

REGISTRO Y CONTROL DE CILINDROS GLP UTILIZANDO PROCESAMIENTO DE IMÁGENES TOMANDO COMO HERRAMIENTA UN DISPOSITIVO MÓVIL

**JORGE ELIECER CASTAÑEDA AGUALIMPIA
CARLOS DANIEL ALVAREZ RUIZ**

**FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA ELECTRÓNICA
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA
2016**

**REGISTRO Y CONTROL DE CILINDROS GLP UTILIZANDO PROCESAMIENTO DE
IMÁGENES TOMANDO COMO HERRAMIENTA UN DISPOSITIVO MÓVIL**

JORGE ELIECER CASTAÑEDA AGUALIMPIA
CARLOS DANIEL ALVAREZ RUIZ

Trabajo de grado presentado como requisito parcial
para aspirar al título de Ingeniero Electrónico

Director
Ing. Julian David Echeverry

FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA ELECTRÓNICA
UNIVERSIDAD TECNOLÓGICA DE PEREIRA
PEREIRA
2016

CONTENIDO

| | PÁG. |
|---|-----------|
| 1. INTRODUCCIÓN..... | 5 |
| 2. PLANTEAMIENTO DEL PROBLEMA..... | 6 |
| 3. MOTIVACIÓN..... | 8 |
| 4. OBJETIVOS..... | 9 |
| 4.1. Objetivo General..... | 9 |
| 4.2. Objetivos Específicos..... | 9 |
| 5. MARCO DE REFERENCIA..... | 10 |
| 5.1. Marco Histórico..... | 10 |
| 5.2. Marco Teórico..... | 12 |
| 6. METODOLOGÍA..... | 19 |
| 6.1. Etapa 1..... | 19 |
| 6.2. Etapa 2..... | 19 |
| 6.3. Etapa 3..... | 20 |
| 6.4. Etapa 4..... | 20 |
| 7. DESARROLLO..... | 21 |
| 7.1. Diagrama General de las Áreas de Desarrollo..... | 21 |
| 7.1.1. Descripción General del Modo de Operar las Áreas de Desarrollo..... | 22 |
| 7.2. Diagrama Detallado del Funcionamiento del Sistema..... | 23 |
| 7.2.1. Descripción Detallada del Funcionamiento del Sistema..... | 24 |
| 7.3. Diagrama del Funcionamiento de la App Móvil..... | 28 |
| 7.3.1. Descripción Detallada del Funcionamiento de la App Móvil.... | 29 |
| 7.4. Diagrama del Funcionamiento del Procesamiento de Imágenes..... | 39 |

| | |
|---|-----------|
| 7.4.1. Descripción Detallada del Funcionamiento del Procesamiento de Imágenes..... | 39 |
| 7.5 Descripción Funcionamiento de la Base de Datos..... | 49 |
| 8. RESULTADOS..... | 51 |
| 8.1. Sistema Obtenido..... | 51 |
| 8.1.2. Pruebas y resultados del sistema obtenido..... | 53 |
| 8.2. Conclusiones..... | 60 |
| 8.2.1. Logros y Reflexiones | 60 |
| 8.2.2. Inconvenientes Presentados en el Desarrollo del Proyecto | 62 |
| 8.3. Trabajo Futuro..... | 64 |
| BIBLIOGRAFÍA..... | 65 |

1. INTRODUCCIÓN

Actualmente se está pasando por una era de innovación incesante, donde existe una gran propagación tanto de redes sociales como de aplicaciones móviles, las cuales funcionan como medio de promoción efectiva de los nuevos avances tecnológicos. Estos avances se apoyan generalmente en los dispositivos electrónicos, por ejemplo, los teléfonos inteligentes.

Los teléfonos inteligentes se han convertido en una herramienta esencial para explotar algunas de las nuevas tecnologías basadas en la comunicación, sirviendo como puente de conexión entre los desarrollos tecnológicos y los usuarios. Estos dispositivos electrónicos (teléfonos inteligentes), tienen la capacidad de entregar información en tiempo real del control que se le quiera realizar a un determinado producto o proceso, cuando son adoptados por las empresas o compañías que buscan una gestión comercial apropiada.

La gestión comercial desempeñada por las compañías es ejecutada por medio de técnicas diversas. Aquellas compañías que optan por utilizar los teléfonos inteligentes deben tener claridad sobre la información que requieren y la forma como piensan obtenerla. En este orden de ideas, las compañías deben contratar un desarrollador de aplicaciones móviles, al cual le presenten el modelo de los datos que requieren y una estructura base. Es de este modo, como el desarrollador de aplicaciones móviles al momento de elaborar la aplicación móvil debe tener en cuenta el tipo de programación que desea implementar, las cuales deben satisfacer las necesidades planteadas por la compañía. También, debe planear la ejecución del proceso y gestionar los permisos a tener en cuenta, según entidades como Google, Microsoft, entre otras. Principalmente si se desea emplear un servidor web.

El resultado obtenido con las aplicaciones móviles en algunas ocasiones debe ser complementado con otros procesos o técnicas basadas en sistemas de computación. En esta ocasión cabe relacionar las técnicas de visión por computador y procesamiento de imágenes. Estos son campos que se encargan de utilizar datos de alta dimensión del mundo real, por medio de métodos de adquisición, análisis y procesamiento de diferentes tipos de datos, con el fin de extraer información numérica o simbólica, entre las más destacadas. Estas técnicas tienen múltiples aplicaciones en la industria en donde se hace apropiado mencionar la organización de la información.

Obtener información y organizarla por parte de las diferentes compañías momento de la gestión comercial es muy importante. Por tal motivo, generalmente las compañías adoptan la tecnología moderna, buscando sistematizar procesos y disminuir el margen de error al momento de adquirir la información.

Una compañía prestadora de servicios puede llegar a verse muy afectada cuando no tiene información precisa de la gestión comercial, al momento de desempeñar sus labores. A continuación, se plantea una problemática que da muestra de ello.

2. PLANTEAMIENTO DEL PROBLEMA

Las compañías prestadoras de servicios se enfocan en obtener dinamismo y lograr una gestión comercial óptima y eficiente. El objetivo de lo anterior es reducir los recursos utilizados para alcanzar las metas de la organización. Esto se logra teniendo el máximo control posible de su gestión comercial. En este orden de ideas, el avance de la tecnología moderna generalmente es adoptada y explotada tanto por microempresas como por multinacionales de gran impacto en el mundo. La gestión comercial es potencializada con el manejo de la información de manera correcta y es de gran interés no solo para las compañías, sino también, para el gobierno de los países.

Los gobiernos de cada país generalmente buscan tener conocimiento sobre las acciones comerciales de las compañías u organizaciones tanto públicas como privadas. Por tal motivo, se han diseñado y puesto en práctica diferentes normas por parte de algunas entidades del estado colombiano, en pro del mejoramiento de prácticas industriales, buscando tener información precisa sobre las gestiones comerciales de las compañías.

Las compañías de envasado y distribución de GLP (Gas Licuado de Petróleo) a nivel regional, buscan cumplir con la normatividad vigente impuesta por entidades del estado, pero no cuentan con las herramientas necesarias para lograrlo.

Con respecto a la normatividad vigente, cabe resaltar la Resolución 023 (5 de marzo de 2008) de la Comisión de Regulación de Energía y Gas (CREG). Esta resolución está enfocada principalmente a que las compañías lleven un registro y control de todos los cilindros que se van a comercializar, [1].

Una forma de registrar la información entre muchas otras es la siguiente:

En primer lugar, el distribuidor encargado debe escribir un número serial del cilindro que va a entregar al cliente, su respectiva capacidad, dirección de la ubicación del mismo y número telefónico del responsable. Además, debe escribir también, el número serial del cilindro vacío que el cliente le entrega y su respectiva capacidad. Todo esto se registra de manera manual en un primer documento.

En segundo lugar, cuando el distribuidor encargado ingresa a la planta de envasado, un funcionario de la compañía toma de manera escrita los números seriales y la respectiva capacidad de los cilindros, tanto llenos como vacíos, que el distribuidor trae en su camión. En el momento en que los distribuidores van a salir de la planta de envasado con una determinada cantidad de cilindros llenos, se toman nuevamente de manera escrita los números seriales y su respectiva capacidad. Todo esto se registra en un segundo documento.

Por último, los documentos anteriormente mencionados, llegan a un funcionario de la compañía que se encarga de ingresar la información a una base de datos.

