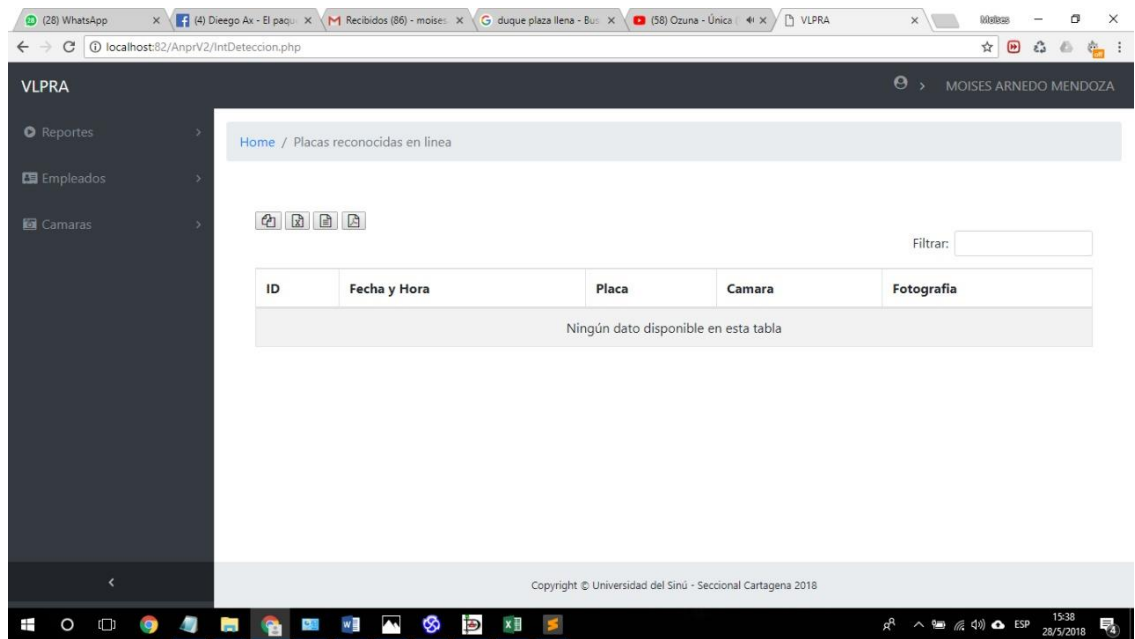
Cámaras: este módulo visualizara una interfaz con el listado de las cámaras registradas, permite registrar y actualizar los datos de las mismas.

Interfaz de Detecciones en línea

Esta interfaz es creada para que el usuario pueda verificar las detecciones que ha realizado el algoritmo de reconocimiento y a su vez mostrarlas en una tabla dinámica creada con el plugin *Datatables* que se actualiza cada 2000 milisegundos mediante la función *setInterval (CrearModal, 2000)* que se encuentra en la clase *CargarDetecciones.js*, esta contiene un *JSON* que carga la información de la base de datos a la tabla para mayor detalle *Ver anexo C*.

En la tabla se listarán los vehículos que han ingresado en el transcurso del día, en la columna "*Fotografía*" se cargará la imagen capturada en la Raspberry y los datos son cargados desde la clase *funcionCargaDeteccion.php* que extrae la información de la base de datos, a su vez crea un componente HTML de etiqueta ** para colocar la imagen. Adicionalmente se configuró un campo *filtrar* para realizar consultas parametrizadas.

Figura 25 Interfaz de Detecciones en línea, Fuente Autor



Interfaz de Historial de Detecciones

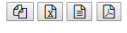
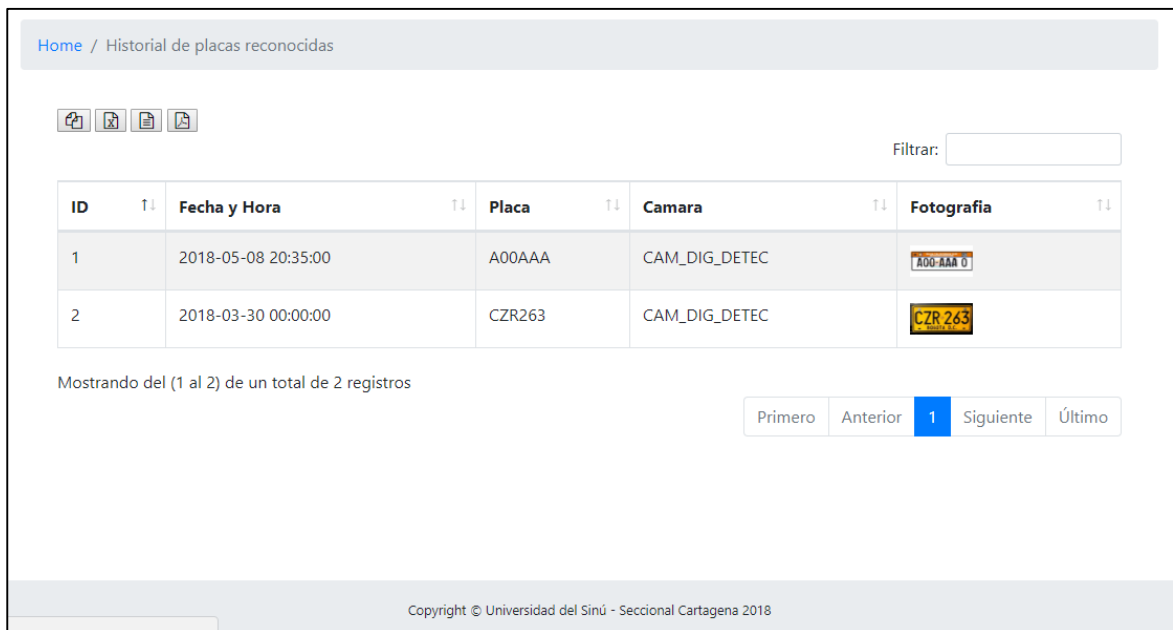

Al igual que la interfaz anterior muestra las detecciones realizadas, la diferencia se encuentra en que está cargará el listado de todas las detecciones que se han capturado como historial de la aplicación. Adicionalmente se puede generar reportes en diferentes tipo de documentos con estos botones , el primero está configurado para copiar la tabla, el segundo genera un archivo .xls, el tercero genera un archivo plano CSV y el cuarto genera un archivo .pdf (vea anexo C).



Figura 26 Interfaz de Historial de Detecciones, Fuente Autor



Home / Historial de placas reconocidas



Filtrar:

ID	Fecha y Hora	Placa	Camara	Fotografia
1	2018-05-08 20:35:00	A00AAA	CAM_DIG_DETEC	
2	2018-03-30 00:00:00	CZR263	CAM_DIG_DETEC	

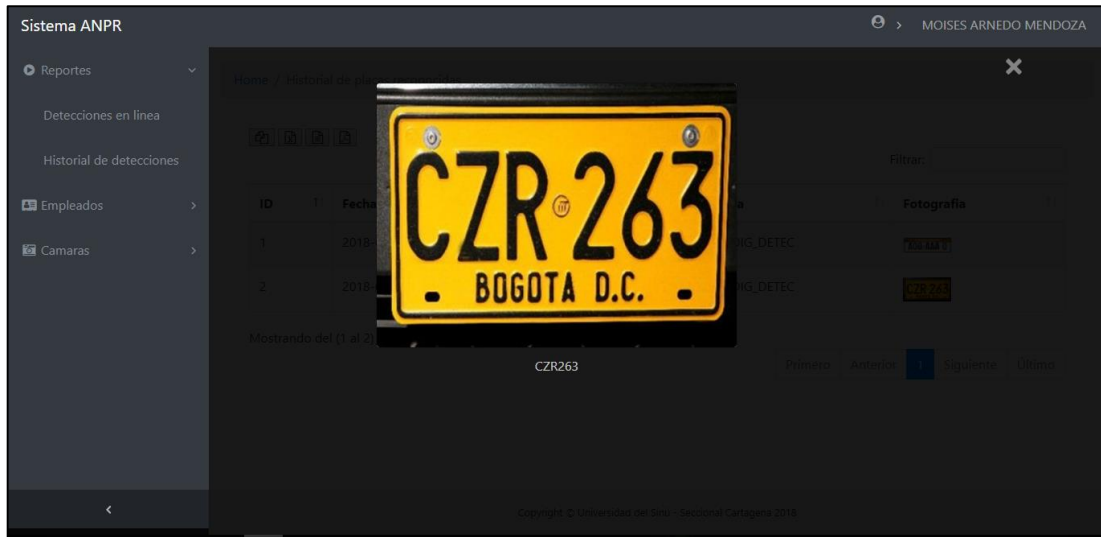
Mostrando del (1 al 2) de un total de 2 registros

Primero Anterior 1 Siguiente Último

Copyright © Universidad del Sinú - Seccional Cartagena 2018

Se puede observar en la figura 26 el modal que muestra la foto del vehículo, para análisis del código implementado ver anexo C.

Figura 27 Imagen de foto guardada en base de datos, Fuente Autor



Interfaz Gestionar Empleados

Muestra el listado de los empleados registrados en el sistema, a su vez permite crear nuevos empleados y actualizar los datos. Se utiliza una tabla elaborada con el plugin Datatables.

Figura 28 Interfaz de listado de empleados, Fuente Autor

Home / Listado de Empleados

[+ Agregar Empleados](#)

Mostrar registros Filtrar:

Identificacion <small>↑↓</small>	Nombres <small>↑↓</small>	Primer Apellido <small>↑↓</small>	Segundo Apellido <small>↑↓</small>	Direccion <small>↑↓</small>	E-mail <small>↑↓</small>	Telefono <small>↑↓</small>	Rol <small>↑↓</small>	Editar <small>↑↓</small>
1	1	1	1	1	1	1	VIGILANTE	Editar
2	1	1	1	1	1	1	ADMINISTRADOR	Editar
959595	PEDRO	PEREZ	PEREZ	PASEO BOLIVAR	PEDRO@HOTMAIL.COM	3163126071	VIGILANTE	Editar
45494360	NELCY	MENDOZA	DE AVILA	NO TIENE	DOZA@GMAIL.COM	316312	ADMINISTRADOR	Editar
73102357	MOISES	ARNEDO	PADILLA	TORICES	MOISES@HOTMAIL.COM	3163126071	VIGILANTE	Editar
1047451298	MOISES	ARNEDO	MENDOZA	TORICES	moises.arnedo14@gmail.com	3046722338	ADMINISTRADOR	Editar

Se configuro un botón llamado “*Agregar Empleados*” quien hace un llamado a una ventana modal con los campos disponibles para registrar la información en la base de datos, el botón *registrar* es vigilado por un *script* que captura el evento de *onclick()* y realiza un llamado a la clase *AddEmpleado.php* esta misma prepara los datos y los envía a la base de datos.

Figura 29 Interfaz Registrar empleados, Fuente Autor

The screenshot shows a web application interface for 'Sistema ANPR'. A modal window is open for adding a new employee. The modal contains the following fields:

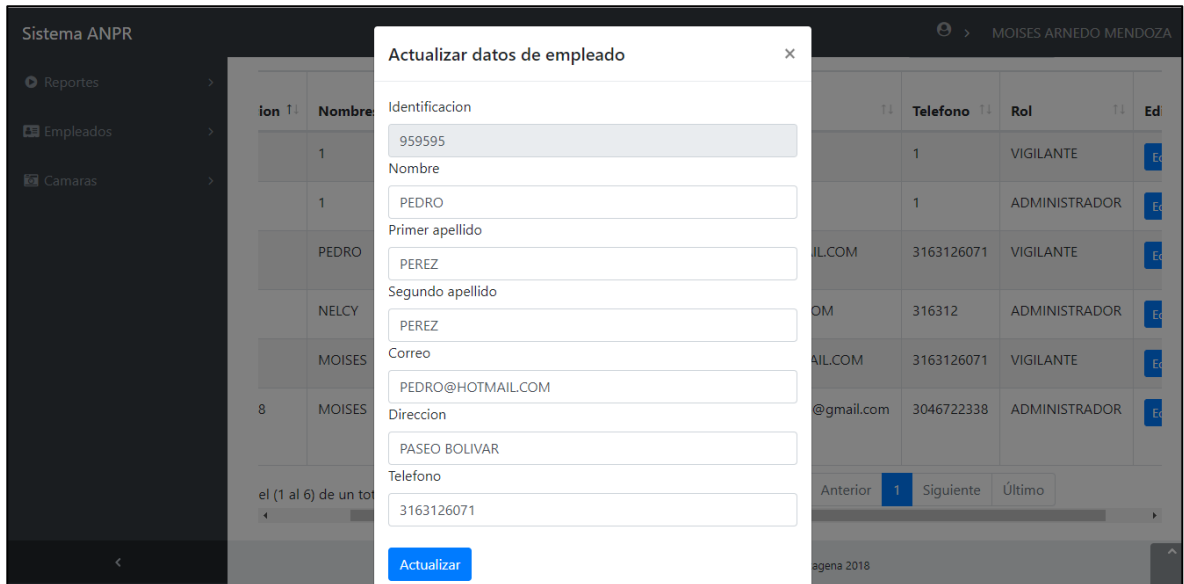
- Identificación:
- Nombre:
- Primer apellido:
- Segundo apellido:
- Correo:
- Dirección:
- Teléfono:
- Clave:
- Rol de empleado:

At the bottom of the modal is a blue button labeled 'Registrar'. In the background, a table displays a list of employees:

Identificación	Nombres	Telefono	Rol	Ed
1	1		VIGILANTE	<input type="button" value="E"/>
2	1		ADMINISTRADOR	<input type="button" value="E"/>
959595	PEDRO	3163126071	VIGILANTE	<input type="button" value="E"/>
45494360	NELCY	316312	ADMINISTRADOR	<input type="button" value="E"/>
73102357	MOISES	3163126071	VIGILANTE	<input type="button" value="E"/>
1047451298	MOISES	3046722338	ADMINISTRADOR	<input type="button" value="E"/>

De igual manera se configuro un botón “*Editar*” integrado en la tabla, este contiene la información de la fila en variables *data* y es creado con la tabla en la clase *funcionCargaEmpleados.php* ver *anexo C*. El botón *editar* hace un llamado a un modal enviando los datos que tiene temporalmente en las variables *data* a los campos integrados en el modal, en la ventana modal se configuró un botón *Actualizar* este es vigilado por un *script* que captura el evento *onclick()* para enviar la información a la clase *actualizaemp.php*; esta clase se encarga de preparar los datos y actualizarlos en la base de datos. Ver *anexo C*.