La información anexada a la base de datos generalmente es poco confiable y verídica lo que afecta la gestión de cilindros GLP, debido que la recopilación de los datos es efectuada por el factor humano, haciendo esto que el margen de error sea mayor. Esto se debe a que las personas no siempre tienen el mismo estado de ánimo y en estado de cansancio son vulnerables a confundir los números en el momento de la lectura de estos. Las compañías de la región no cuentan con las herramientas necesarias para recopilar los datos ya mencionados con un margen de error mínimo, por esta razón, se hace muy difícil efectuar un buen control del proceso. Cuando no se tiene la información correcta respecto a la gestión comercial de los cilindros GLP es posible no tener la ubicación exacta de los cilindros, lo que se ve reflejado en la pérdida de los mismos. En este orden de ideas, otras compañías logran apoderarse de los cilindros a los cuales no se les puede hacer un seguimiento preventivo generando déficit financiero a la compañía de procedencia.

Una posible solución a la problemática consiste en una mejor gestión y control de cilindros aprovechando las herramientas electrónicas y computacionales, usando principalmente un teléfono inteligente y un servidor orientado a procesar y organizar datos.

Buscando detallar un poco la normatividad que deben cumplir las compañías de envasado y distribución de GLP, a continuación, se relaciona lo que motiva específicamente al desarrollo del proyecto.

3. MOTIVACIÓN

La Comisión de Regulación de Energía y Gas (CREG), es una de las entidades del estado Colombiano encargada de regular la forma de distribución de los cilindros GLP, y las diferentes actividades que se desarrollan en pro de tener seguridad respecto a la distribución comercial para uso doméstico e industrial. En este orden de ideas, a continuación se resaltan tres apartes de la Resolución 023 del 05 de marzo del año 2008 (de la CREG) de gran interés que justifican el desarrollo de este proyecto.

El ítem 3 del artículo 3 del capítulo 1 estipula que las compañías deben: “Crear las condiciones e instrumentos para la operación eficiente y segura de la distribución y comercialización minorista de GLP”, [1].

El ítem 4 del artículo 6 del capítulo 3 estipula que: “Las compañías deben garantizar para sí mismo el uso exclusivo de los cilindros de su propiedad y su trazabilidad, para lo cual, entre otras cosas, establecerá los mecanismos de seguimiento y control que sean necesarios con el fin de conocer en todo momento la localización del parque de su propiedad”, [1].

El ítem 7 del artículo 7 del capítulo 3 estipula que: “Las compañías deben llevar un registro consecutivo de datos de todos los cilindros que envasa que permita la trazabilidad de los mismos y para tal fin debe dotarlos de un mecanismo apropiado. En ambos casos deberá ajustarse a las exigencias técnicas que para el efecto establezca el ministerio de minas y energía”, [1].

Actualmente, no se conocen sistemas confiables de gestión de cilindros en la región que ayuden a cumplir con la normatividad vigente para las empresas del sector. Una posible razón a lo anterior, radica en no se emplean herramientas electrónicas enfocadas en realizar un registro y control de cilindros GLP con un mínimo margen de error.

Las empresas del sector se encuentran vulnerables a una serie de multas y sanciones impuestas por el estado Colombiano, además, financieramente también se ven afectadas con la pérdida de los cilindros de su propiedad por falta de una trazabilidad correcta.

De este modo, se exponen a continuación los objetivos de este trabajo, partiendo del hecho que todo está direccionado en desarrollar un sistema que ayude a las compañías de envasado y distribución de GLP a cumplir con la normatividad vigente.

4. OBJETIVOS

Para trazar la ruta a seguir y lograr cumplir con las metas a desarrollar en el proyecto se han estipulado los siguientes objetivos:

4.1. Objetivo General

Desarrollar un sistema de gestión de cilindros mediante procesamiento de imágenes tomando como herramienta un dispositivo móvil.

4.2. Objetivos Específico

1. Adecuar una base de datos según las necesidades de programación que requiera una compañía para lograr la interoperabilidad con la aplicación móvil desarrollada.
2. Desarrollar un sistema de procesamiento de imágenes que permita extraer información útil para alimentar una base de datos.
3. Diseñar una aplicación móvil que permita gestionar los cilindros GLP para evitar la pérdida de los mismos.
4. Verificar que el sistema diseñado permita dar cumplimiento por parte de las empresas de envasado y distribución de GLP (Gas Licuado de Petróleo) al ítem 4 del artículo 6 del capítulo 3 de la resolución 023 (5 de marzo de 2008) y el ítem 7 del artículo 7 del capítulo 3 de esta misma resolución, de la Comisión de Regulación de Energía y Gas (CREG).

Para lograr cumplir con los objetivos previstos, es necesario obtener información útil sobre las compañías de envasado y distribución de GLP, desarrollo de aplicaciones móviles, sistemas de procesamiento de imágenes y organización de información en bases de datos. Por tal motivo, En el siguiente capítulo se relaciona el marco de referencia con información apropiada para el desarrollo de este proyecto.

5. MARCO DE REFERENCIA

El modelo explicativo del marco de referencia que se presenta a continuación está conformado por el marco histórico y el marco teórico. Estos marcos están estructurados de manera que se relacionan tres temas importantes para el desarrollo de este trabajo, los cuales son, aplicaciones móviles, procesamiento de imágenes y bases de datos. Hay que destacar que a manera de información se exponen temas sobre la industria del GLP.

5.1. Marco Histórico

El desarrollo de este proyecto está enfocado en ayudar a mitigar una problemática que presentan ciertas compañías de envasado y distribución del GLP (Gas Licuado de Petróleo). Por tal motivo, vale la pena mencionar algunos datos históricos sobre esta industria.

A inicios del siglo XX se presentaba un problema con la gasolina que se producía y era almacenada. Esta se evaporaba rápidamente y la razón era el propano y butano que contenía. Luego de unos estudios, un químico norteamericano logra separar estos gases de la gasolina. Es así como da inicio en los años 20 la producción del Gas Licuado de Petróleo (GLP) que es producto del gas propano, [2].

Entre los años 30 y 40 se introdujo el GLP en Francia. Poco a poco el mercado del GLP fue creciendo a medida que aumentaba la producción de los cilindros y la construcción de nuevas refinerías. En Europa en los años 50, 60 y 70, se vendieron 300 mil, 3 millones y 11 millones de toneladas respectivamente, [2].

Culminando los años 30 da inicio la industria del GLP en Colombia, con una pequeña producción entregada por las refinerías de Tibú y Barrancabermeja, [2].

El uso del GLP en Colombia en sus inicios estaba dirigido principalmente al sector doméstico. Entró a competir con el carbón y la energía eléctrica que eran los principales energéticos en el momento. En este orden de ideas, la demanda del GLP fue aumentando considerablemente debido a su bajo precio, [3].

Hasta el año 2000 en Colombia el consumo del GLP estuvo aumentando gradualmente. Luego empieza a disminuir debido principalmente por un alza considerable en los precios y la llegada del gas natural, [3].

En el 2008 la Comisión de Regulación de Energía y Gas (CREG) presentó el nuevo modelo de comercialización de GLP, destacando normas y reglas enfocadas a mejorar el servicio y la seguridad para el usuario final, [3].

Ahora se da lugar a detallar información destacada sobre las aplicaciones móviles.

APLICACIONES MÓVILES

Los teléfonos inteligentes han servido como herramienta para promocionar nuevas ideas de negocio. Pero algo a destacar es que estos teléfonos inteligentes cobran mayor interés con las aplicaciones móviles que se pueden alojar en él. Generalmente las aplicaciones móviles no generan ganancias (en cuanto a lo financiero se refiere) directas para la empresas que las utilizan, pero sirven como soporte para mejorar la comunicación y calidad de servicios que prestan, [4].

El uso de las aplicaciones móviles crece de una manera exponencial, debido a la gran acogida por parte de los usuarios. En 2013 el número de aplicaciones móviles aumentó un 115% más que el año anterior, [4].

Se espera que para los próximos años, el mercado de las aplicaciones móviles en el mundo crezca alrededor de un 85%, lo que equivale a un volumen de negocio de 27 millones de dólares, según datos de ABI Research, iOS y android, [11].

Cabe mencionar que, según un informe firmado por ‘*Forbes Insights y Adobe*’, el 83% de las compañías utilizan las aplicaciones móviles para mejorar la comunicación con sus clientes, [4].

Ahora se da lugar a detallar información destacada sobre el campo del procesamiento de imágenes.

PROCESAMIENTO DE IMÁGENES

A lo largo de la historia de la humanidad, existe el deseo de registrar una instancia de tiempo para las generaciones futuras mediante el uso de imágenes. Esto se ve reflejado en los dibujos creados por los cavernícolas. Las imágenes tienen la facilidad de comunicar información de un humano a otro, [5].

La necesidad por buenas imágenes para el reconocimiento militar durante la segunda guerra mundial, generó o promovió avances en los métodos o técnicas para procesos fotográficos. Un ejemplo de este desarrollo fue agrandar y cambiar el contraste de una imagen para facilitar el trabajo de reconocimiento de objetivos militares. El procesamiento de imágenes comenzó a existir con el desarrollo de los computadores digitales. Los primeros procesamientos de imágenes se hicieron en la nasa en los laboratorios JPL (*Jet Propulsion Laboratory*) en Pasadena California para sus programas espaciales, [5].

El procesamiento de imágenes es usado en criminalística, en los campos de la astronomía, detección de huellas dactilares, en la industria manufacturera, en la industria del entretenimiento, entre otros. En astronomía es muy utilizado filtrar imágenes y eliminar el ruido espacial aplicando diferentes tipos de filtros, [6].

El campo de la medicina ha sido beneficiado con el uso de procesamiento de imágenes. Las técnicas de procesamiento han sido aplicadas a las imágenes de ultrasonido, para el control de los fetos durante el cuidado prenatal. También se han utilizado para la detección de cáncer de mama y fracturas por medio de rayos x, [6].

Luego de haber relacionado alguna información de interés, se da paso a exponer conceptos importantes a modo explicativo.

5.2. Marco Teórico

En esta sesión se relacionan temas sobre las aplicaciones móviles, procesamiento de datos y bases de datos.

APLICACIONES MÓVILES

Para diseñar una aplicación móvil en la plataforma Android Studio se maneja un lenguaje de programación en java orientada a objetos. De este modo, a continuación se relaciona información de interés sobre este lenguaje de programación.

PROGRAMACIÓN ORIENTADA A OBJETOS

Con el paso de los años se han creado diferentes herramientas enfocadas al desarrollo de software buscando aumentar su productividad, entre las cuales cabe destacar, el lenguaje de programación orientado a objetos.

La programación orientada a objetos es un procedimiento de diseño de software que define programas en términos de clases de objetos. Estos objetos combinan datos y métodos, [7]

Este tipo de programación es un programa conformado por objetos. Los objetos tienen características y procedimientos definidos. Es así cómo se logra una comunicación entre ellos para realizar la labor que se requiera, enfocándose en un modelo de representación del mundo real, [7].

En la programación orientada a objetos cuando se diseñan adecuadamente las clases, estas se pueden usar en diferentes partes del programa, convirtiendo esto en una gran ventaja al momento de programar. Cabe aclarar que con el término clases, se hace referencia a una especie de plantilla para programar objetos, [7].

Una forma de definir los objetos en programación es la siguiente: Son unidades básicas de construcción enfocadas a diseñar y programar, [7].

Los objetos se comunican a través de señales con información, en base a estas señales el objeto da una respuesta. En este orden de ideas, con esta señal se le da una orden al objeto de ejecutar alguno de sus métodos. El método es una acción que indica que labor debe desempeñar el objeto cuando recibe la señal, [7].

Java es uno de los lenguajes orientados a objetos más utilizados. A continuación se relaciona un poco de información sobre este lenguaje de programación.

PROGRAMACIÓN ORIENTADA A OBJETOS EN JAVA

El lenguaje de programación java tiene como características principales que es sencillo, dinámico, distribuido, seguro y de alto rendimiento. Este emplea un compilador que traduce de código fuente a ejecutable, [7].

Java tiene un kit de desarrollo conocido como JDK, el cual contiene las librerías necesarias para la creación y ejecución de diferentes aplicaciones, [7].

En base a lo ya mencionado, un objeto tiene una serie de atributos y métodos. Vale la pena mencionar que partes conforman un método. Este cuenta con el nombre, valor que regresa, Argumentos, modificadores y el cuerpo, [7].

Como se ha podido notar, la programación orientada a objetos es una herramienta muy utilizada para diseñar aplicaciones móviles. A continuación, se expone brevemente la relación de las aplicaciones móviles con el campo de la visión por computador.

APLICACIONES MÓVILES CON VISIÓN POR COMPUTADOR

La visión por computador es un campo que se encarga de utilizar los datos de alta dimensión del mundo real, por medio de los métodos de adquisición, análisis y procesamiento de imágenes, con el objeto de extraer información numérica o simbólica, [8].

Algunas de sus aplicaciones incluyen sistemas de control para procesos en la industria robótica, detección de eventos (Ej. vigilancia visual y conteo de personas), organización de la información, modelado de objetos y entornos entre otros.

Este campo puede ser muy aprovechado tomando como herramienta los teléfonos inteligentes, debido que si se cuenta con una aplicación móvil apropiada, es posible recopilar información útil para que sea procesada por un sistema de visión por computador o procesamiento de imágenes.

Por consiguiente, se da espacio a relacionar temas sobre procesamiento de imágenes.

PROCESAMIENTO DE IMÁGENES

En primer lugar, se mencionan a continuación algunos fundamentos teóricos importantes.

FUNDAMENTOS DE PROCESAMIENTO DE IMÁGENES

El mundo se puede percibir con infinidad de formas y colores, que son un influyente esencial para emplear técnicas de procesamiento de imágenes.

La percepción humana tiene la capacidad para adquirir, integrar e interpretar la información que nos rodea. Los estudios científicos buscan vincular estas capacidades a una máquina de modo que la información pueda ser procesada y analizada con sistemas precisos y autónomos. En este orden de ideas, es fundamental comprender las diferentes técnicas que entregan los estudios hasta el momento enfocadas al tratamiento de imágenes, [9].

Para diseñar un sistema de procesamiento de imágenes, se deben adquirir las mismas de forma digital, utilizando diferentes herramientas, por ejemplo, cámaras digitales. Las cámaras capturan la imagen bajo conceptos que no caben mencionar en esta oportunidad. En ocasiones las imágenes resultan ruidosas producto de la influencia de mecanismos externos. Este ruido es conocido también como una degradación en la imagen. Una fuente común de degradación de la imagen es la lente óptica, encargada de adquirir información de tipo visual. Si la cámara no se centra adecuadamente o se configura mal, es muy probable que se obtengan imágenes borrosas o con efectos de difuminación, [9].

El mejoramiento de las imágenes, filtrado, y la restauración han sido algunas de las aplicaciones importantes de procesamiento de imágenes desde los primeros días de estudio de esta rama de la ciencia, [9].

Para realizar procesamiento de imágenes buscando acondicionar las mismas, se deben realizar diferentes procesos. A continuación se da lugar a explicar un poco la segmentación.

La segmentación es el proceso que subdivide una imagen en un número de regiones uniformemente homogéneas. De este modo, se puede decir que la segmentación de una imagen es definida por un conjunto de regiones que están conectados y no se solapan. La imagen adquiere una etiqueta única por pixel que indica la región que pertenece. La segmentación es uno de los elementos más importantes en el análisis automatizado de imágenes, sobre todo porque en este paso los objetos u otras regiones de interés se extraen de una imagen para su posterior procesamiento, como la descripción y el reconocimiento, [9].

Después de extraer cada segmento; la siguiente tarea consiste en extraer un conjunto de indicadores y características significativas tales como color y forma, de manera que sirvan de identificación. Para ello es importante poder medir las diversas propiedades de los segmentos de imagen. Los descriptores de forma más comunes son la longitud, ancho, área, ubicación, entre otros. Por consiguiente, basado en las características extraídas, cada objeto segmentado se clasifica en un conjunto de clases significativas. La segmentación y clasificación de objetos son dos áreas integradas a la visión por computador, [9].

Se da paso a otro aspecto de procesamiento de imágenes, que consiste en la compresión y codificación de la información de tipo visual. Los requisitos de almacenamiento de imágenes digitales están creciendo muy rápidamente. Representar datos en forma compacta de una imagen para almacenarlos o transmitirlos a través de ancho de banda de comunicación, es una área de gran investigación en el momento, todo por el interés existente en lograr más avances en la materia, [9].