Figura 30 Interfaz Actualizar empleados, Fuente Autor



Interfaz de Cámaras

Permite la visualización de las cámaras registradas en la base de datos al igual que la creación de nuevas y la actualización de las mismas.

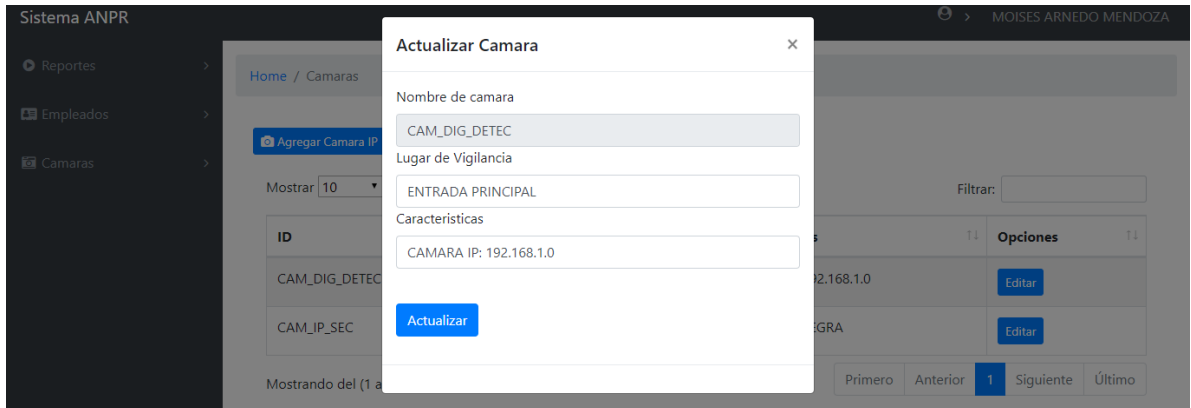
Figura 31 Interfaz de Cámaras, Fuente Autor



En la columna *opciones* de la tabla, se agrega un botón llamado *editar* que dispone de variables tipo data que contienen los valores de cada columna por fila,

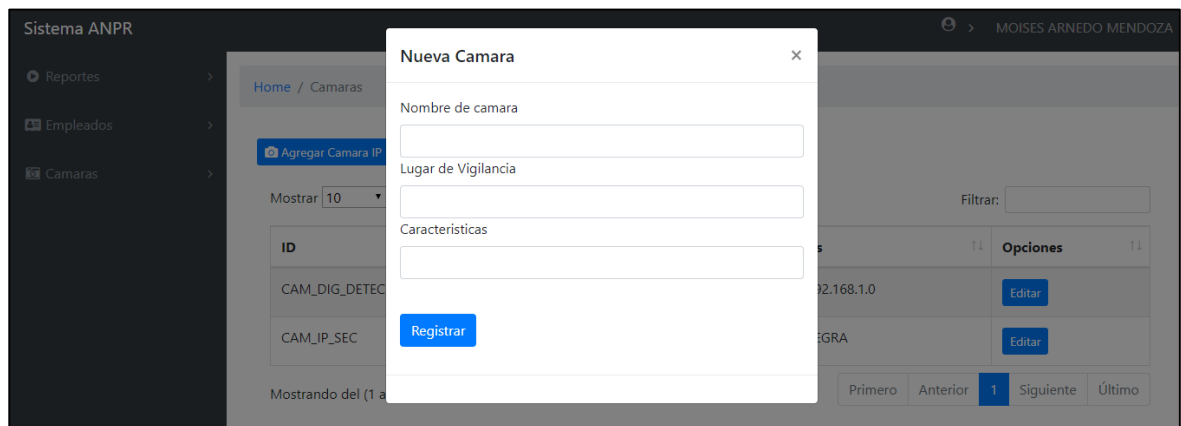
este botón llama un modal enviándole estas variables y mostrándolas en los campos correspondientes. Este modal es supervisado por un *script* que captura el evento *onclick()* del botón *actualizar*, al momento de este ser presionado realiza un llamado a la clase *actualizaCam.php* que se encarga de actualizar la información en la base de datos. Véase anexo C.

Figura 32 Interfaz Actualizar Cámara, Fuente Autor



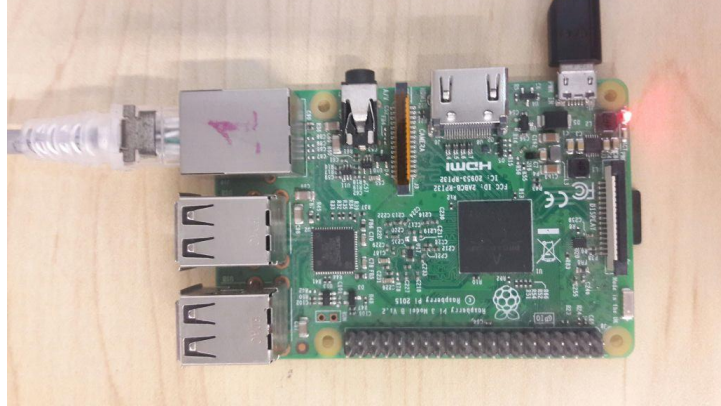
Se configuro un botón llamado “Agregar Cámara IP”, este llama una ventana modal con los campos disponibles para registrar una nueva cámara en el sistema. El modal es supervisado por un *script* que captura en el evento *onclick()* del botón *registrar* haciendo un llamado a la clase *AddCam.php* que se encarga de enviar la información a la base de datos. Ver anexo C.

Figura 33 Interfaz Registrar Cámara, Fuente Autor



4.2 RASPBERRY PI 3 MODEL B

Figura 34 Raspberry PI 3 implementada, Fuente Autor



Su elección está motivada por el hecho de que esta unidad cumple con las siguientes características: cantidad de software disponible, bajo coste, alto volumen de documentación disponible, bajo consumo energético, tamaño y peso reducidos y puede ser transportado fácilmente.

Comparación de modelos de Raspberry Pi

Las principales características de los modelos de Raspberry Pi se detallan a continuación:

Modelo	Raspberry Pi 1 B+	Raspberry Pi 2 B	Raspberry Pi 3 B
Fecha	2012 Feb 15	2015 Feb 1	2016 Feb 29
SOC	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837
Core	ARM1176JZF-S	Cortex-A7	Cortex-A53 64-bit
Nº Cores	1	4	4
CPU Clock	700 MHz	900 MHz	1.2 GHz
RAM	512 MB	1 GB	1 GB
Memoria	Micro SD	Micro SD	Micro SD
USB	2	4	4

Ethernet	SI	SI	SI
Wi-Fi	No	No	SI
Bluetooth	No	No	SI
GPIO	8	17	17
Consumo	700 mA	820 mA	1400 mA

Tabla 17 Modelos de Raspberry pi, Tomado de [33]

Con base a las características de los modelos de las RP se escogió la 3 B, este cuenta con un procesador mucho más rápido y potente que las versiones anteriores. Se trata de un ARM Cortex A53 de 4 núcleos, 1.2 Ghz y 64 bits. Su rendimiento es al menos un 50% superior a la versión 2 de Raspberry, la necesidad de contar con una CPU potente viene dada por el elevado número de operaciones exigentes que se realizan: procesamiento de imágenes, transmisión de video y comunicaciones, igualmente se incluyó un cambio en la conectividad, ya que instaló Bluetooth y Wifi.

Para el montaje de la Raspberry Pi (RP) es necesario tener los siguientes materiales:

- **Tarjeta SD:** Este es un punto importante para la configuración de la RP. La tarjeta instalada es Kingston clase 4 de 32GB. En esta tarjeta se instala el sistema operativo (SO), siempre debe de estar conectada, la información de arranque de la placa estará en la tarjeta.
- **Alimentación:** Para alimentar nuestra placa necesitaremos un cargador micro usb de 5V.
- Para la conexión con la RP se debe conectar un cable de red o bien un adaptador inalámbrico (normalmente plug and play).

Instalación y configuración del sistema operativo Raspbian

Raspbian es el sistema operativo optimizado de la raspberry PI y viene preinstalado con una gran cantidad de software para la educación, programación y uso general. Tiene Python, Scratch, Sonic Pi, Java, Mathematica y más. Incluye

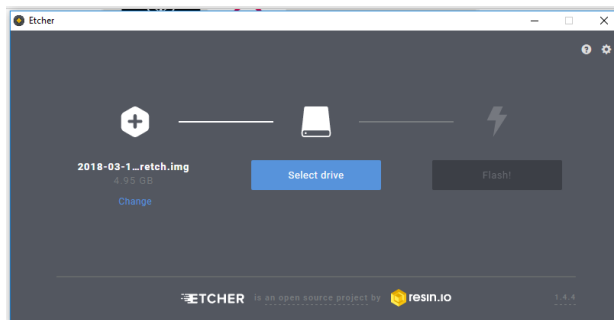
más de 35.000 paquetes que permiten obtener el mayor rendimiento posible y una interfaz sencilla.

Para descargar Raspbian se ingresa a la página de RP <https://www.raspberrypi.org/downloads/> seleccionando la versión 2018-03-13.

Descargar Etcher desde su página oficial <https://etcher.io/> , siendo este una aplicación gratuita y de código abierto desarrollado para grabar fácilmente imágenes ISO o IMG a memorias USB.

Ejecutar etcher y seleccionar el archivo imagen anteriormente descargado

Figura 35 Ejecución de etcher, Fuente Autor



Seleccionar la SD a instalar en la RP, después presionar Flash, con esto se inicia el cargue del archivo imagen en la SD. Al finalizar el proceso exitosamente mostrará el mensaje de “Flash Complete”, así la SD esta lista para instalar en la raspberry pi 3.

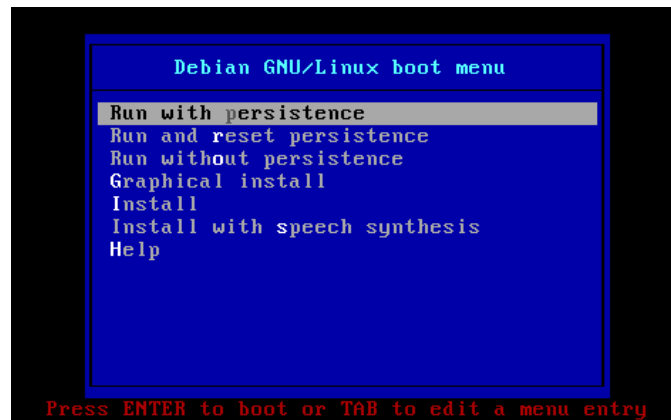
Figura 36 Instalar SO Raspbian a SD, Fuente Autor



Se inserta la SD en la ranura de la RP. Para visualizar la información de la RP se conecta una pantalla por medio del cable HDMI, el dispositivo debe tener todos los periféricos conectados.

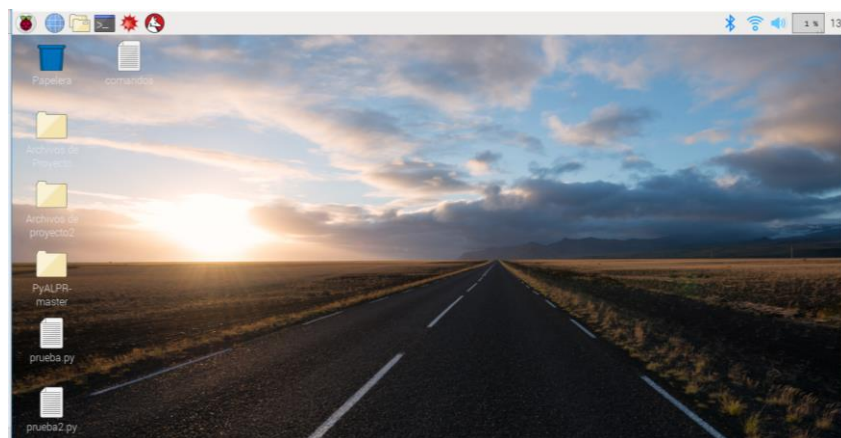
Se instala el tipo de sistema operativo, correr con persistencia.

Figura 37 Instalar SO Raspbian, Fuente Autor



Se visualizará el escritorio como se visualiza en la Figura 35.

Figura 38 SO Raspbian, Fuente Autor



Para configurar la RP se inicia con la instalación de los paquetes, se ejecuta la ventana de consola por medio del comando `sudo apt-get install`.

Los paquetes instalados son:

- `sudo apt-get install p7zip p7zip-full`: instala el descomprimidor de archivos .zip
- `sudo apt-get update`: descarga la lista de paquetes de los repositorios y actualizaciones para obtener información sobre las versiones más recientes de los paquetes y sus dependencias.
- `sudo apt-get upgrade`: Instalar las actualizaciones descargadas.

Es necesario configurar un usuario ROOT con los comandos:

- `Sudo su`; se ingresa la clave deseada

Con “su root” se ingresa al usuario colocando la clave raspberry.

4.3 IDENTIFICACIÓN Y RECONOCIMIENTO DE PLACAS VEHICULARES

Lenguaje de programación

El lenguaje de programación escogido para la implementación del algoritmo de identificación y reconocimiento de placas vehiculares es Python, este es un lenguaje interpretado, en el que se pueden escribir sencillos programas o instrucciones y ejecutarlos directamente sin necesidad de compilación en código máquina, además dispone de miles de librerías con las que se consigue una productividad en el desarrollo de las aplicaciones.

Este lenguaje trabaja en diversos sistemas operativos siendo uno de estos Raspbian, el cual, en su distribución viene instalado. Por estas razones se escogió como lenguaje de programación.

Instalación de Python

Python se instala en el Sistema Operativo basado en Raspbian instalado en la Raspberry Pi. El primer paso es abrir una terminal, una vez abierta se escribe Python para comprobar si ya viene instalado el software Python. En caso de que saliera un mensaje de error, similar a *"Python: command not found"* se procederá a actualizar primero la lista de los repositorios tecleando *"sudo apt-get update"*, después el Sistema Operativo con *"sudo apt-get upgrade"* y por ultimo instalar Python con *"sudo apt-get install python2.7"*.

Algoritmo de identificación y reconocimiento de placas vehiculares

Un algoritmo es un conjunto de instrucciones que realizadas en orden conducen a obtener la solución de un problema. Las librerías son un conjunto de algoritmos prefabricados, que pueden ser utilizados por el programador para realizar determinadas operaciones.