Respecto a la imagen, a continuación se relacionan datos sobre su formación.

FORMACIÓN DE LA IMAGEN

Comprender la física de la iluminación es necesario para la comprensión de la formación de una imagen. Comenzamos una descripción breve en el siguiente texto.

ILUMINACIÓN

La iluminación es un componente fundamental en el proceso de formación de la imagen, que cobra sentido en el órgano visual (los ojos). La luz produce una sensación psicológica cuando incide sobre los ojos y excita el sentido de la vista. La fuerza de esta sensación, que es la sensación de brillo, puede ser identificada promediando las respuestas de muchos observadores humanos. Partiendo de este concepto, se analizan y desarrollan diferentes elementos electrónicos diseñados para capturar imágenes. [9].

Cuando se maneja una iluminación apropiada se logran aprovechar mucho mejor las diferentes técnicas de procesamiento de imágenes. En este orden de ideas, cabe destacar el proceso de muestreo y cuantización.

MUESTREO Y CUANTIZACIÓN

La comprensión del proceso de muestreo y cuantificación es una de las zonas claves de procesamiento de imágenes. Las teorías de muestreo de imagen y cuantificación se han investigado a partir de las diferentes dimensiones de las mismas. El concepto de muestreo se refiere al proceso de subdividir una imagen en porciones. La cuantificación es el proceso que se aplica luego del muestro, este permite tomar valores de amplitud de la señal, [9].

Entre las técnicas de procesamiento es fundamental destacar aquellas que se emplean para acondicionar la imagen. Veamos algunos resultados de estas.

IMAGEN BINARIA

Las imágenes pueden ser binarias, grises o a color. Los píxeles en una imagen binaria puede asumir únicamente dos valores, cero (0) o uno (1). Las imágenes binarias son de menos peso computacionalmente hablando. Estas requieren menos espacio para su almacenamiento y menor capacidad computacional para su procesamiento digital. Estas imágenes contienen suficiente información acerca de los objetos de la imagen y se pueden reconocer fácilmente. Las imágenes binarias son utilizadas en el campo de la visión por computador. Cabe aclarar que una imagen en escala de grises se puede pasar a imagen binaria por el proceso de umbralización, [9].

El procesamiento de imágenes es un campo de estudio e investigación muy amplio. Por tal motivo, A continuación se destacan algunos aspectos a destacar.

DATOS A DESTACAR EN PROCESAMIENTO DE IMÁGENES

Hay algunas notas importantes que necesitan ser recordados para el diseño de un sistema de procesamiento de imágenes. Es importante asegurarse de que el aparato electrónico con el que se capturen las imágenes esté configurado de manera apropiada antes de hacer una captura. También, manejar un entorno de iluminación óptimo. Si el nivel de iluminación es baja, puede dar lugar a la baja exposición del sensor del dispositivo electrónico, mientras que demasiada iluminación puede conducir a la sobreexposición. De forma general, en todos los casos se necesitan pre-procesamiento para la detección de objetos en una imagen. También es importante elegir la correcta resolución del dispositivo electrónico, de manera que las imágenes capturadas tengan la resolución adecuada del objeto, y de este modo, se logre un mejor reconocimiento automatizado, [9].

Por consiguiente, se da espacio a relacionar temas sobre bases de datos.

BASE DE DATOS

En primer lugar, se da una definición sobre lo que es un sistema de base de datos.

SISTEMA DE BASE DE DATOS

Un sistema de base de datos es un sistema computarizado utilizado para almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en secuencias y solicitudes. La información puede ser variable y de cualquier tipo. El objetivo principal de almacenar esta información es que pueda ser utilizada para apoyar diferentes procesos empresariales y comerciales que aporten a una buena administración empresarial, [10].

Los sistemas de bases de datos se ejecutan generalmente en diferentes computadores independientes de su capacidad, estos sistemas son diseñados partiendo de las especificaciones técnicas de la máquina. También, dependiendo de las necesidades de los usuarios u organizaciones se pueden tener sistemas para un usuario o multiusuarios. Un sistema de un solo usuario es aquel en el que sólo un usuario puede tener acceso a la base de datos en un momento dado; un sistema multiusuario es aquel en el cual múltiples usuarios pueden tener acceso simultáneo a la base de datos, [10].

VENTAJAS BASE DE DATOS

Existen diferentes razones y ventajas de peso que justifican el uso de las bases de datos. Es de destacar, si se trata de un sistema de base de datos de un usuario o multiusuario las ventajas son muy variables. Una de las principales ventajas a resaltar en este caso, es que facilita la organización y agilidad en el registro y control de procesos y servicios prestados por parte de compañías o usuarios finales, sirviendo esta base de datos como herramienta para realizar una trazabilidad correcta, [10].

En una base de datos es esencial manejar la independencia de los mismos.

Existen dos clases de independencia datos, física y lógica. Para este caso se enfatizará en los datos de clase física, [10].

Partiendo de los datos lógicos se podrá dar una mejor explicación a los datos de clase física. Tomando como ejemplo las aplicaciones no recientes, estas suelen ser dependientes de los datos. Esta idea esta enfatizada en las técnicas de acceso a los datos por medio del almacenamiento secundario, esto se logra por la implementación de código propio de la aplicación, [10]

Ahora con algunos conceptos claros, en el siguiente capítulo se relaciona una metodología implementada trazando un modelo a seguir para cumplir con los objetivos previstos.

6. METODOLOGÍA

Se diseña una metodología descrita a continuación por etapas. En cada una de estas etapas se exponen los objetivos logrados y las actividades que se llevaron a cabo para dar cumplimiento con los mismos.

6.1 Etapa 1

Se Adecua una base de datos según las necesidades de programación que requería una compañía para lograr la interoperabilidad con la aplicación móvil desarrollada.

Actividad 1 Se realiza un estudio de la base de datos que manejaba una compañía de envasado y distribución de GLP del sector, detectando sus faltantes y, por consiguiente, se adecua de manera que tuviera una independencia física y lógica de los datos y presentará una redundancia mínima.

Actividad 2 Es acondicionada la base de datos, de tal forma que es posible anexar información organizada, por medio de un número definido de dispositivos móviles enlazados de forma individual a un servidor (computador principal).

6.2 Etapa 2

Es desarrollado un sistema de procesamiento de imágenes el cual permite extraer información útil para alimentar la base de datos acondicionada.

Actividad 3 Se desarrolla una herramienta de software para la obtención del número serial del cilindro mediante procesamiento digital de imágenes, tomando los datos entregados por un dispositivo móvil (Imágenes capturadas).

Actividad 4 Es implementada la herramienta de software desarrollada. Es decir, la programación necesaria para compartir la información procesada (entregada por un dispositivo móvil) con la base de datos acondicionada.

6.3 Etapa 3

Se diseña una aplicación móvil la cual permite gestionar los cilindros GLP logrando también, disminuir el porcentaje de pérdida de estos.

Actividad 5 Se realiza un estudio de las diferentes plataformas disponibles en el desarrollo de las aplicaciones móviles, y así se pudo determinar cuál de ellas tenía mejores prestaciones para solucionar la problemática de gestión de cilindros GLP.

Actividad 6 Es desarrollada una aplicación móvil teniendo en cuenta las restricciones encontradas.

6.4 Etapa 4

Se verifica que el sistema diseñado ayude a dar cumplimiento por parte de las compañías de envasado y distribución de GLP con el ítem 4 del artículo 6 del capítulo 3 y el ítem 7 del artículo 7 del capítulo 3 de la resolución 023 (5 de marzo de 2008) de la Comisión de Regulación de Energía y Gas (CREG).

Actividad 7 Es implementado un plan de pruebas para verificar el funcionamiento y desempeño del sistema desarrollado.

Actividad 8 Se ejecuta un plan de pruebas de forma aleatoria durante un intervalo de tiempo determinado y se calcularon índices de desempeño.

Teniendo la metodología clara, en el siguiente capítulo se explica el modo de operación del sistema obtenido y cómo fue su desarrollo por medio de diagramas de proceso, tablas de información y argumentos detallados.

7. DESARROLLO

El desarrollo del proyecto se dividió por áreas. A continuación, se dará muestra del diagrama general de las áreas de desarrollo y su modo de operación:

7.1 Diagrama General de las Áreas de Desarrollo

En la **Figura 1** Relacionada a continuación se puede apreciar el diagrama general de las áreas de desarrollo.

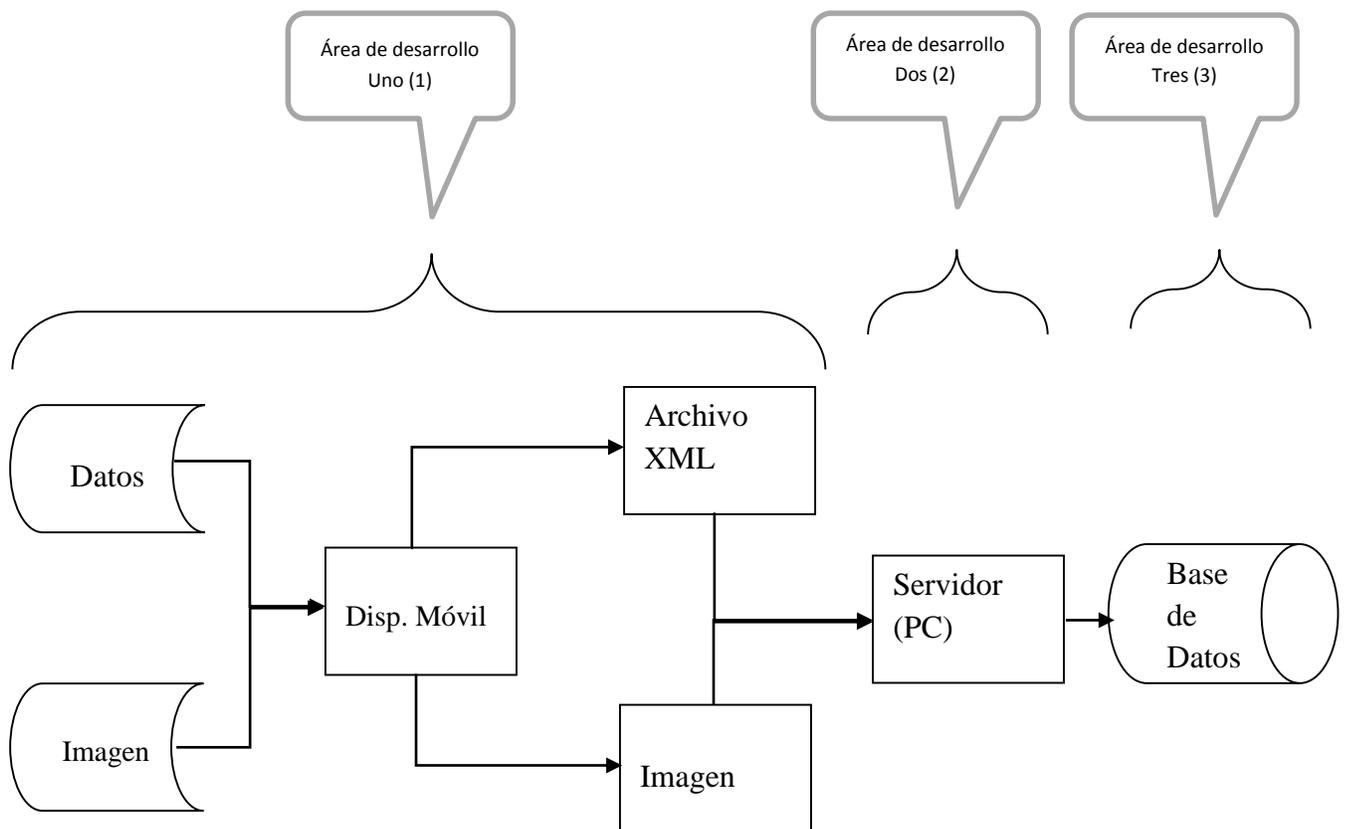


Figura 1. Diagrama general de las áreas de desarrollo

Las tres (3) áreas de desarrollo que se pueden apreciar en la **Figura 1** y su modo de operación será descrito a continuación.

7.1.1. Descripción General del Modo de Operar las Áreas de Desarrollo

En el área de desarrollo uno (1) se destaca una herramienta de trabajo fundamental, se trata de un dispositivo móvil o más conocido como teléfono inteligente. Un usuario determinado se encarga de capturar unas fotografías e ingresar una serie de datos a la App móvil 'GESTIÓN GLP' diseñada (Esta es explicada más adelante). La aplicación está diseñada para almacenar esta información en un formato XML. La información en este formato se encuentra ordenada de una manera específica, logrando así que el código de procesamiento alojado en un servidor (computador principal) pueda procesar los datos más óptimamente.

Es importante aclarar que las fotografías se le toman a la identificación (etiqueta) de los cilindros GLP que se comercializan.

A continuación se da muestra gráfica en la **Figura 2** de la puesta en práctica de este desarrollo.



Figura 2. Puesta en práctica área de desarrollo uno (1)

En la **Figura 2(a)** Se puede apreciar el modo en que se debe capturar la imagen de la identificación del cilindro. Por consiguiente, en la **Figura 2(b)** se nota que el paso a seguir es recopilar la información necesaria en la aplicación móvil haciendo uso del teléfono inteligente.

En el área de desarrollo dos (2) se destaca un herramienta de trabajo fundamental, se trata de un servidor (computador principal), el cual contiene un código diseñado para efectuar un

procesamiento de imágenes y organizar datos, los cuales están en un formato XML. Luego de procesar y organizar los datos, el código se encarga de complementar el proceso enviando esta información a una base de datos.

La base de datos es la conforma el área de desarrollo tres (3). Esta es la encargada de recibir la información enviada por el sistema de procesamiento de imágenes y datos. En esta están contenidas algunas fórmulas que realizan unos determinados cálculos para complementar la información. En este orden de ideas, la base de datos sirve como herramienta para obtener una trazabilidad apropiada de los cilindros comercializados. Cabe aclarar que el servidor (computador principal) debe ser manipulado por un usuario determinado, y este debe tener una buena ética laboral.

En la **Figura 3** relacionada a continuación, se da una breve muestra gráfica de la forma de concatenar las tres áreas de desarrollo:



Figura 3. Puesta en práctica concatenación áreas de desarrollo

La información recopilada y las imágenes capturadas en el teléfono inteligente, son descargadas al servidor el cual procesa y envía la información a una base de datos. En la **Figura 3** se puede apreciar como el operario descarga la información del celular al servidor (computador principal).

Ahora se da paso a tratar el tema relacionado al funcionamiento del sistema.

7.2. Diagrama Detallado del Funcionamiento del Sistema

En la **Figura 4** Relacionada a continuación se puede apreciar el diagrama detallado del funcionamiento del sistema.