OpenALPR, es una librería de reconocimiento de matrículas automáticas de código abierto, que analiza imágenes y secuencias de video para identificar placas de vehículos, esta implementa en su codificación librerías que se basa en técnicas de visión por computadora, robustas y potentes como lo son OpenCV y Tesseract OCR, las cuales son de código abierto y manejan algoritmos de aprendizaje automático, siendo esto una rama de la Inteligencia Artificial que tiene como objetivo desarrollar técnicas que permitan a las computadoras aprender.

La librería se divide en dos partes: la detección de la región que contiene la matrícula aplicando las funciones de opencv, luego aplicará una segmentación para obtener caracteres en la matrícula y el reconocimiento o identificación del número de matrícula por tesseract-ocr.

La escogencia de esta librería se da por ser específica en el campo de placas vehiculares y por su capacidad de ser entrenado.

Diseño OpenALPR

OpenALPR funciona como una tubería. La entrada es una imagen o secuencia de video, varios procesos se producen en etapas, y la salida son los posibles números de placa en la imagen.

Las etapas de interconexión se producen en el siguiente orden:

Fases	C++ clase	DESCRIPCIÓN
<i>Detección</i>	regiondetector.cpp	Encuentra regiones potenciales de matrículas
<i>Binarización</i>	binarizewolf.cpp	Convierte la imagen de la región de placa en blanco y negro

<i>Análisis de Char</i>	characteranalysis.cpp	Encuentra "blobs" del tamaño de un personaje en la región de la placa
<i>Bordes de placa</i>	platelines.cpp y platecorners.cpp	Encuentra los bordes / forma de la matrícula
<i>Deskew</i>	licenseplatecandidate.cpp	Transforma la perspectiva en una vista directa basada en el tamaño ideal de la matrícula.
<i>Segmentación de personajes</i>	charactersegmenter.cpp	Aísla y limpia los caracteres para que puedan ser procesados individualmente
<i>OCR</i>	ocr.cpp	Analiza cada imagen de personaje y proporciona múltiples cartas / confianzas posibles
<i>Postprocesamiento</i>	postprocess.cpp	Crea una lista superior de posibilidades de placa basadas en confianzas OCR. También realiza una coincidencia Regex contra plantillas de región si así lo solicita.

Tabla 18 fases de OpenALPR, Tomado de [34]

Instalación de OpenALPR

Para su instalación es necesario tener acceso a internet desde la Raspberry, esta se configuró desde la consola de la siguiente manera:

Ejecutar los comandos:

Instalar requisitos previos, se instalan Las Librerías OpenCv y tesseract, indispensable para su funcionamiento.

```
sudo apt-get install libopencv-dev libtesseract-dev git cmake build-essential libleptonica-dev
```

```
sudo apt-get install liblog4cplus-dev libcurl3-dev
```

Si usa el daemon, instale beanstalkd

```
sudo apt-get install beanstalkd
```



```
# Clona el último código de GitHub, donde está La Librería de OpenAlpr
git clone https://github.com/openalpr/openalpr.git

# Configurar el directorio de compilación

cd openalpr/src
mkdir build
cd build

# configurar el entorno de compilación

cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr -DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..

# Compilar la biblioteca
make

# Instale los binarios / bibliotecas en su sistema local (el prefijo es / usr)

sudo make install

# Pruebe la biblioteca
wget http://plates.openalpr.com/h786poj.jpg -O lp.jpg
alpr lp.jpg
```

4.4 INTEGRACIÓN DE LOS COMPONENTES

El prototipo de identificación y reconocimiento de placas vehiculares se compone de la siguiente manera:

A través de la cámara IP (Figura 40), ver características en Anexo E, se envía el video en tiempo real de los vehículos que están ingresando a la Universidad del Sinú, sede Santillana.

Figura 39 cámara IP, Fuente Autor



La Raspberry (Figura 34) recibirá el video de la cámara por medio de la red, al llegar un vehículo para su ingreso se captura uno de los fotogramas durante el streaming de la cámara la cual se procesa por medio de la aplicación. Esto a través de la librería OpenALPR determina si se encuentra placa en la imagen y realiza el reconocimiento de los caracteres. Al detectar la placa se envía el número de la placa y el ID de la cámara que captura a la base de datos, la imagen de la captura se guarda en el servidor, esto se realiza por medio de carpetas compartidas del servidor a la Raspberry.

La información registrada se visualiza desde la aplicación Web, esta se encuentra en el servidor y puede ser accedida desde los navegadores de internet.

5. PRUEBAS DEL PROTOTIPO

5.1 PRUEBAS A LA APLICACIÓN WEB

Las pruebas realizadas a la aplicación web se ejecutaron de dos tipos, una a la interfaz gráfica y otra de funcionalidad con respecto a los requerimientos planteados. Se implementó un ciclo de prueba al no obtener no conformidades.

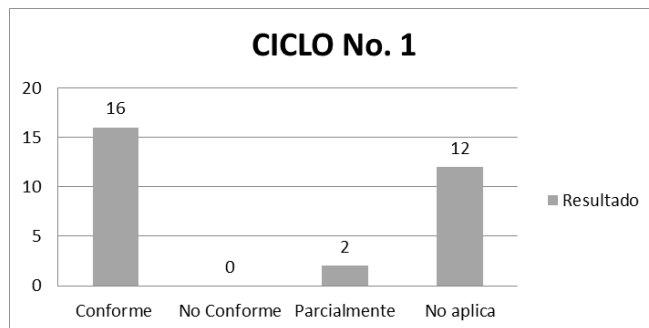
5.1.1 Pruebas de interfaz gráfica para requerimientos Web

Esta prueba consta de 30 ítems en los que se revisan los requerimientos de acuerdo a las interfaces gráfica.

Modulo Reportes –Detecciones en línea

En este módulo no se obtuvo no conformes, pero si, dos parcialmente, que corresponden a la ubicación del cuadro de texto con respecto a la etiqueta y la repartición de las celdas dentro de la grilla, estas se ubican de acuerdo al diseño establecido. Para mayor detalle ver Anexo F.

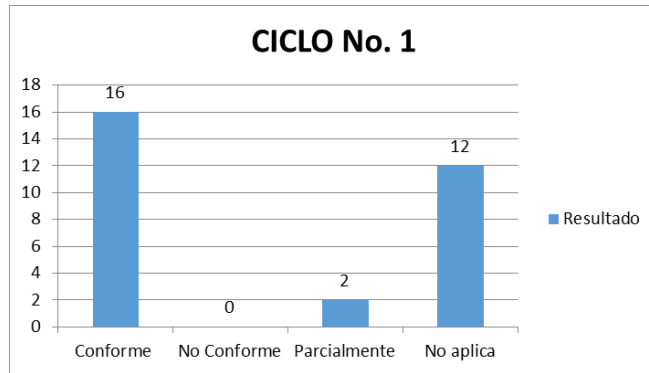
Figura 40 Prueba-Detección en línea, Fuente Autor



Modulo Reportes – Historial de detecciones

En este módulo no se obtuvo no conformes, pero si, dos parcialmente, que corresponden a la ubicación del cuadro de texto con respecto a la etiqueta y la repartición de las celdas dentro de la grilla, estas se ubican de acuerdo al diseño establecido. Para mayor detalle ver Anexo F.

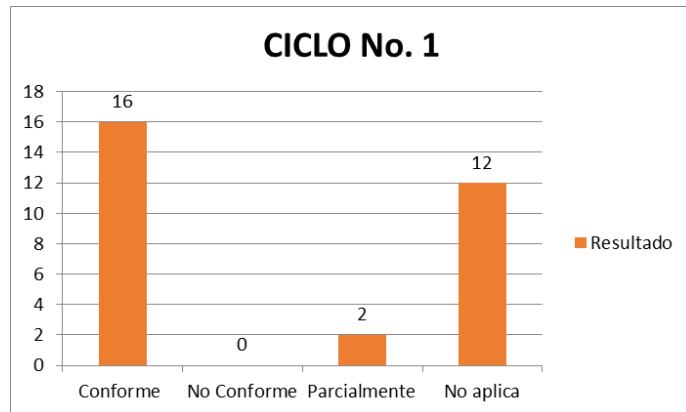
Figura 41 Prueba-Historial de detecciones, Fuente Autor



Modulo Empleados – Gestionar empleados

En este módulo no se obtuvo no conformes, pero si, dos parcialmente, que corresponden a la ubicación del cuadro de texto con respecto a la etiqueta y la repartición de las celdas dentro de la grilla, estas se ubican de acuerdo al diseño establecido. Para mayor detalle ver Anexo F.

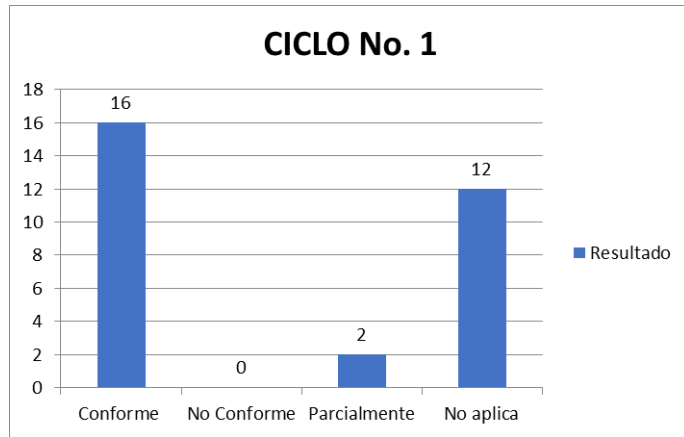
Figura 42 Prueba- Gestionar empleados, Fuente Autor



Módulo Cámaras – Gestionar cámaras

En este módulo no se obtuvo no conformes, pero si, dos parcialmente, que corresponden a la ubicación del cuadro de texto con respecto a la etiqueta y la repartición de las celdas dentro de la grilla, estas se ubican de acuerdo al diseño establecido. Para mayor detalle ver Anexo F.

Figura 43 Prueba- Gestionar cámaras, Fuente Autor



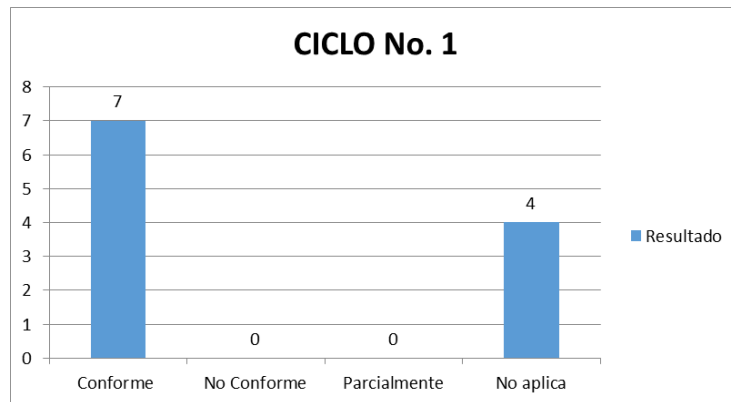
5.1.2 Pruebas de funcionalidad para los requerimientos

Esta prueba consta de 11 ítems en los que se revisan los requerimientos funcionales del prototipo.

RF01 – Autenticación de Usuario

Con las pruebas realizadas se evidenció que el aplicativo en siete condiciones se encuentra conforme y en cuatro no aplica, Para mayor detalle ver Anexo F en este se encuentran los valores evaluados.

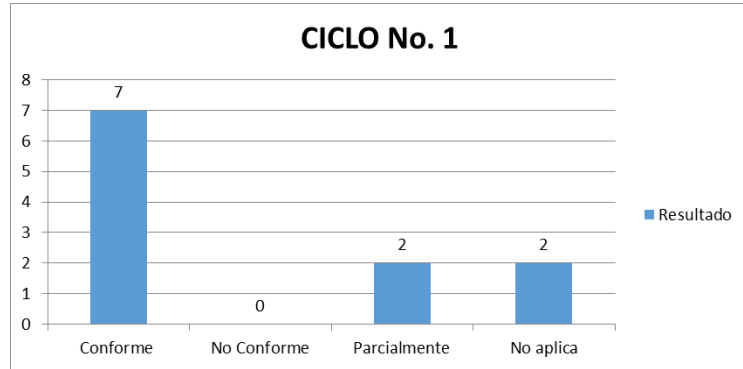
Figura 44 Prueba- Autenticación de usuario, Fuente Autor



RF02 – Gestionar Usuario

Con las pruebas realizadas se evidenció que el aplicativo en siete condiciones se encuentra conforme, en dos parcialmente y dos no aplica, Para mayor detalle ver Anexo F en este se encuentran las condiciones evaluadas.

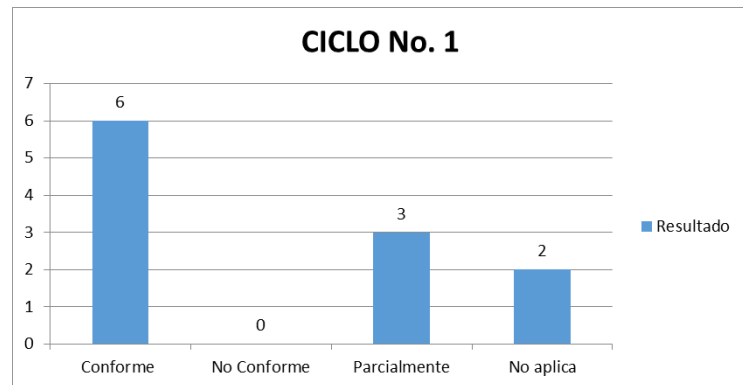
Figura 45 Prueba-Gestionar usuario, Fuente Autor



RF03 – Registrar Usuario

Con las pruebas realizadas se evidenció que el aplicativo en seis condiciones se encuentra conforme, en tres parcialmente y dos no aplica, Para mayor detalle ver Anexo F.

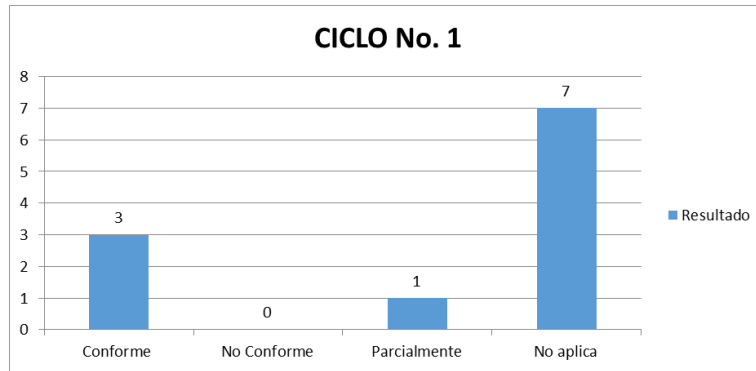
Figura 46 Prueba-Registrar usuario, Fuente Autor



RF04 – Consultar usuario

Con las pruebas realizadas se evidenció que el aplicativo en tres condiciones se encuentra conforme, en uno parcialmente y en siete no aplica, Para mayor detalle ver Anexo F.