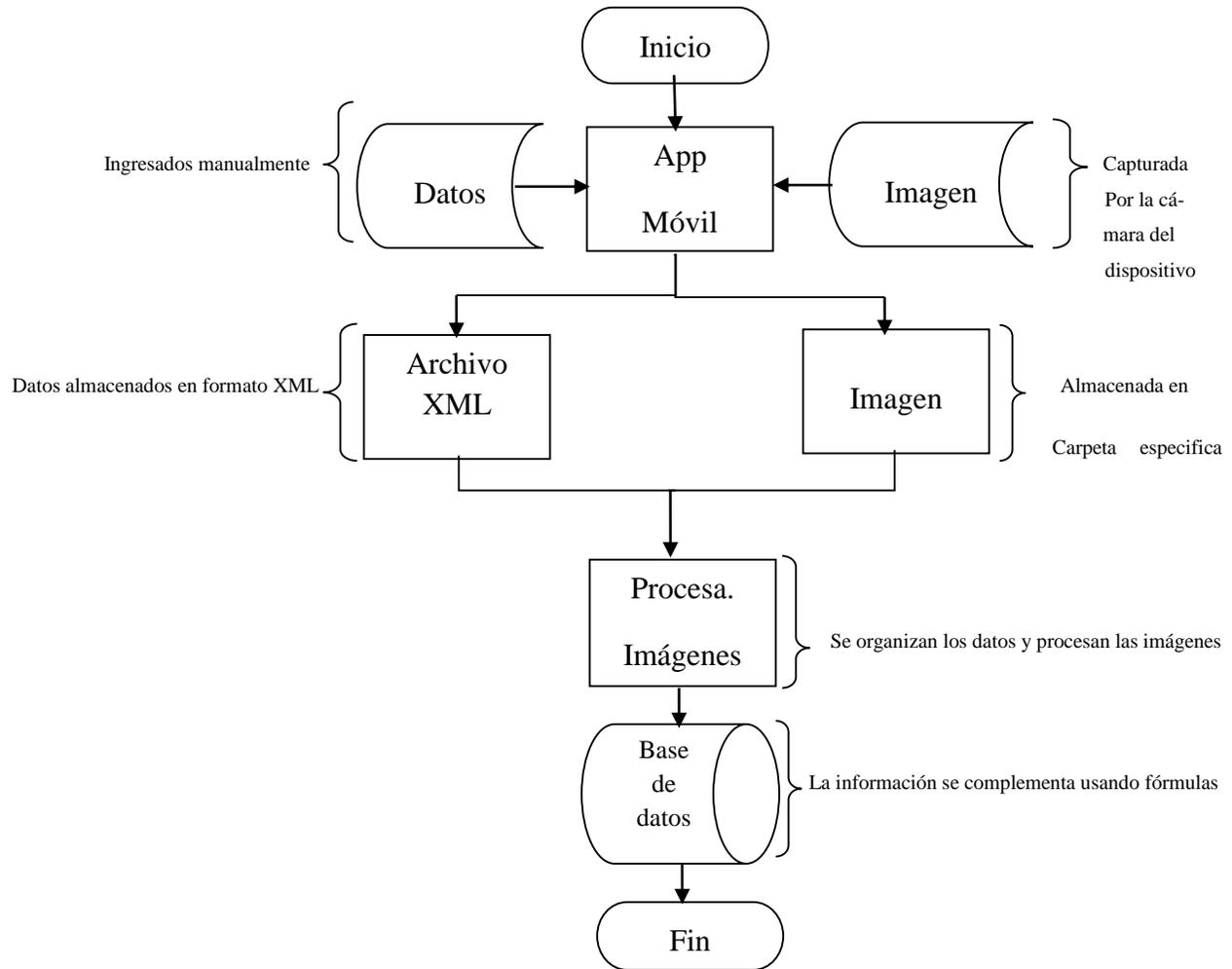


Figura 4. Diagrama detallado del funcionamiento del sistema

Teniendo claro cuáles son las tres áreas de desarrollo se da lugar a detallar su funcionamiento.

7.2.1. Descripción Detallada del Funcionamiento del Sistema

La App móvil se desarrolló en el software libre 'Android Studio' (Esta será explicada detalladamente más adelante). El usuario que manipula esta App móvil por medio del teléfono inteligente debe ingresar una serie de datos y fotografiar una etiqueta de los cilindros que se van a comercializar.

La App móvil tiene dos tipos de usuario, pero el principio de diseño de ambos es el mismo. A continuación se da muestra en la **Figura 5** y **Figura 6** de los tipos de usuario y la forma de cómo están estructurados:

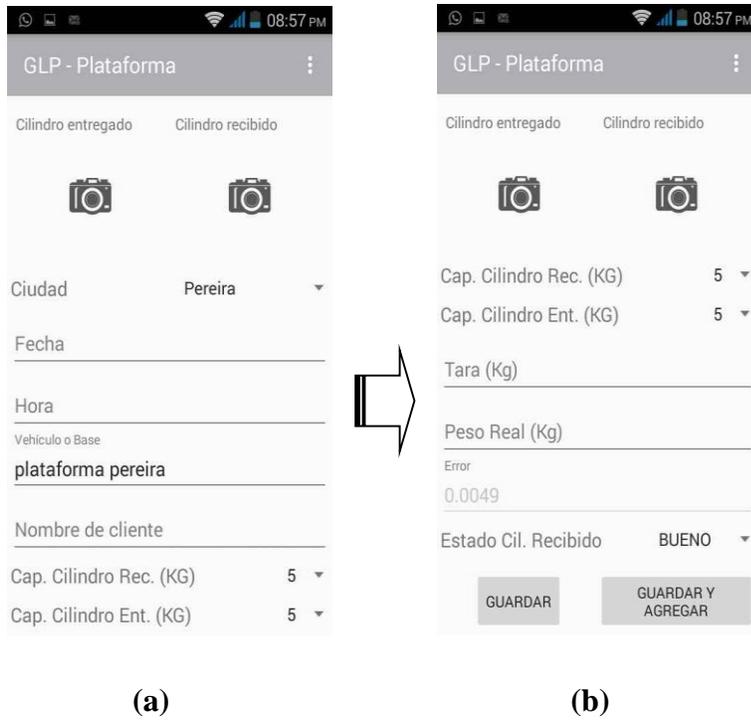


Figura 5. Modelo App móvil usuario plataforma

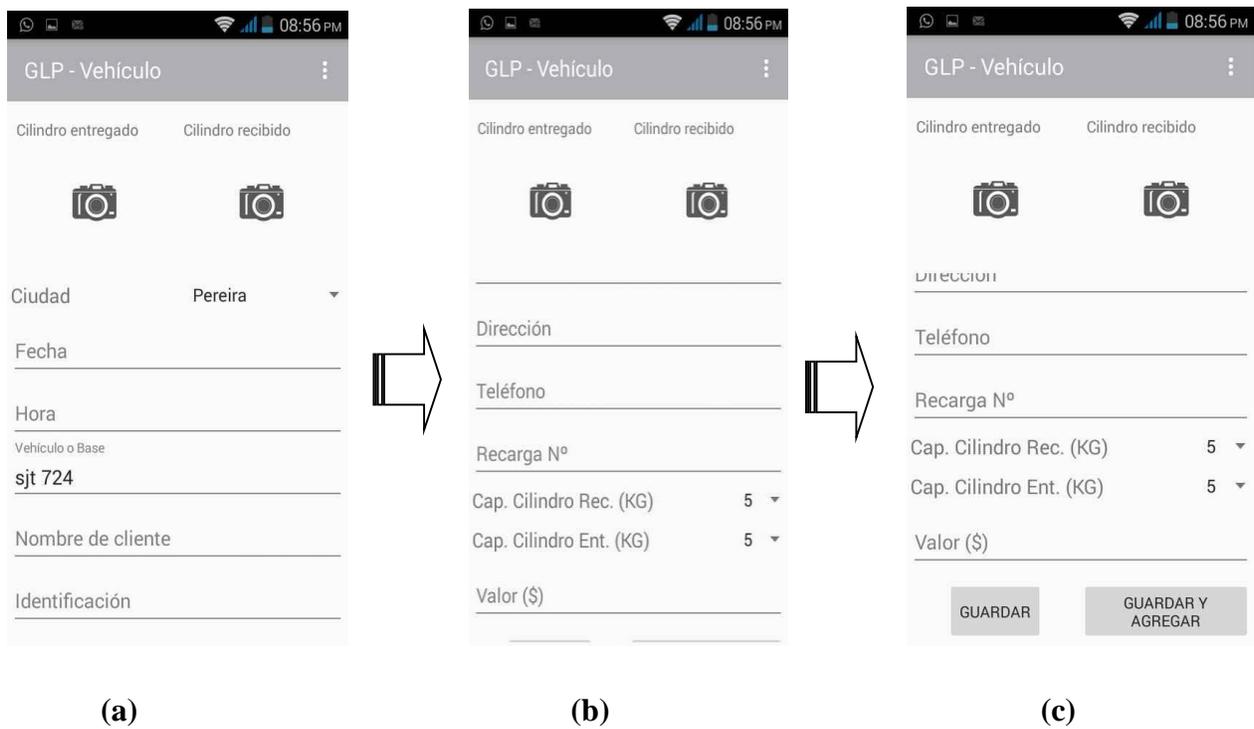


Figura 6. Modelo App móvil usuario vehículo

Por cuestiones de limitaciones gráficas, se da muestra de los tipos de usuario en diferentes fases como se pudo observar en las **Figuras 5** y **Figura 6** relacionadas anteriormente.

En este orden de ideas, la aplicación móvil se encarga de entregar un archivo en formato XML y una serie de fotografías almacenadas en una carpeta determinada. El archivo en formato XML contiene toda la información ordenada de una manera específica. La información está constituida por dos grupos de datos, un grupo está conformado por los datos ingresados por el usuario, y el otro grupo por datos que se cargan predeterminadamente. Las fotografías son almacenadas con un nombre de identificación estratégico. A continuación se da muestra del modelo estructural del archivo XML. También, se muestra en la **Figura 7** la forma de almacenamiento de las fotografías en el dispositivo móvil.

Modelo estructural archivo XML

```
<? xml version="1.0" encoding="utf-8" standalone="yes" ?>
<entries>
<entry>
<N>8</N>
<CIUDAD>Pereira</CIUDAD>
<FECHA>25/3/2016</FECHA>
<HORA>17:1</HORA>
<VEHICULO>plataforma pereira</VEHICULO>
<NOMBRE_CLIENTE>simón castañeda</NOMBRE_CLIENTE>
<IDENTIFICACION>1088654328</IDENTIFICACION>
<DIRECCION>transv. 7 # 26-04 Sta. isabel </DIRECCION>
<TELEFONO >3137811940</TELEFONO>
<CILINDRO_RECIBIDO>/storage/sdcard0/Pictures/GLP/CR_8.jpg</CILINDRO_RECIBIDO>
<CAP_CIL_REC>15</CAP_CIL_REC>
<CILINDRO_ENTREGADO>/storage/sdcard0/Pictures/GLP/CE_8.jpg</CILINDRO_ENTREGADO>
<CAP_CIL_ENT>15</CAP_CIL_ENT>
<TARA_CIL_ENT>13,5</TARA_CIL_ENT>
<PESO_REAL>28,5</PESO_REAL>
<ERROR>0,0049</ERROR>
```

```
<ESTADO>BUENO</ESTADO>
<VALOR>46100</VALOR >
<N_RECARGA>23456</N_RECARGA>
<PESO_ESPERADO>28.5</PESO_ESPERADO>
<C_C_E_ERROR>0.074</C_C_E_ERROR>
<CUMPLE>SI</CUMPLE>
</entry>
</entries>
```

El anterior ejemplo es un archivo en formato XML entregado por la aplicación móvil desarrollada. Cabe destacar que XML es un lenguaje de marcas muy utilizado para almacenar datos de forma legible. Un lenguaje de marcas se refiere a una forma de codificar un documento, el cual incorpora etiquetas que junto con el texto entreguen información adicional sobre su estructura.

El archivo XML y las imágenes capturadas son almacenadas en una carpeta específica en el teléfono inteligente como se puede observar a continuación:

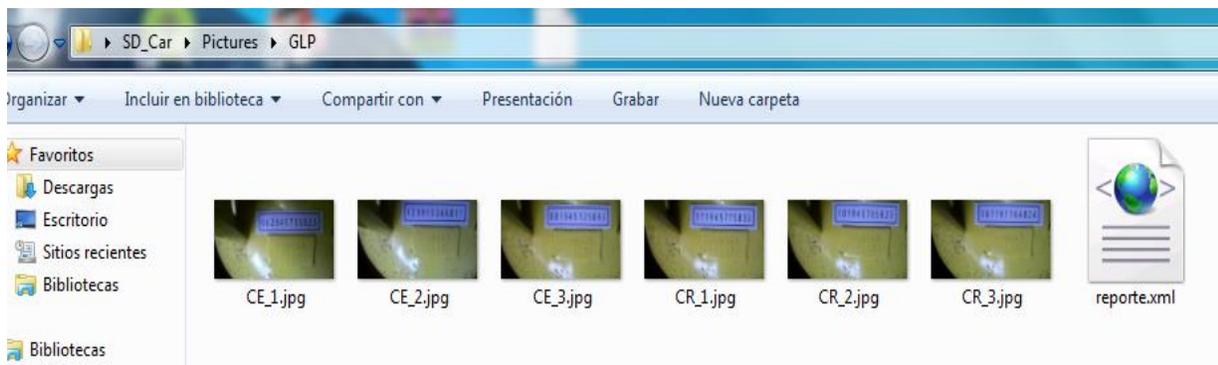


Figura 7. Forma de almacenamiento de fotografías y archivo XML

En la **Figura 7**. Se puede observar que las fotografías y el archivo XML generado se almacenan en el teléfono inteligente en la carpeta *GLP* que se encuentra en la carpeta *Pictures* de la *SD Car*.

Haciendo uso de una App móvil gratuita conocida como **airdroid** que se encuentra en la Play Store de Android, se enlazan mediante la red WIFI del escenario el dispositivo móvil (con la aplicación móvil desarrollada) con el computador que contiene el código de procesamiento de imágenes y datos. Por consiguiente, un usuario determinado pasa la información necesaria entregada por la

aplicación móvil al computador en un lugar específico y ejecuta el código, enviando la información procesada a una base de datos.

Ahora se da lugar a explicar el proceso realizado en las diferentes áreas de desarrollo.

7.3 Diagrama del Funcionamiento de la App Móvil

En la **Figura 8** Relacionada a continuación se puede apreciar el diagrama del funcionamiento de la App móvil.

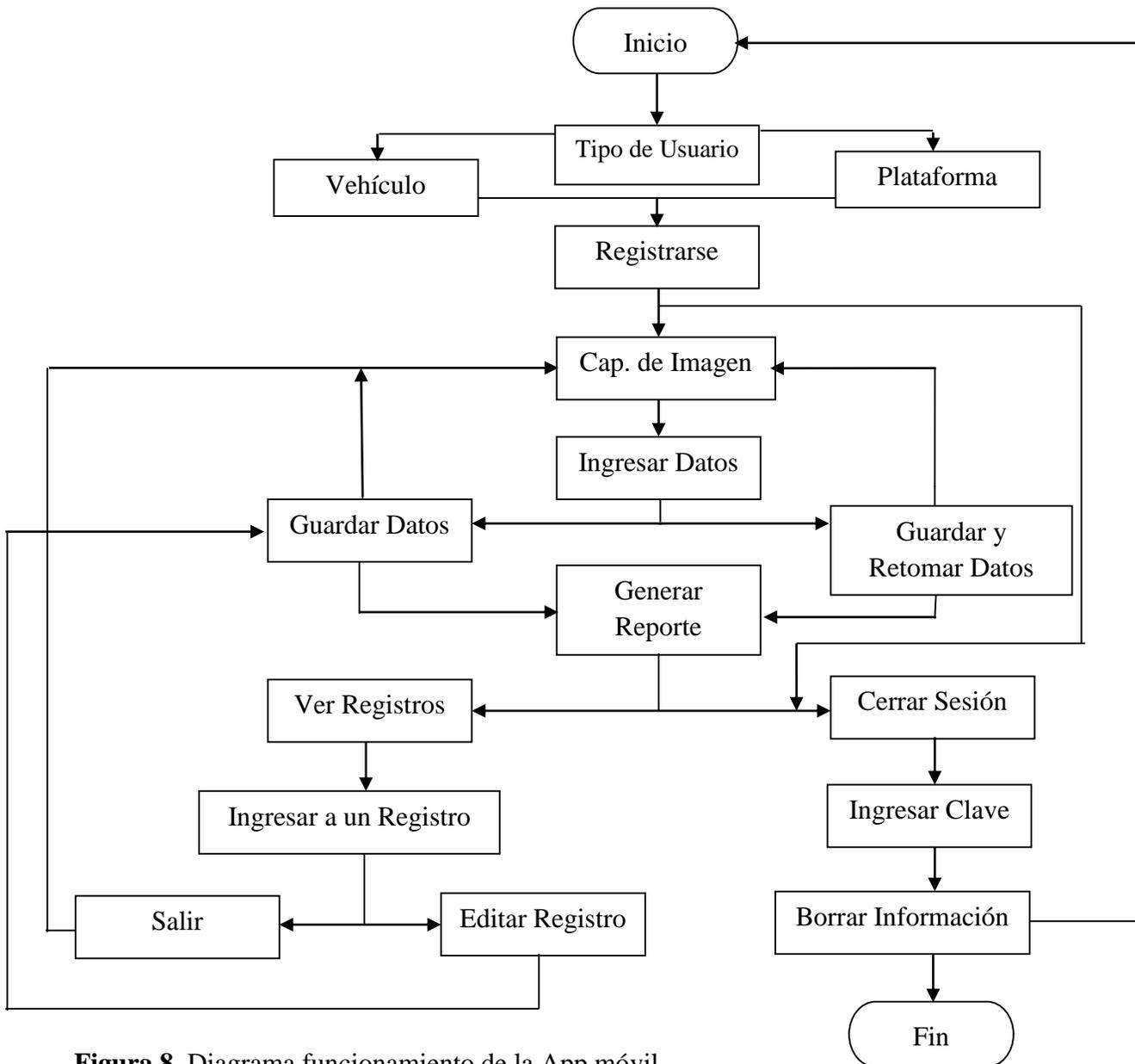


Figura 8. Diagrama funcionamiento de la App móvil

En el diagrama de funcionamiento de la aplicación móvil relacionado en la **Figura 8** se puede interpretar lo siguiente: Al iniciar la aplicación móvil por primera vez se debe elegir el tipo de usuario que puede ser tipo vehículo o tipo plataforma. En la segunda etapa, el usuario debe registrarse e ingresar al área de formulario. En ese punto el usuario captura las imágenes e ingresa los datos. Por consiguiente, se procede a guardar los datos, con la opción de retomar los últimos que hayan ingresados al formulario. Luego de esto se puede seguir agregando y guardando información y en la siguiente etapa se puede generar el reporte, es decir, se almacena la información en un formato XML. A continuación se pueden ver los registros generados hasta el momento e ingresar a cada uno de ellos para editarlos si es necesario. También, se puede cerrar la sesión ingresando una clave y borrar la información recopilada.