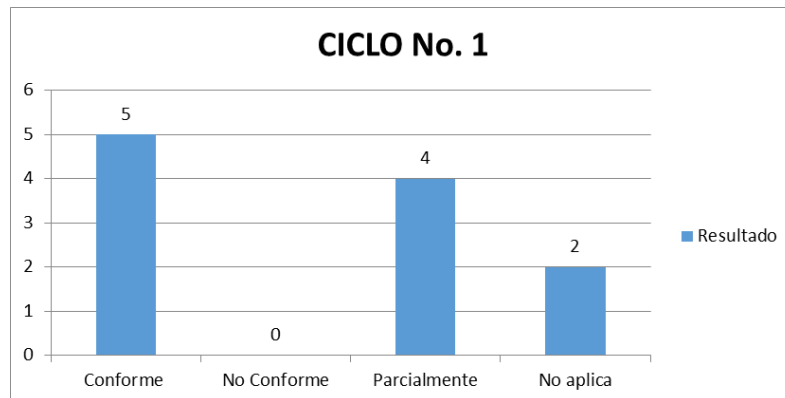
Figura 47 Prueba-Consultar usuario, Fuente Autor



RF05 – Modificar usuario

Con las pruebas realizadas se evidenció que el aplicativo en cinco condiciones se encuentra conforme, en cuatro parcialmente y en dos no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

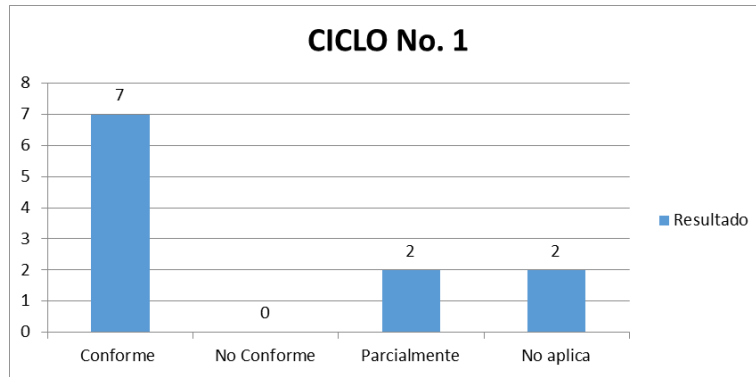
Figura 48 Prueba-Modificar usuario, Fuente Autor



RF06 – Gestionar Cámara

Con las pruebas realizadas se evidenció que el aplicativo en siete condiciones se encuentra conforme, en dos parcialmente y en dos no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

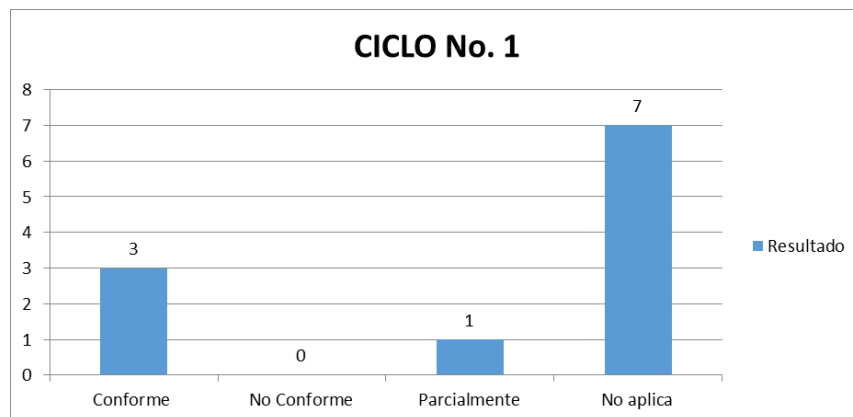
Figura 49 Prueba-Gestionar cámara, Fuente Autor



RF07 – Consultar Cámara

Con las pruebas realizadas se evidenció que el aplicativo en tres condiciones se encuentra conforme, en uno parcialmente y en siete no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

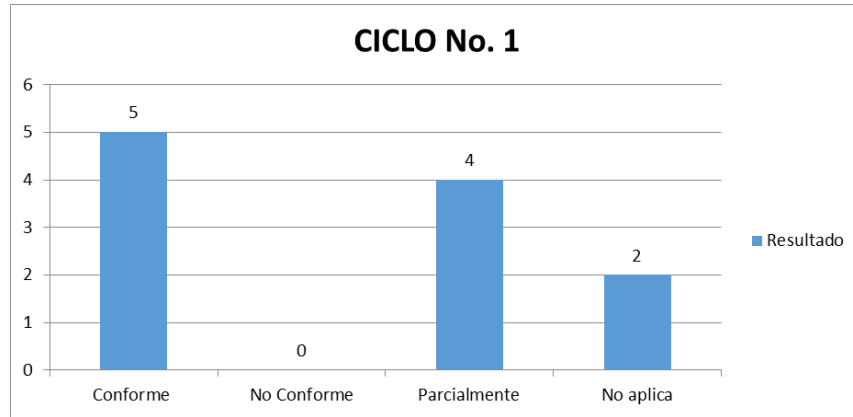
Figura 50 Prueba-Consultar cámara, Fuente Autor



RF08 – Modificar Cámara

Con las pruebas realizadas se evidenció que el aplicativo en cinco condiciones se encuentra conforme, en cuatro parcialmente y en dos no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

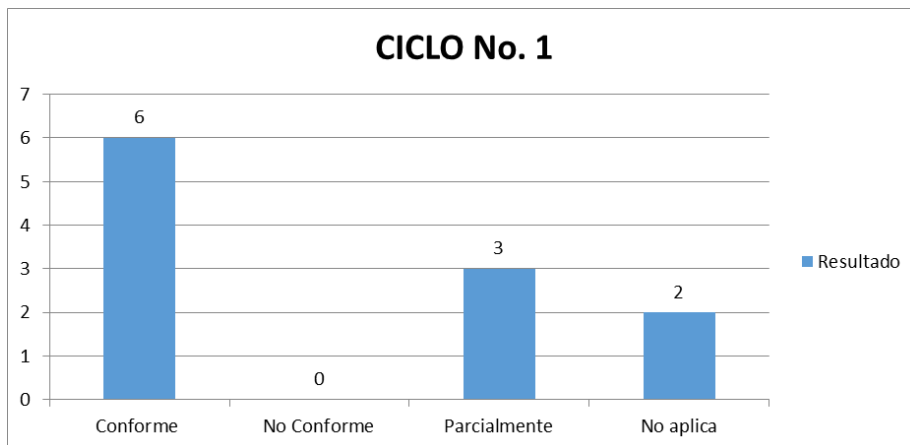
Figura 51 Prueba-Modificar cámara, Fuente Autor



RF09 – Registrar Cámara

Con las pruebas realizadas se evidenció que el aplicativo en seis condiciones se encuentra conforme, en tres parcialmente y en dos no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

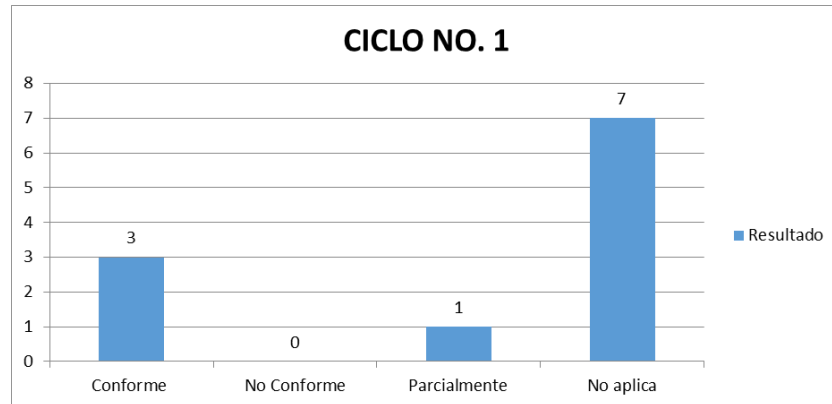
Figura 52 Prueba-Registrar cámara, Fuente Autor



RF10 – Consultar informe detecciones

Con las pruebas realizadas se evidenció que el aplicativo en tres condiciones se encuentra conforme, en uno parcialmente y en siete no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

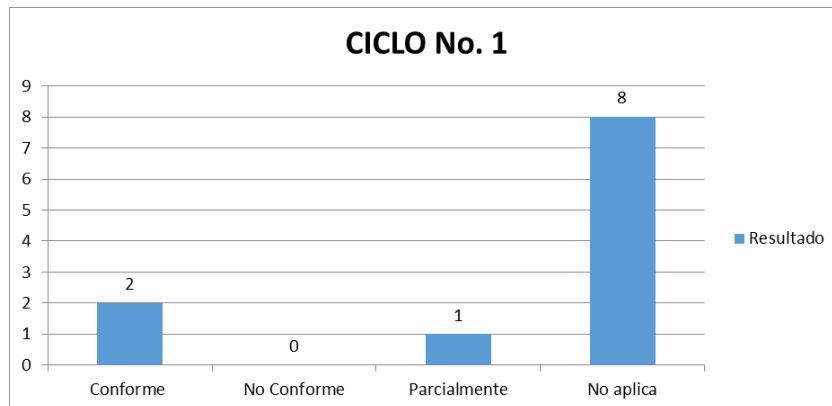
Figura 53 Prueba-Consultar informe detecciones, Fuente Autor



RF12 – Registrar Entradas

Con las pruebas realizadas se evidenció que el aplicativo en dos condiciones se encuentra conforme, en uno parcialmente y en ocho no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

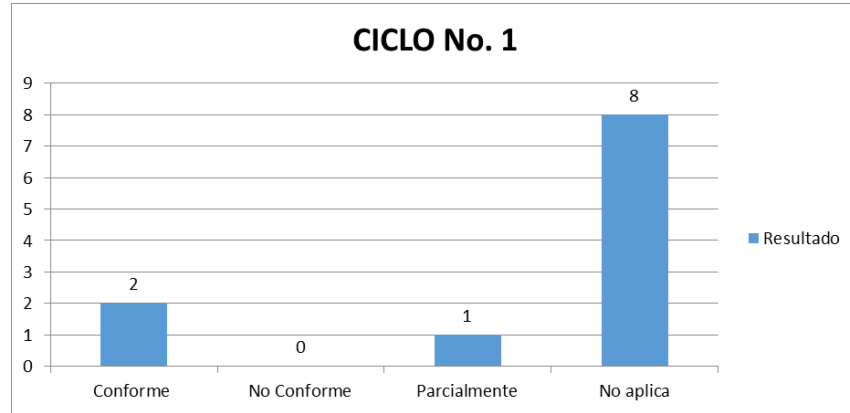
Figura 54 Prueba-Registrar entradas, Fuente Autor



RF13 – Captura automática de la placa

Con las pruebas realizadas se evidenció que el aplicativo en dos condiciones se encuentra conforme, en uno parcialmente y en ocho no aplica, Para mayor detalle ver Anexo F, en este se encuentran las condiciones evaluadas.

Figura 55 Prueba-Captura automática de la placa, Fuente Autor



5.2 PRUEBA DE RECONOCIMIENTO DE PLACAS VEHICULARES

Las pruebas de reconocimiento de placas vehiculares se realizaron a través de un ordenador que reproducía videos en los cuales se encontraban los vehículos, por medio de la cámara se capturaba dicho video y se enviaba a la Raspberry. Esto por la dificultad que conlleva el montaje del prototipo en sede, se es necesario de infraestructura de red en la entrada de la universidad la cual no se encuentra establecida en el alcance del proyecto.

A continuación, se detallan los resultados del reconocimiento:

Figura 56 Prueba de Reconocimiento, Fuente Autor

PLACA	FOTOGRAFÍA	CONDICION	PLACA RECONOCIDA	OBSERVACION
MSQ664		Normal	0SQ664	En la imagen capturada se evidencia un vehiculo al fondo a mayor distancia de la camara y la biblioteca reconoce primero dicho vehiculo con la placa DXT577, siendo la correcta HXT577.
UEZ524		DIAGONAL		No reconoce la placa
BLH789		PLACA DETERIORADA	7BL89	Error en la H7
MUP783		Normal	HUP783	Error en la M
MSQ664		Normal	S0664I	Error en la MQ
MGV395		Normal	MGV395	
HEY266		Normal	HEY266	
KHR013		Normal	KHR013	
MVW162		Normal	MV62	Error W1
LEA356		Normal	LEA356	
BXR194		Normal	BXR194	
FGL463		Normal	FGL63	Error en la 4

MLU937		Normal	MLU937	
MOV176		Normal	MOV176	
GNJ260		Normal	GNJ260	
JBV032		DIAGONAL		No reconoce la placa

Las pruebas se realizaron con 16 placas, cada una está formada por seis letras para un total de 96 caracteres por reconocer. Se presentaron tres condiciones en las que se probó el algoritmo, estos son: Imagen en diagonal, Imagen de frente (normal) e imagen con placa deteriorada.

Los resultados obtenidos son los detallados en la imagen anterior. De acuerdo a la condición se evidenció:

- Imagen en diagonal: Las imágenes donde la placa se encuentran diagonal el algoritmo no procesa, ya que no identifica la placa, esta condición se presentó dos veces para un total de 12 caracteres sin reconocer.
- Imagen frontal (normal): Son las imágenes en las cuales siempre hay un procesamiento, pero puede presentar errores en el reconocimiento de caracteres, esta condición se presentó 13 veces para un total de 78 caracteres de las cuales hubo error en 7 caracteres, siendo esta forma la más conveniente para capturar la placa.
- imagen con placa deteriorada: Al estar la placa del vehículo deteriorada y no tener unos caracteres definidos, puede presentarse errores al reconocer dichos caracteres, En la imagen detectada hubo error en dos caracteres y generó uno adicional al buscar dentro de su entrenamiento un carácter parecido.

6 RECOMENDACIONES

Como recomendaciones para continuar el prototipo se sugiere:

- Utilizar una cámara que se adecue a la Raspberry PI 3 y a la red del lugar donde se desea instalar, debido a que la escogida tiene mucha resolución y con la red de la universidad tarda en el procesamiento del reconocimiento.
- Entrenar la librería OPENCV con las placas de Colombia, para obtener una mayor precisión en su identificación y reconocimiento.
- Validar que la placa reconocida este formada por tres letras y tres números si es vehículo o por tres letras, dos números y una letra si es motocicleta, esto para descartar los reconocimientos que no corresponden a placas vehiculares.
- Entrenar al algoritmo que a través del video en tiempo real se capture una fotografía si se aproxima un vehículo y no por procesamiento de fotogramas, este envía una foto constantemente a la Raspberry este o no este algún vehículo, lo cual, consume memoria de la RP.
- Enseñar a la librería OPENCV la diferencia entre caracteres semejantes, como: O y 0, M y H, Q y O, D y O, ya que suele presentarse errores de identificación con estas letras.
- Entrenar a la librería OPENCV la identificación de las placas vehiculares cuando estas se encuentren en algún grado de inclinación, ya que en las pruebas se determinó que no realiza tal reconocimiento.
- Realizar las pruebas de reconocimiento en el campo, ya que las realizadas no se ejecutaron de dicha manera por no contar con la infraestructura de red y así determinar la condición climática, la luminosidad, distancia máxima o mínima del vehículo con respecto a la cámara y todo lo que influya para que el algoritmo pueda realizar el procesamiento de placa.
- Establecer un reductor de velocidad a una distancia determinada para que el vehículo se deba detener y el algoritmo procese la placa con más facilidad.

7 CONCLUSIONES

Se ha construido un prototipo de identificación y reconocimiento de placas vehiculares de acuerdo al objetivo principal, con el propósito de controlar el acceso de estos a la Universidad del Sinú sede Santillana, automatizando el proceso de registro de la información mediante la librería OpenALPR que está instala desde el dispositivo Raspberry PI, igualmente brindándole al usuario la facilidad de realizar consultas de los historiales de los vehículos que han ingresado. Para llegar al cumplimiento de este objetivo se ejecutaron cuatro objetivos específicos, los cuales se realizaron de la siguiente manera:

- Para el análisis y el diseño del prototipo se plasmaron los capítulos dos y tres, en el primero se hizo el análisis del problema con el levantamiento de los requisitos funcionales y no funcionales de acuerdo a la referencia de Standard IEEE 830 – 1998, para el segundo se estableció el diseño de la solución, el cual se dividió en diseño de la aplicación, que consta de las interfaces que tendrían esta, en diagramas y en la arquitectura de red compuesta para su funcionamiento.
- Para el desarrollo de la aplicación Web Responsive se estableció el capítulo cuatro que se conforma del desarrollo total de esta, pero el encargado de detallar el cumplimiento del objetivo es el subcapítulo 4.1, en este se detalla las imágenes en las que está conformada la aplicación a partir de su funcionamiento.
- OpenALPR fue el algoritmo escogido para implementar en el prototipo, este se encuentra detallado en el subcapítulo 4.3, el cual tiene el lenguaje de programación escogido y la explicación de la librería. De igual manera se detallan las características de la Raspberry en el subcapítulo 4.2, siendo este el componente principal del proyecto.
- En la realización de las pruebas, detallado en el capítulo cinco se validó la funcionalidad del prototipo con las pruebas de la aplicación web que consta de prueba a la interfaz gráfica y prueba de funcionamiento de acuerdo a los requerimientos planteados y por ultimo prueba de reconocimiento, en el que se valida el alcance de este con respecto a varias condiciones en que se puede presentar un vehiculo.

BIBLIOGRAFÍA

- [1] M. J. S. Rodríguez Claudia and Godoy Juan Felipe, “CESED informe de seguridad 2016,” Universidad de los Andes, Colombia, Seguridad 6, Jul. 2017.
- [2] Bernal Lara Eduardo, “Accesos seguros en instituciones educativas,” *segurilatam*. [Online]. Available: <http://www.segurilatam.com/seguridad-integral/control-de-accesos/accesos-seguros-en-instituciones-educativas>.
- [3] Superintendencia de Vigilancia y Seguridad Privada, “Protocolo de Operación servicios de vigilancia y seguridad privada para el Sector Residencial,” *supervigilancia*, 2017. [Online]. Available: <https://www.supervigilancia.gov.co/publicaciones/5563/protocolo-de-operacion-servicios-de-vigilancia-y-seguridad-privada-para-el-sector-residencial/>.
- [4] M. R. L. D. Cindy Johanna González Díaz and Diego González Castillo, “Desarrollo de un sistema de visión artificial para el reconocimiento de placas en vehículos particulares,” Ingeniería Mecatrónica, Universidad de San Buenaventura, Bogotá, 2008.
- [5] Roberto Muñoz Manso, “Sistema de visión artificial para la detección y lectura de matrículas,” Departamento de Ingeniería de Sistemas y Automática, UNIVERSIDAD DE VALLADOLID, Valladolid- España, 2014.
- [6] i+D3 Equipamientos Tecnológicos SL, “3LPR Sistema LPR reconocimiento de matrículas automático,” *i+D3*, 2018. [Online]. Available: <https://imasdetres.com/sistema-anpr-reconocimiento-automatiko-matriculas/>.
- [7] I. E. M. Carmen López de la Torre, “TUTORIAL III: PRIMEROS PASOS CON RASPBERRY PI.” sociotal.

[8] Universitat Politecnica de Valencia, "RASPBERRY PI," Blog Historia de la Informática, de diciembre del-2013. [Online]. Available: <https://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>.

[9] RASPBERRY PI FOUNDATION, "RASPBERRY PI 3 MODEL B," raspberrypi. [Online]. Available: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>.

[10] Guido van Rossum, "El tutorial de Python." Copyright © Python Software Foundation, Sep-2009.

[11] Miguel Angel Alvarez, "Qué es Python," Desarrollo Web, 19-Nov-2003. [Online]. Available: <https://desarrolloweb.com/articulos/1325.php>.

[13] P. B. B. Dhiraj Y. Gaikwad, "A Review Paper on Automatic Number Plate Recognition (ANPR) System." International Journal of Innovative Research in Advanced Engineering (IJIRAE), Abril-2014.

[14] Campos Albuixech Arnau, "DESARROLLO DE SOFTWARE DE RECONOCIMIENTO DE MATRÍCULAS DE COCHE," Trabajo final de carrera, Universidad Politécnica de Valencia, Gandia, 2014.

[15] D. M. E. Daniel Lelis Baggio, R. S. Naureen Mahmood, K. L. Shervin Emami, and Jason Saragih, Dominar OpenCV con proyectos prácticos de visión artificial. PACKT Open Source.

[16] OpenALPR Technology, "Documentación de OpenALPR," OpenALPR, 2017. [Online]. Available: <http://doc.openalpr.com/>.

[17] Tarik Moustaid, "¿Cuáles son las relaciones entre OpenAlpr, OpenCv, Tesseract?," quora, 15-Jun-2016. [Online]. Available: <https://www.quora.com/What-are-the-relationships-between-OpenAlpr-OpenCv-Tesseract>.

[18] Equipo de OpenCV, "Opencv," Opencv, 2018. [Online]. Available: <https://opencv.org/>.

[19] Equipo de OpenCV, "Acerca de," OpenCv, 2018. [Online]. Available: <https://opencv.org/about.html>.

[20] Fundación Wikimedia, "Tesseract OCR," Wikipedia, 01-Nov-2017. [Online]. Available: https://es.wikipedia.org/wiki/Tesseract_OCR.

[21] Nicolás Vega Beltrán, "Identificación de vehículos de transporte público utilizando Visión por Computador," Tesis, Pontificia Universidad Javeriana, Bogotá - Colombia, 2010.

[22] Kulturaren Euskal Behatokia, "KULTURA 2.0 - Píldoras de formación: OCR: tecnología para el reconocimiento óptico de caracteres en una imagen." abril-2011.

[23] P. R. U. Alba Raquel Paguay, "Recuperación de imágenes mediante extracción de Blobs aplicando el operador Laplaciano de Gauss y el Kernel Gaussiano y desarrollo de un prototipo," Tesis, Politécnica Salesiana, Cuenca, 2012.

[24] Margaret Rouse, "Aprendizaje automático (machine learning)," SearchDataCenter en Español, enero-2017. [Online]. Available: <https://searchdatacenter.techtarget.com/es/definicion/Aprendizaje-automatico-machine-learning>.

[25] Bea Navarro, "Claves de la Inteligencia Artificial: Machine Learning y Deep Learning.," planeta chat bot, 30-Sep-2017. [Online]. Available: <https://planetachatbot.com/claves-de-inteligencia-artificial-machine-learning-y-deep-learning-53a2032aaad>.

[26] The MathWorks, Inc., “Aprendizaje automático Tres cosas que es necesario saber,” MATLAB, 2018-1994. [Online]. Available: <https://es.mathworks.com/discovery/machine-learning.html>.

[27] Alberto Escudero Pardo, “RECONOCIMIENTO DE OBJETOS,” Trabajo final de carrera, Universidad de Barcelona, Barcelona, 2009.

[28] Miguel Ángel Abellán, “¿Qué es Raspbian?,” *programo ergo sum*. [Online]. Available: <https://www.programoergosum.com/cursos-online/raspberry-pi/232-curso-de-introduccion-a-raspberry-pi/instalar-raspbian>.

[29] Rodríguez Daniela, “Investigación aplicada: características, definición, ejemplos,” *ifeder*. [Online]. Available: <https://www.lifeder.com/investigacion-aplicada/>.

[30] Karel Gomez, “Top 5 Metodologías de Desarrollo de Software,” *Megapractical*, 27-Jul-2017. [Online]. Available: <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>.

[31] “Modelo de Prototipos,” *ecured*, 29-May-2018. [Online]. Available: [https://www.ecured.cu/Modelo de Prototipos](https://www.ecured.cu/Modelo_de_Prototipos).

[32] Google Sites, “Construcción de prototipos,” Metodologías de desarrollo de software. [Online]. Available: <https://sites.google.com/site/metdlgsddesarrollodesoftware/5-construccion-de-prototipos>.

[33] Luis Llamas, “MODELOS DE RASPBERRY PI,” LUIS LLAMAS, 17-Nov-2017. [Online]. Available: <https://www.luisllamas.es/modelos-de-raspberry-pi/>.

[34] “Fuente abierta,” OpenALPR Documentation, 2017. [Online]. Available: <http://doc.openalpr.com/opensource.html#openalpr-design>.

ANEXOS

ANEXO A. Especificación de requisitos de software

Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema de información
VLPR	Sistema de Información Web para identificación de placas vehiculares y control de acceso
ERS	Especificación de Requisitos Software
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional

Tabla 19 Especificación de requisitos de software -Definiciones, acrónimos y abreviaturas, Fuente Autor

Referencias

Título del Documento	Referencia
Standard IEEE 830 – 1998	IEEE

Tabla 20 Especificación de requisito de software- Referencias, Fuente Autor

Resumen

Para la presentación de las especificaciones de los requisitos del software se dividió en dos secciones. En la primera sección se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar y las restricciones que afectan el desarrollo, sin entrar en excesivos detalles.

Por último, se definen detalladamente los requisitos que debe satisfacer el sistema.

DESCRIPCIÓN GENERAL

En esta sección se describen los factores generales que afectan el producto y sus requisitos. Esta sección no declara los requisitos específicos.

Funciones del producto

El sistema de información VLPRA Unisinu será un prototipo implementado en Raspberry que automatizará el proceso de la captura de placas vehiculares para controlar el acceso al parqueadero de la Universidad del Sinú sede santillana, en este sistema se desarrollará una página web que permitirá consultar los accesos de los automotores con su respectiva imagen y datos del ingreso.

Características de los usuarios

Esta subdivisión describe las características generales de los usuarios.

Tipo de usuario	Administrador
Formación	Jefe de vigilancia
Actividades	Control y manejo del sistema en general

Tabla 21 Usuario- Administrador, Fuente Autor

Tipo de usuario	Vigilante
Formación	Bachiller con conocimiento en manejo de sistemas
Actividades	Persona que verifica el ingreso del vehículo al sistemas

Tabla 22 Usuario-Vigilante, Fuente Autor

Restricciones

Esta subdivisión proporciona una descripción general de cualquier otro punto que limitará las opciones de los diseñadores. Éstos incluyen:

- Interfaz para ser usada con internet.
- Lenguajes y tecnologías en uso: python.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema se diseñará según el patrón de diseño Modelo - Vista - Controlador (MVC)
- El sistema deberá tener un diseño e implementación sencilla, independiente de la plataforma o del lenguaje de programación.

ANEXO B . Descripción de los campos de la aplicación

AUTENTICACIÓN DE USUARIO			
Campo*:	Identificación		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Numérico entero
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo en el cual se diligencia el número de identificación del usuario a ingresar.		
Rango de valores permitidos	No aplica		
Observaciones*	La persona debe de estar previamente registrada.		
Campo*:	Contraseña		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Carácter
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para la digitación de la clave del usuario ingresada al momento del registro en la base de datos.		
Rango de valores permitidos	No aplica		
Observaciones*	La persona debe de estar previamente registrada		
Campo*:	Tipo de usuario		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo	Tipo de dato	Texto
Valor por defecto	<ul style="list-style-type: none"> ➤ Administrador ➤ Vigilante 	Tooltip	No aplica
Descripción*	Combo para la selección del tipo de usuario de acuerdo al rol registrado del usuario.		
Rango de valores permitidos	No aplica		
Observaciones*	La persona debe de estar previamente registrada.		

Tabla 23 Iniciar sesión, Fuente Autor

GESTIONAR USUARIOS			
Campo*:	Filtrar		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo por medio del cual se podrá filtrar en la grilla por cualquiera de las columnas que coincida el criterio de búsqueda.		
Rango de valores permitidos	No aplica		
Observaciones*	Muestra los datos de los usuarios registrados en la base de datos		
Campo*:	Botón agregar empleado		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Boton	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción que genera una ventana modal con los campos requeridos para el registro de la información de los empleados y guardarlos en la base de datos.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Botón editar empleado		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Boton	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción que genera una ventana modal. En los campos estarán los datos anteriormente ingresados por el usuario, los que estén activos son los permitidos para su modificación.		
Rango de valores permitidos	No aplica		
Observaciones*	El usuario debe de estar registrado.		
Campo*:	Mostrar registros		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Combo	Tipo de dato	Numérico entero
Valor por defecto	10	Tooltip	No aplica
Descripción*	Campo por medio el cual se podrá editar la cantidad de registros que mostrará la grilla activando la paginación.		

Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Grilla Lista de empleados		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Grid	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	
Descripción*	Grilla que contiene todos los empleados registrados en el sistema, no puede ser editable.		
Rango de valores permitidos	No aplica		
Observaciones*	<p>Esta grilla estará formada de nueve columnas.</p> <p>Columna "Edición": Botón que abre una ventana modal para visualizar la información del empleado y poder modificarla, no se pueden modificar los datos de "contraseña" este no se mostrara por ser dato personal e "identificación" por ser una llave primaria en la base de datos,</p> <p>Columnas "Nombres", "Primer Apellido", "Segundo Apellido", "Dirección", "E-Mail", "Teléfono", la columna "Rol" mostrara el tipo de empleado.</p>		

Tabla 24 Gestionar Usuarios, Fuente Autor

REGISTRAR USUARIOS			
Campo*:	Identificación		
Longitud	11	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Numérico entero
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo para digitar el número de identificación del usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	La identificación corresponde al número de la cedula de ciudadanía o cualquier documento.		
Campo*:	Nombre		
Longitud	100	Obligatorio*	Si

Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar el nombre del usuario a registrar		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Primer apellido		
Longitud	100	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar el primer apellido del usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Segundo apellido		
Longitud	100	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar el segundo apellido del usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Correo		
Longitud	100	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Carácter
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar el correo electrónico del usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	Preferiblemente correo institucional		
Campo*:	Dirección		
Longitud	100	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Carácter
Valor por defecto	No aplica	Tooltip	No aplica

Descripción*	Campo para digitar la dirección de la vivienda adonde vive el usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Teléfono		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Numérico decimal
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar el número telefónico del usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Clave		
Longitud	50	Obligatorio*	Si
Tipo control*	Casilla de verificación (Check)	Tipo de dato	Carácter
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar la clave de ingreso del usuario al sistema.		
Rango de valores permitidos	No aplica		
Observaciones*	La clave debe de ser digitada por el usuario que está registrando.		
Campo*:	Rol del empleado		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo	Tipo de dato	Texto
Valor por defecto	<ul style="list-style-type: none"> • Administrador • Vigilante 	Tooltip	No aplica
Descripción*	Campo para digitar el primer apellido del usuario a registrar.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Tabla 25 Registrar Usuario, Fuente Autor

GESTIONAR CÁMARAS			
Campo*:	Filtrar		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo por medio del cual se podrá filtrar en la grilla por cualquiera de las columnas que coincida al criterio de búsqueda.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Botón agregar cámara IP		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Boton	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción que genera una ventana modal con los campos requeridos para el registro de la información de las cámaras y guardarlos en la base de datos.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Botón actualizar cámara IP		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Boton	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción que genera una ventana modal. En los campos estarán los datos anteriormente ingresados de las cámaras, los que estén activos son los permitidos para su modificación.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Mostrar registros		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Combo	Tipo de dato	Numérico entero
Valor por defecto	10	Tooltip	No aplica
Descripción*	Campo por medio el cual se podrá editar la cantidad de registros que mostrará la grilla activando la paginación.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Campo*:	Grilla Lista de cámaras		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Grid	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Grilla que contiene todas las cámaras registradas en el sistema, no puede ser editable.		
Rango de valores permitidos	No aplica		
Observaciones*	Esta grilla estará formada de nueve columnas. Columna "Opciones": Botón que abre una ventana modal para visualizar la información de la cámara y poder modificarla, no se pueden modificar los datos de "id" por ser una llave primaria en la base de datos, Columnas "Id", "Lugar", "Características".		

Tabla 26 Gestionar Cámaras, Fuente Autor

REGISTRAR CÁMARAS			
Campo*:	Nombre cámara		
Longitud	50	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo para digitar el nombre otorgado a la cámara.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Lugar de vigilancia		
Longitud	50	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar la ubicación en la cual estará la cámara de vigilancia.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Características		
Longitud	500	Obligatorio*	No
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico

Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para digitar las características de la cámara a registrar		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Tabla 27 Registrar Cámaras, Fuente Autor

DETECCIONES EN LÍNEA			
Campo*:	Filtrar		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo por medio del cual se podrá filtrar en la grilla por cualquiera de las columnas que coincida el criterio de búsqueda.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Grilla de detecciones en línea		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Grid	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Grilla que contiene los registros de las detecciones que han realizado las cámaras en lo que ha transcurrido del día, no puede ser editable.		
Rango de valores permitidos	No aplica		
Observaciones*	Esta grilla estará formada de nueve columnas. Columna "Fotografía": Botón que abre una ventana modal para visualizar la foto tomada por la cámara, Columnas "Id", "Fecha y Hora" esta solo mostrara la información del día en vigencia, la Columna "Placa" mostrara el texto extraído de la foto tomada por la cámara al momento de la detección.		

Tabla 28 Detecciones en línea, Fuente Autor

Botón copiar: esta función te permite copiar la información que tiene la grilla.

Botón Print: esta función te permite imprimir la información que tiene la grilla.

Botón PDF: esta función te permite exportar la información que tiene la grilla en un archivo PDF.

Botón EXCEL: esta función te permite exportar la información que tiene la grilla en un archivo EXCEL.

HISTORIAL DE LAS DETECCIONES			
Campo*:	Filtrar		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	vacío	Tooltip	No aplica
Descripción*	Campo por medio del cual se podrá filtrar en la grilla por cualquiera de las columnas que coincida el criterio de búsqueda.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Grilla de detecciones		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Grid	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Grilla que contiene los registros de las detecciones que han realizado las cámaras, no puede ser editable.		
Rango de valores permitidos	No aplica		
Observaciones*	Esta grilla estará formada de nueve columnas. Columna "Fotografía": Botón que abre una ventana modal para visualizar la foto tomada por la cámara, Columnas "Id", "Fecha y Hora", la Columna "Placa" mostrara el texto extraído de la foto tomada por la cámara al momento de la detección.		

Tabla 29 Historial de las Detecciones, Fuente Autor

Botón copiar: esta función te permite copiar la información que tiene la grilla.

Botón Print: esta función te permite imprimir la información que tiene la grilla.

Botón PDF: esta función te permite exportar la información que tiene la grilla en un archivo PDF.

Botón EXCEL: esta función te permite exportar la información que tiene la grilla en un archivo EXCEL.

ANEXO C . Desarrollo página web

Función **JavaScript** utilizada para realizar la validación de inicio de sesión, verifica que los datos ingresados en pantalla se encuentran almacenados en la base de datos por medio de Ajax quien prepara la información para enviarla al servidor. Al momento de que Ajax ejecuta la información enviando los parámetros a clase *logueame.php* del servidor, la función *success* extrae los datos que fueron consultados y validados, si esta información es correcta se re direccionará con la función `$(location).attr('href','index.php')` de lo contrario se genera un mensaje de error.

```
<script>
$(document).ready(function(){
    $('#login').click(function(){
        var user = $('#user').val();
        var pass = $('#pass').val();
        var tipoUser = $('#tipoUser').val();
        if ($.trim(user).length > 0 && $.trim(pass).length > 0) {
            $.ajax({
                url:"logueame.php",
                type:"POST",
                data:{user:user, pass:pass, tipoUser:tipoUser},
                cache:"false",
                beforeSend:function(){
                    $('#login').val("Conectando...Espere un momento");
                },
                success:function(data){
                    $('#login').val("Login");
                    if (data=="1") {
                        $(location).attr('href','index.php');
                    } else {
                        $("#result").html("<div class='alert alert-dismissible alert-danger'><button
type='button' class='close' data-dismiss='alert'>&times;</button><strong>¡Error!</strong>
las credenciales son incorrectas.</div>");
                    }
                }
            });
        }
    });
}
</script>
```

Logueame.php

Clase que contiene la consulta a la base de datos para la verificación de los datos de inicio de sesión, si estos datos concuerdan con los registrados anteriormente, este accede a la página siguiente.

```
<?php
    session_start();
    $connect = mysqli_connect("localhost","root","","bdproyecto");
    if (isset($_POST["user"]) && isset($_POST["pass"])) {
        $user =mysqli_real_escape_string($connect, $_POST["user"]);
        $pass =mysqli_real_escape_string($connect, $_POST["pass"]);
        $tipoUser =mysqli_real_escape_string($connect, $_POST["tipoUser"]);
        $sql = "SELECT id, clave,rol.nomrol, GROUP_CONCAT( CONCAT_WS( ' ', nombres, apellido1,
        apellido2 ) ) nombreCompleto
                FROM usuario,rol
                WHERE id=$user
                AND clave='$pass' AND rol.idrol=rol_idrol AND rol_idrol='$tipoUser'";
        $result = mysqli_query($connect, $sql);
        $num_row = mysqli_num_rows($result);
        if ($num_row == "1") {
            $data = mysqli_fetch_array($result);
            if ($data["nombreCompleto"] != null) {
                # code...
                $_SESSION["user"] = $data["nombreCompleto"];
                $_SESSION["tip"] = $data["nomrol"];
                echo "1";
            }else{
                echo "error";
            }
        } else {
            echo "error";
        }
    } else {
        echo "error";
    }
    ?>
```

Cargar Empleados a Datatables

En la aplicación se ha implementado tablas dinámicas para el manejo de la información de manera rápida utilizando JQuery, JavaScript, JSON y Ajax, estas tablas varían la información en sus filas dependiendo el criterio de búsqueda, se trabaja con un Plugin llamado Datatables.

En la siguiente clase llamada *CargaEmpleados.js* se utiliza la función Datatables quien recibe un JSON con toda la configuración que tendrá la tabla, este JSON

llama otra clase llamada *funcionCargaEmpleados.php* quien contiene la consulta para traer de la base de datos toda la información correspondiente a los empleados registrados; la clase *CargaEmpleados.js* se ejecuta cada vez que se inicia la tabla que se encuentra en la vista del módulo de empleados.

CargaEmpleados.js

```
$(document).ready(function() {
    $('#TablEmpleados').DataTable( {
        "bDeferRender": true,
        "sPaginationType": "full_numbers",
        "ajax": {
            "url": "funcionCargaEmpleados.php",
        },
        "type": "POST"
    },
    "columns": [
        { "data": "identificacion" },
        { "data": "nombres" },
        { "data": "pApellido" },
        { "data": "sApellido" },
        { "data": "direccion" },
        { "data": "email" },
        { "data": "telefono" },
        { "data": "rol" },
        { "data": "Editar" }
    ],
    "oLanguage": {
        "sProcessing": "Procesando...",
        "sLengthMenu": 'Mostrar <select>'+
            '<option value="2">2</option>'+
            '<option value="5">5</option>'+
            '<option value="10">10</option>'+
            '<option value="20">20</option>'+
            '<option value="-1">All</option>'+
            '</select> registros',
        "sZeroRecords": "No se encontraron resultados",
        "sEmptyTable": "Ningún dato disponible en esta tabla",
        "sInfo": "Mostrando del (_START_ al _END_) de un total de _TOTAL_ registros",
        "sInfoEmpty": "Mostrando del 0 al 0 de un total de 0 registros",
        "sInfoFiltered": "(filtrado de un total de _MAX_ registros)",
        "sInfoPostFix": "",
        "sSearch": "Filtrar:",
        "sUrl": "",
        "sInfoThousands": ",",
        "sLoadingRecords": "Por favor espere - cargando...",
        "oPagate": {
            "sFirst": "Primero",
            "sLast": "Último",
            "sNext": "Siguiete",
            "sPrevious": "Anterior"
        },
    },
    "oAria": {
        "sSortAscending": ": Activar para ordenar la columna de manera ascendente",
        "sSortDescending": ": Activar para ordenar la columna de manera descendente"
    }
});
```

Los datos extraídos de la consulta se transmiten por medio de variables data que contienen la información de cada celda consultada.

FuncionCargaEmpleados.php

Esta clase es la encargada de traer la información de la base de datos a la aplicación, al realizar la consulta creamos un JSON llamado tabla, esta variable es la contenedora de todos los registros consultados, que serán enviados a una tabla determinada en el sistema. Además de mostrar los datos en las tablas se adiciona una columna para las acciones, esta columna contiene un botón que carga un modal con toda la información de la fila para ser editados; ese botón contiene variables data para identificar cada índice de los valores a mostrar en el modal y se

```
<?php
//cadena
de
conexion

    $mysqli = new mysqli("localhost", "root", "", "bdproyecto");
//validacion de conexion a la base de datos
    if ($mysqli->connect_errno) {
        echo "Fallo al conectar a MySQL: (" . $mysqli->connect_errno . ") " . $mysqli->connect_error;
        exit();
    }
    $consulta="SELECT u.id, u.nombres, u.apellido1, u.apellido2, u.direccion, u.email,
u.telefono, r.nomrol
                FROM usuario u, rol r
                WHERE u.rol_idrol = r.idrol";
    $tabla = "";
    if ($resultado = $mysqli->query($consulta)) {
        //while ($fila = $resultado->fetch_row()) {
        while ($fila = $resultado->fetch_array()){
            $Edicion = '<a class="btn btn-primary btn-sm" style="color:white;" data-
target="#modalEdicion" data-toggle="modal" '.
            'data-edidad="\'.$fila['id'].'\" '.
            'data-ednombread="\'.$fila['nombres'].'\" '.
            'data-edapellido1ad="\'.$fila['apellido1'].'\" '.
            'data-edapellido2ad="\'.$fila['apellido2'].'\" '.
            'data-edcorad="\'.$fila['email'].'\" '.
            'data-eddirad="\'.$fila['direccion'].'\" '.
            'data-edtelad="\'.$fila['telefono'].'\" '.
            '>Editar</a>';
            $tabla.='{'
                "identificacion": "\'.$fila['id'].'",
                "nombres": "\'.$fila['nombres'].'",
                "pApellido": "\'.$fila['apellido1'].'",
                "sApellido": "\'.$fila['apellido2'].'",
                "direccion": "\'.$fila['direccion'].'",
                "email": "\'.$fila['email'].'",
                "telefono": "\'.$fila['telefono'].'",
                "rol": "\'.$fila['nomrol'].'",
                "Editar": "\'.$Edicion.'"
            },';
        }
    }
//Elinamos la coma que sobra
    $tabla = substr($tabla,0, strlen($tabla) - 1);
    echo '{"data":['.$tabla.']}';
?>
```

carga en la variable tabla como una variable de texto llamada Edicion.

De esta manera se realizan las consultas de la información en la base de datos y son enviadas a las tablas correspondientes de la misma manera para los diferentes módulos que existen en el sistema.

Registrar y Actualizar

Actualizar

Para realizar esta acción es necesario que la tabla contenedora de la información tenga todos los datos en vista y que el botón contenedor de la variable data tenga la información de la fila. Los datos que serán actualizados serán mostrados a través de una ventana modal y se cargan a la ventana de la siguiente manera:

```
<script type="text/javascript">

    debugger;
    var button = $(event.relatedTarget)
    // Button that triggered the modal
    var recipient0 = button.data('edidad')
    var recipient1 = button.data('ednombread')
    var recipient2 = button.data('edapellido1ad')
    var recipient3 = button.data('edapellido2ad')
    var recipient4 = button.data('edcorad')
    var recipient5 = button.data('eddirad')
    var recipient6 = button.data('edtelad')

    // Extract info from data-* attributes
    // If necessary, you could initiate an AJAX request here (and then do the updating in a
    // callback).
    // Update the modal's content. We'll use jQuery here, but you could use a data binding
    // library or other methods instead.

    var modal = $(this)
    modal.find('.modal-body #medidad').val(recipient0)
    modal.find('.modal-body #mednombread').val(recipient1)
    modal.find('.modal-body #medapellido1ad').val(recipient2)
    modal.find('.modal-body #medapellido2ad').val(recipient3)
    modal.find('.modal-body #medcorad').val(recipient4)
    modal.find('.modal-body #meddirad').val(recipient5)
    modal.find('.modal-body #medtelad').val(recipient6)
});
</script>
```

Ya con la información en el modal se puede editar solo los campos habilitados, se edita al momento de presionar el botón “actualizar” se hace llamado al siguiente script, quien se encarga de preparar los datos para enviarlos a la clase *actualizaEmp.php*.

```

$('#edEmp').click(function(){
    var idAdm = $('#medidad').val();
    var nomAdm = $('#mednombread').val();
    var apeAdm = $('#medapellido1ad').val();
    var ape2Adm = $('#medapellido2ad').val();
    var dirAdm = $('#meddirad').val();
    var corAdm = $('#medcorad').val();
    var telAdm = $('#medtelad').val();
    $.ajax({
        url:"actualizaEmp.php",
        type:"POST",
        data:{didAdm:idAdm,
            dnomAdm:nomAdm,
            dapeAdm:apeAdm,
            dape2Adm:ape2Adm,
            ddirAdm:dirAdm,
            dcorAdm:corAdm,
            dtelAdm:telAdm},
        cache:"false",
        beforeSend:function() {
            $('#edEmp').val("Por Favor Espere...");
        },
        success:function(data) {
            $('#edEmp').val("Actualizar");
            if (data.trim()=="actualizado") {
                Mensaje("Notificar","Actualizacion completa...!!");
                document.location.reload()
            } else if (data.trim()=="Error"){
                Mensaje("Alerta","No Se pudo Realizar la actualizacion!! :( ");
            }
        }
    });
});

```

ActualizaEmp.php

Esta clase es la que se comunica con la base de datos para realizar las actualizaciones de la información.

```

<?php
session_start();
    $mysqli = new mysqli("localhost", "root", "", "bdproyecto");

    if (isset($_POST["didAdm"])
        && isset($_POST["dnomAdm"])
        && isset($_POST["dapeAdm"])
        && isset($_POST["dape2Adm"])
        && isset($_POST["ddirAdm"])
        && isset($_POST["dcorAdm"])
        && isset($_POST["dtelAdm"])) {
        $vId = mysqli_real_escape_string($mysqli, $_POST["didAdm"]);
        $vNombre = mysqli_real_escape_string($mysqli, $_POST["dnomAdm"]);
        $vApellido = mysqli_real_escape_string($mysqli, $_POST["dapeAdm"]);
        $vApellido2 = mysqli_real_escape_string($mysqli, $_POST["dape2Adm"]);
        $vDireccion = mysqli_real_escape_string($mysqli, $_POST["ddirAdm"]);
        $vCorreo = mysqli_real_escape_string($mysqli, $_POST["dcorAdm"]);
        $vTelefono = mysqli_real_escape_string($mysqli, $_POST["dtelAdm"]);

        $sql = "UPDATE usuario
                SET
nombres='$vNombre', apellido1='$vApellido', apellido2='$vApellido2', email='$vCorreo', telefono=$
vTelefono,direccion='$vDireccion'
                WHERE id=$vId";
        if(mysqli_query($mysqli,$sql)){
            echo"actualizado";
        }else{
            echo"Error" . mysql_error($mysqli);
        }
        mysqli_close($mysqli);
    }
?>

```

Esta clase verifica que los datos transiten por el método `$_POST`, estos son organizados en la cadena `$SQL` y ejecutados por `(mysqli_query($mysqli,$sql))` para actualizar la información.

Registrar

En algunos módulos tendremos la opción de agregar registros, esta opción tendrá lugar por medio un botón que generara una ventana modal con todos los campos disponibles para insertarlo en la base de datos, el modal es quien llamara al siguiente script para empezar el proceso de agregar registro:

```

$('#AddEmp').click(function(){
    var idAdm = $('#maddidAd').val();
    var nomAdm = $('#maddnombreAd').val();
    var ape1Adm = $('#maddapellido1Ad').val();
    var ape2Adm = $('#maddapellido2Ad').val();
    var corAdm = $('#maddcorAd').val();
    var dirAdm = $('#madddirAd').val();
    var telAdm = $('#maddtelAd').val();
    var claAdm = $('#maddClaAd').val();
    var TRolad = $('#maddTRolad').val();
    debugger;
    $.ajax({
        url:"AddEmpleado.php",
        type:"POST",
        data:{didAdm:idAdm,
            dnomAdm:nomAdm,
            dape1Adm:ape1Adm,
            dape2Adm:ape2Adm,
            dcorAdm:corAdm,
            ddirAdm:dirAdm,
            dtelAdm:telAdm,
            dclaAdm:claAdm,
            dTRolad:TRolad},
        cache:"false",
        beforeSend:function() {
            $('#AddEmp').val("Por Favor Espere...");
        },
        success:function(data) {
            $('#AddEmp').val("Registrar");
            if (data.trim()=="registrado") {
                Mensaje("Notificar","Registro ingresado Exitosamente...!!");
                document.location.reload()
            } else if (data.trim()=="Error"){
                Mensaje("Alerta","Error!!... Consulte al Administrador del sistema");
            } else if (data.trim()=="encontrado"){
                Mensaje("Alerta","identificacion ya se encuentra registrado ");
            }
        }
    });
});

```

Este script prepara los datos tomados del modal para enviarlos a la clase *AddEmpleado.php*, espera la respuesta para informar si el registro fue exitoso o si ocurre un error, todo esto ocurre en la función success.

AddEmpleado.php

Esta clase también verifica que la información transite por medio del método `$_POST`, esta ejecuta la secuencia SQL para insertar el nuevo registro a la base de datos, pero antes de ello verifica que el registro no exista y controlar la excepción de primary keys generada en la base de datos por duplicidad.


```

<?php
    $mysqli = new mysqli("localhost", "root", "", "bdproyecto");

    if (isset($_POST["didAdm"])
        && isset($_POST["dnomAdm"])
        && isset($_POST["dape1Adm"])
        && isset($_POST["dape2Adm"])
        && isset($_POST["dcorAdm"])
        && isset($_POST["ddirAdm"])
        && isset($_POST["dtelAdm"])
        && isset($_POST["dclaAdm"])
        && isset($_POST["dTrolad"])) {

        $id = mysqli_real_escape_string($mysqli, $_POST["didAdm"]);
        $nom = mysqli_real_escape_string($mysqli, $_POST["dnomAdm"]);
        $ape1 = mysqli_real_escape_string($mysqli, $_POST["dape1Adm"]);
        $ape2 = mysqli_real_escape_string($mysqli, $_POST["dape2Adm"]);
        $corr = mysqli_real_escape_string($mysqli, $_POST["dcorAdm"]);
        $dir = mysqli_real_escape_string($mysqli, $_POST["ddirAdm"]);
        $tel = mysqli_real_escape_string($mysqli, $_POST["dtelAdm"]);
        $cla = mysqli_real_escape_string($mysqli, $_POST["dclaAdm"]);
        $Trol = mysqli_real_escape_string($mysqli, $_POST["dTrolad"]);

        $sql = "SELECT id, nombres, apellido1, apellido2
                FROM usuario
                WHERE id=$id";

        $result = mysqli_query($mysqli, $sql);
        $num_row = mysqli_num_rows($result);

        if ($num_row == "1") {
            echo "encontrado";
        }else{

            $sql = "INSERT INTO usuario (id, nombres, apellido1, apellido2, telefono,
            direccion, email, rol_idrol, clave, estadoemp)
            VALUES ($id, '$nom', '$ape1', '$ape2', $tel, '$dir', '$corr', $Trol,
            '$cla', 'ACTIVO')";

            if(mysqli_query($mysqli,$sql)){
                echo"registrado";
            }else{
                echo"Error";
            }
        }

        mysqli_close($mysqli);
    } else{
        echo "algun dato no esta publicado ";
    }

?>

```

En el módulo de detecciones en línea se observara que la tabla se actualiza automáticamente, esto sucede debido a la clase *CargarDetecciones.js* quien contiene la función *SetInterval()* refrescándose cada 2000 milisegundos de la siguiente manera

```

$(document).ready(function() {
    CrearModal();
    setInterval(CrearModal, 2000)
});
var datatable = null;
function CrearModal(){

    if(datatable != null){
        datatable.ajax.reload();
    }else{
        datatable = $('#TablDeteccion').DataTable( {
            "bDeferRender": true,
            "sPaginationType": "full_numbers",
            "ajax": {
                "url": "funcionCargaDeteccion.php",
            },
            "type": "POST",
            "columns": [
                { "data": "iddet" },
                { "data": "fechrdet" },
                { "data": "placadet" },
                { "data": "camaradet" },
                { "data": "fotodet" }
            ],
            "oLanguage": {
                "sProcessing": "Procesando...",
                "sLengthMenu": 'Mostrar <select>'+
                    '<option value="2">2</option>'+
                    '<option value="5">5</option>'+
                    '<option value="10">10</option>'+
                    '<option value="20">20</option>'+
                    '<option value="-1">All</option>'+
                    '</select> registros',
                "sZeroRecords": "No se encontraron resultados",
                "sEmptyTable": "Ningún dato disponible en esta tabla",
                "sInfo": "Mostrando del (_START_ al _END_) de un total de _TOTAL_ registros",
                "sInfoEmpty": "Mostrando del 0 al 0 de un total de 0 registros",
                "sInfoFiltered": "(filtrado de un total de _MAX_ registros)",
                "sInfoPostFix": "",
                "sSearch": "Filtrar:",
                "sUrl": "",
                "sInfoThousands": ",",
                "sLoadingRecords": "Por favor espere - cargando...",
                "oPaginate": {
                    "sFirst": "Primero",
                    "sLast": "Último",
                    "sNext": "Siguiente",
                    "sPrevious": "Anterior"
                },
                "oAria": {
                    "sSortAscending": ": Activar para ordenar la columna de manera ascendente",
                    "sSortDescending": ": Activar para ordenar la columna de manera descendente"
                }
            }
        },
        {
            dom: "Bfrtip",
            buttons: [
                {
                    extend: 'copyHtml5',
                    text: '<i class="fa fa-files-o"></i>',
                    titleAttr: 'Copy'
                },
                {
                    extend: 'excelHtml5',
                    text: '<i class="fa fa-file-excel-o"></i>',
                    titleAttr: 'Excel'
                },
                {
                    extend: 'csvHtml5',
                    text: '<i class="fa fa-file-text-o"></i>',
                    titleAttr: 'CSV'
                },
                {
                    extend: 'pdfHtml5',
                    text: '<i class="fa fa-file-pdf-o"></i>',
                    titleAttr: 'PDF'
                }
            ]
        }
    );
}
}

```

Este código verifica el estado del *Datatables* de esa ventana, en el caso de que su valor sea diferente a null solamente recargara la tabla y de lo contrario si la tabla está vacía, cargará una nueva tabla con el contenido consultado en la base de datos.

ANEXO D. Creación de las tablas en la base de datos

```
-- phpMyAdmin SQL Dump
-- version 3.5.1
-- http://www.phpmyadmin.net
--
-- Servidor: localhost
-- Versi3n del servidor: 5.5.24-log
-- Versi3n de PHP: 5.4.3

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `bdproyecto`
--
-----

--
-- Estructura de tabla para la tabla `acceso`
--

CREATE TABLE IF NOT EXISTS `acceso` (
  `idacceso` int(11) NOT NULL,
  `fec_hr_entrada` datetime DEFAULT NULL,
  `fotoalmacen` varchar(3000) DEFAULT NULL,
  `observaciones` varchar(500) DEFAULT NULL,
  `camara_idcam` varchar(30) NOT NULL,
  `placa` varchar(10) DEFAULT NULL,
  PRIMARY KEY (`idacceso`),
  KEY `camara_idcam` (`camara_idcam`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-----

--
-- Estructura de tabla para la tabla `camara`
--
```

```

CREATE TABLE IF NOT EXISTS `camara` (
  `idcam` varchar(30) NOT NULL,
  `zona` varchar(50) DEFAULT NULL,
  `caracteristicas` varchar(500) DEFAULT NULL,
  PRIMARY KEY (`idcam`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Estructura de tabla para la tabla `rol`
--

```

```

CREATE TABLE IF NOT EXISTS `rol` (
  `idrol` int(11) NOT NULL,
  `nomrol` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`idrol`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Estructura de tabla para la tabla `usuario`
--

```

```

CREATE TABLE IF NOT EXISTS `usuario` (
  `id` int(11) NOT NULL,
  `nombres` varchar(100) DEFAULT NULL,
  `apellido1` varchar(100) DEFAULT NULL,
  `apellido2` varchar(100) DEFAULT NULL,
  `telefono` double DEFAULT NULL,
  `direccion` varchar(100) DEFAULT NULL,
  `email` varchar(100) DEFAULT NULL,
  `rol_idrol` int(11) NOT NULL,
  `clave` varchar(50) DEFAULT NULL,
  `estadoemp` varchar(40) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `usuario_rol_fk` (`rol_idrol`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

--
-- Restricciones para tablas volcadas
--

```

```

--
-- Filtros para la tabla `acceso`

```

```
--  
ALTER TABLE `acceso`  
  ADD CONSTRAINT `acceso_ibfk_1` FOREIGN KEY (`camara_idcam`)  
REFERENCES `camara` (`idcam`);  
  
--  
-- Filtros para la tabla `usuario`  
--  
ALTER TABLE `usuario`  
  ADD CONSTRAINT `usuario_rol_fk` FOREIGN KEY (`rol_idrol`)  
REFERENCES `rol` (`idrol`);
```

ANEXO E. Características de la cámara

CÁMARA SEGURIDAD IP 4 MP ANTIVANDALICA HIKVISION HD DS-2CD1141-I

Marca: HikVision

Modelo: DS-2CD1141-I

Cámara domo de red IP resistente a vandalismo de 4 MegaPíxeles

- MegaPíxeles de alta resolución (2688 × 1520)
- Monitoreo desde su celular a través de la aplicación Hik-Connect
- Opciones de lente fija de 2,8 mm, 4 mm y 6 mm
- Reducción de ruido digital 3D (3D-DNR)
- Amplio rango dinámico (WDR)
- Funciones inteligentes
- PoE (802.3af)
- Alcance IR 100 pies (30 m)
- Protección IP67 e IK10

ANEXO F. Pruebas a la aplicación web

Pruebas de interfaz gráfica para requerimientos Web

MODULO REPORTES –DETECCIONES EN LÍNEA			
<i>Escala de calificación:</i>			
Conforme	No Conforme	Parcialmente	No aplica
<i>Sección</i>	<i>Calificación</i>	<i>Observación</i>	
1. Interfaz General o Ventana Nueva			
Se visualiza los controles como activos cuando esto es necesario	No aplica		
En la barra de título aparece el nombre representativo de la interfaz	Conforme	Aparece la parte superior de la interfaz.	
Primera letra en mayúsculas y las demás en minúscula	Conforme		
3. Cuadros de Texto			
¿Los cuadros de texto están alineados correctamente?	Conforme	Solo existe un cuadro de texto	
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	No aplica		
El ancho es correspondiente con la información que digitará el usuario.	Conforme		
Las cajas de texto ubican al lado derecho de las etiquetas.	Parcialmente		
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme		
4. Campos de Fecha			
El ancho del campo deber ser correspondiente con la información que se va a ingresar.	Conforme		
Se ubica al lado derecho de las etiquetas	Conforme	La etiqueta se ubica en la parte superior de la tabla.	
El formato manejado es de dd/mm/yyyy	Conforme	El formato es yyyy-mm-dd hh:mm:ss	
5. Etiquetas			
¿Al final de estas ahí dos puntos?	Conforme		

¿Las etiquetas están alineadas correctamente?	Conforme	
Las etiquetas están escritas con buena ortografía	Conforme	
El color de las etiquetas es legible	Conforme	
6. Combos		
La longitud del combo corresponde a la información a mostrar	No aplica	
Se ubican al lado derecho de las etiquetas	No aplica	
Los combos están alineados correctamente	No aplica	
7. Botones		
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme	Los botones no contienen el nombre pero si una imagen de referencia al archivo que genera y un Tooltip con el nombre
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme	
Si el botón tiene que abrir una nueva ventana lo hace correctamente	No aplica	
8. Grillas		
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme	
La repartición de las celdas que están dentro de la grilla está bien distribuida.	Parcialmente	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	Existe un solo botón con una imagen que se muestra en una ventana emergente
9. Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	No aplica	
Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	No aplica	
¿Se utilizan para alertar al usuario?	No aplica	
Se utilizan para brindar información cuando haya ocurrido un error	No aplica	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	No aplica	
¿Los mensajes desplegados están escritos con buena ortografía?	No aplica	

Tabla 30 Prueba- Detecciones en línea, Fuente Autor

MODULO REPORTES – HISTORIAL DE DETECCIONES			
<i>Escala de calificación:</i>			
Conforme	No Conforme	Parcialmente	No aplica
Sección	Calificación	Observación	
1. Interfaz General o Ventana Nueva			
Se visualiza los controles como activos cuando esto es necesario	No aplica		
En la barra de título aparece el nombre representativo de la interfaz	Conforme	Aparece la parte superior de la interfaz.	
Primera letra en mayúsculas y las demás en minúscula	Conforme		
3. Cuadros de Texto			
¿Los cuadros de texto están alineados correctamente?	Conforme	Solo existe un cuadro de texto	
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	No aplica		
El ancho es correspondiente con la información que digitará el usuario.	Conforme		
Las cajas de texto ubican al lado derecho de las etiquetas.	Parcialmente		
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme		
4. Campos de Fecha			
El ancho del campo deber ser correspondiente con la información que se va a ingresar.	Conforme		
Se ubica al lado derecho de las etiquetas	Conforme	La etiqueta se ubica en la parte superior de la tabla.	
El formato manejado es de dd/mm/yyyy	Conforme	El formato es yyyy-mm-dd hh:mm:ss	
5. Etiquetas			
¿Al final de estas ahí dos puntos?	Conforme		
¿Las etiquetas están alineadas correctamente?	Conforme		
Las etiquetas están escritas con buena ortografía	Conforme		
El color de las etiquetas es legible	Conforme		
6. Combos			

La longitud del combo corresponde a la información a mostrar	No aplica	
Se ubican al lado derecho de las etiquetas	No aplica	
Los combos están alineados correctamente	No aplica	
7. Botones		
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme	Los botones no contienen el nombre pero si una imagen de referencia al archivo que genera y un Tooltip con el nombre
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme	
Si el botón tiene que abrir una nueva ventana lo hace correctamente	No aplica	
8. Grillas		
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme	
La repartición de las celdas que están dentro de la grilla está bien distribuida.	Parcialmente	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	Existe un solo botón con una imagen que se muestra en una ventana emergente
9. Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	No aplica	
Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	No aplica	
¿Se utilizan para alertar al usuario?	No aplica	
Se utilizan para brindar información cuando haya ocurrido un error	No aplica	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	No aplica	
¿Los mensajes desplegados están escritos con buena ortografía?	No aplica	

Tabla 31 Prueba-Historial detecciones, Fuente Autor

MODULO EMPLEADOS – GESTIONAR EMPLEADOS			
<i>Escala de calificación:</i>			
Conforme	No Conforme	Parcialmente	No aplica
Sección	Calificación	Observación	
1. Interfaz General o Ventana Nueva			
Se visualiza los controles como activos cuando esto es necesario	No aplica		
En la barra de título aparece el nombre representativo de la interfaz	Conforme	Aparece la parte superior de la interfaz.	
Primera letra en mayúsculas y las demás en minúscula	Conforme		
3. Cuadros de Texto			
¿Los cuadros de texto están alineados correctamente?	Conforme	Solo existe un cuadro de texto	
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	No aplica		
El ancho es correspondiente con la información que digitará el usuario.	Conforme		
Las cajas de texto ubican al lado derecho de las etiquetas.	Parcialmente		
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme		
4. Campos de Fecha			
El ancho del campo deber ser correspondiente con la información que se va a ingresar.	Conforme		
Se ubica al lado derecho de las etiquetas	Conforme	La etiqueta se ubica en la parte superior de la tabla.	
El formato manejado es de dd/mm/yyyy	No aplica		
5. Etiquetas			
¿Al final de estas ahí dos puntos?	Conforme		
¿Las etiquetas están alineadas correctamente?	Conforme		
Las etiquetas están escritas con buena ortografía	Conforme		
El color de las etiquetas es legible	Conforme		
6. Combos			
La longitud del combo corresponde a la información a mostrar	No aplica		
Se ubican al lado derecho de las etiquetas	No aplica		

Los combos están alineados correctamente	No aplica	
7. Botones		
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme	
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme	
Si el botón tiene que abrir una nueva ventana lo hace correctamente	Conforme	Los botones “agregar Empleado” y “editar” generan ventanas emergentes con formularios organizados
8. Grillas		
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme	
La repartición de las celdas que están dentro de la grilla está bien distribuida.	Parcialmente	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	Existe un solo botón que muestra en una ventana emergente con la información de la grilla para ser editada
9. Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	No aplica	
Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	No aplica	
¿Se utilizan para alertar al usuario?	No aplica	
Se utilizan para brindar información cuando haya ocurrido un error	No aplica	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	No aplica	
¿Los mensajes desplegados están escritos con buena ortografía?	No aplica	

Tabla 32 Prueba-Gestionar empleados, Fuente Autor

MODULO CÁMARAS – GESTIONAR CÁMARAS			
<i>Escala de calificación:</i>			
Conforme	No Conforme	Parcialmente	No aplica
Sección	Calificación	Observación	
1. Interfaz General o Ventana Nueva			
Se visualiza los controles como activos cuando esto es necesario	No aplica		
En la barra de título aparece el nombre representativo de la interfaz	Conforme	Aparece la parte superior de la interfaz.	
Primera letra en mayúsculas y las demás en minúscula	Conforme		
3. Cuadros de Texto			
¿Los cuadros de texto están alineados correctamente?	Conforme	Solo existe un cuadro de texto	
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	No aplica		
El ancho es correspondiente con la información que digitará el usuario.	Conforme		
Las cajas de texto ubican al lado derecho de las etiquetas.	Parcialmente		
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme		
4. Campos de Fecha			
El ancho del campo deber ser correspondiente con la información que se va a ingresar.	Conforme		
Se ubica al lado derecho de las etiquetas	Conforme	La etiqueta se ubica en la parte superior de la tabla.	
El formato manejado es de dd/mm/yyyy	No aplica		
5. Etiquetas			
¿Al final de estas ahí dos puntos?	Conforme		
¿Las etiquetas están alineadas correctamente?	Conforme		
Las etiquetas están escritas con buena ortografía	Conforme		
El color de las etiquetas es legible	Conforme		
6. Combos			
La longitud del combo corresponde a la información a mostrar	No aplica		
Se ubican al lado derecho de las etiquetas	No aplica		
Los combos están alineados correctamente	No aplica		

7. Botones		
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme	
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme	
Si el botón tiene que abrir una nueva ventana lo hace correctamente	Conforme	Los botones “agregar cámara ip” y “editar” generan ventanas emergentes con formularios organizados
8. Grillas		
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme	
La repartición de las celdas que están dentro de la grilla está bien distribuida.	Parcialmente	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	Existe un solo botón que muestra en una ventana emergente con la información de la grilla para ser editada
9. Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	No aplica	
Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	No aplica	
¿Se utilizan para alertar al usuario?	No aplica	
Se utilizan para brindar información cuando haya ocurrido un error	No aplica	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	No aplica	
¿Los mensajes desplegados están escritos con buena ortografía?	No aplica	

Tabla 33 Prueba- Gestionar empleados, Fuente Autor

Pruebas de funcionalidad para los requerimientos

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF01 – AUTENTICACIÓN DE USUARIO</i>		
<i>Escala de Calificación</i>		
Conforme	No Conforme, Parcialmente, No Aplica	
<i>Ítems a Revisar</i>	<i>Calificación</i>	<i>Observación</i>
Los botones funcionan adecuadamente.	Conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	No aplica	Solamente permite consultas
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	Conforme	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Conforme	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	No aplica	

Tabla 34 Prueba- Autenticación de usuario, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF02 – GESTIONAR USUARIO</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	Conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	Conforme	
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	Conforme	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Parcialmente	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Parcialmente	

Tabla 35 Prueba- Gestionar usuario, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF03 – REGISTRAR USUARIO</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	Conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza registros
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	Conforme	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Parcialmente	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Parcialmente	

Tabla 36 Prueba- Registrar usuario, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF04 – CONSULTAR USUARIO</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	No aplica	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	No aplica	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza consulta que se cargan en un grid
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	No Aplica	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No aplica	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Conforme	

Tabla 37 Prueba - Consultar usuario, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF05 – MODIFICAR USUARIO</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	Conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	Parcialmente	
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Parcialmente	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	Parcialmente	
Se muestra un mensaje cuando se ha cumplido una operación.	Parcialmente	

Tabla 38 Prueba- Modificar usuario, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF06 – GESTIONAR CÁMARA</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	Conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	Conforme	
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	Conforme	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Parcialmente	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Parcialmente	

Tabla 39 Prueba- Gestionar cámara, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF07 – CONSULTAR CÁMARA</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	No aplica	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	No aplica	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza consulta que se cargan en un grid
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No Aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	No Aplica	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No aplica	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Conforme	

Tabla 40 Prueba- Consultar cámara, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF08 – MODIFICAR CÁMARA</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	Parcialmente	
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Parcialmente	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	Parcialmente	
Se muestra un mensaje cuando se ha cumplido una operación.	Parcialmente	

Tabla 41 Prueba- Modificar cámara, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF09 – REGISTRAR CÁMARA</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	Conforme	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	Conforme	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza registros
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	Conforme	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	Conforme	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	Parcialmente	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Parcialmente	

Tabla 42 Prueba - Registrar cámara, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF10 – CONSULTAR INFORME DETECCIONES</i>		
Escala de Calificación		
Conforme	No Conforme, Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	No aplica	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	No aplica	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza consulta que se cargan en un grid
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No Aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	No Aplica	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No aplica	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	Conforme	

Tabla 43 Prueba - Consultar informe detecciones, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
RF12 – REGISTRAR ENTRADAS		
<i>Escala de Calificación</i>		
Conforme	No Conforme, Parcialmente, No Aplica	
<i>Ítems a Revisar</i>	<i>Calificación</i>	<i>Observación</i>
Los botones funcionan adecuadamente.	No aplica	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	No aplica	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza registros
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	No aplica	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No aplica	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	No aplica	

Tabla 44 Prueba- Registrar entradas, Fuente Autor

PRUEBAS DE FUNCIONALIDAD PARA LOS REQUERIMIENTOS		
<i>RF13 – CAPTURA AUTOMÁTICA DE LA PLACA</i>		
Escala de Calificación		
Conforme	No Conforme Parcialmente, No Aplica	
Ítems a Revisar	Calificación	Observación
Los botones funcionan adecuadamente.	No aplica	
Los cuadros de Texto permiten ingresar solo el tipo de valor que está estipulado en el documento en donde se explica el requerimiento.	No aplica	
Permite realizar el CRUD Correctamente.	Parcialmente	Solamente realiza registros
Las validaciones descritas en la documentación del requerimiento se están cumpliendo correctamente.	Conforme	
Lo probado cumple con los requerimientos funcionales que están descritos en la documentación del requerimiento.	Conforme	
Los combos cargan la información adecuada según el requerimiento.	No aplica	
Los Check-Box cumplen con la función que les corresponde según la documentación del requerimiento	No Aplica	
Si los datos son de carácter obligatorio, se despliegan un mensaje de advertencia si el usuario no los ha digitado.	No aplica	
Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No aplica	
El requerimiento permite visualizar la ventana o el formulario con los datos precargados.	No aplica	
Se muestra un mensaje cuando se ha cumplido una operación.	No aplica	

Tabla 45 Prueba- Captura automática de la placa, Fuente Autor