7.3.1. Descripción Detallada del Funcionamiento de la App Móvil

Esta descripción se detalla en dos etapas, en la primera se explica cómo es su funcionamiento y modo de emplearse para el registro y control de cilindros GLP, en la segunda etapa se tratan temas más técnicos relacionados con su desarrollo en la plataforma ‘Android Studio’.

FUNCIONAMIENTO Y MODO DE EMPLEARSE

Luego de instalarse la Aplicación móvil ‘GESTIÓN CILINDROS GLP’ en el dispositivo, el usuario debe registrarse, para ello al abrir la Aplicación por primera vez aparece en la pantalla el modo de elegir el tipo de usuario (‘Vehículo’ o ‘Plataforma’) y un espacio para ingresar la placa del vehículo o el nombre de la plataforma del usuario. Cabe destacar, que dependiendo del tipo de usuario se debe ingresar la placa o nombre de la plataforma. Por consiguiente, se pulsa ‘Registrar’ y se diligencian los datos necesarios.

Luego de registrarse el usuario, se llega al formulario de registro, para este caso el formulario varía dependiendo del tipo de usuario, es decir, cambian algunos espacios requeridos para diligenciar, pero técnicamente el principio de ambos es el mismo.

En el formulario de registro lo primero que se solicita es capturar las imágenes de los cilindros a comercializar, es decir, el cilindro que se va a entregar y el cilindro que se va recibir. Seguidamente, se ingresan determinados datos necesarios para realizar un registro adecuado. Habiendo capturado las imágenes y diligenciado todos los espacios necesarios del formulario se puede pulsar, ‘Guardar

datos' o 'Guardar y retomar datos'. Cuando se pulsa 'Guardar datos', se registra la información y queda el formulario solo con los datos precargados desde el registro, cuando se pulsa 'Guardar y retomar datos' se registra la información y queda el formulario con los mismos datos, esto con el fin de aprovecharlos para un siguiente registro del mismo cliente minimizando el tiempo de operación. El formulario se diligencia la cantidad de veces que se debe entregar y recibir un cilindro al usuario final.

Después de haber realizado diferentes registros en el formulario se debe buscar la opción de 'Generar reporte'. Al pulsar 'Generar reporte' todos estos registros son almacenados en formato XML, buscando con esto facilitar la exportación de los registros hacia un servidor o computador principal.

También se puede elegir la opción 'Ver registros' en donde se pueden observar todos los registros realizados e ingresar a cada uno de ellos si se desea editar por cualquier motivo. También se puede elegir la opción 'Cerrar sesión' la cual solicita ingresar una clave y permite borrar todos los registros obtenidos hasta el momento.

Cabe destacar, que luego de haberse registrado una vez, para volver a ingresar a los formularios solo se debe ingresar el mismo nombre de identificación (placa o nombre de plataforma) y pulsar ingresar.

DESARROLLO EN LA PLATAFORMA 'ANDROID STUDIO'

La aplicación móvil es desarrolla en la plataforma Android Studio. Ahora se da lugar a explicar datos de interés para lograr este proceso

ESTRUCTURA DEL PROYECTO EN 'ANDROID STUDIO'

La estructura del proyecto básicamente se tiene en dos componentes que están localizadas en la división 'Android'. Las componentes son la carpeta 'app' y la sesión 'Gradle Scripts'. La carpeta 'app' contiene 'Scripts' del proyecto, es decir, todo aquello que se realiza, lo que se programa y lo que se ejecuta. La componente 'Gradle Scripts' contiene información de compilación, ahí se guarda la versión, se agregan librerías o dependencias del proyecto, y otras propiedades y configuraciones respecto a cómo se compila y como se llega al instalador final.

VERSIÓN Y DEPENDENCIAS

Son las configuraciones para indicar cuál es la versión actual de la aplicación y cuáles son las librerías que se están utilizando, para que en el momento de la compilación de 'Android Studios' se descarguen e instalen. En la estructura llamada 'build.gradle' va el nombre de la versión y las versiones mínimas y máximas de compatibilidad del Aplicativo móvil. Para este caso, la versión va desde la sdk 15 que es la versión de sistema operativo de Android 4.1 hasta la sdk 23 que es la versión de sistema operativo de Android 6.0.

En las dependencias se tienen librerías que permiten compatibilidad entre diferentes versiones. También, se manejan librerías para el soporte de la interfaz (botones y apariencia) y librerías para la serialización de objetos, es decir, para lograr enviarlos y recibirlos entre funciones, esta facilita mucho el envío y recepción de información entre contextos de la aplicación.

MANIFESTS

El 'manifests' es un archivo en el cual también se ingresan parámetros de configuración como lo son, permisos necesarios para determinados módulos de lectura y escritura de archivos del aplicativo móvil. También se agregan permisos para acceder a la cámara del dispositivo.

En el 'manifests' también se carga el ícono, tema y nombre del aplicativo que se va a visualizar en el sistema operativo. Se definen las actividades (son ventanas) que se van a emplear. Para este caso se tiene la actividad 'login' (actividad principal) y la actividad de la lista de registros.

VISTAS

Las 'vistas' son los componentes que tienen la función de visualizar elementos de la pantalla. Aquí se define la apariencia del aplicativo móvil, es decir, en esta parte se designan los botones, las listas, las imágenes, etc., pero sin funcionalidades, son únicamente los moldes en donde se intercalan los elementos de la interfaz.

Los archivos de las 'vistas' son archivos de texto en lenguaje de etiquetas XML. XML es el lenguaje utilizado para definir las 'vistas', estas están definidas en la carpeta 'app' - 'res' - 'layout'. En la **Figura 9** Relacionada a continuación se puede apreciar un Fragmento de diseño de vistas en 'Android Studio'.

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:showIn="@layout/activity_lista_registros"
    tools:context="com.aplicaciones.glp.ListaRegistrosActivityFragment">

    <ListView
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@+id/listView"
        android:layout_marginTop="5dp"
        android:layout_marginBottom="5dp"
        android:layout_marginLeft="5dp"
        android:layout_marginRight="5dp">

    </ListView>
</RelativeLayout>

```

Figura 9. Fragmento de diseño de vistas en ‘Android Studio’.

Ahora se explican las diferentes vistas diseñadas.

VISTA LOGIN

La vista del login es la primer vista que aparece cuando se abre la aplicación, para este caso tiene el título ‘GESTIÓN CILINDRO GLP’ Y tiene el campo de texto para ingresar la placa del vehículo o nombre de la plataforma y el espacio de selección para definir qué tipo de usuario se desea, es decir, tipo vehículo o plataforma. Además, se aprecia el botón de ingresar o botón de registro. Esta ‘vista’ se llama ‘fragment_principal.xml’. La ‘vista’ se puede apreciar como texto (código) o como un diseño preliminar gráfico. En la **Figura 10** Relacionada a continuación se puede apreciar de forma gráfica la vista Login.



Figura 10. VISTA' LOGIN

La estructura de texto (código) de esta vista tiene un elemento '`<LinearLayout...>`', este es el contenedor de todos los elementos, por ejemplo el elemento '`<TextView>`' contiene el texto 'Gestión Cilindros GLP'. También se destacan los siguientes: '`<Spinner>`' es el select o lista de desplegable que se tiene para definir el tipo de usuario, el '`<EditText>`' donde se agrega el nombre de usuario (palca o nombre de plataforma). Se tiene un contenedor adicional donde se designan los dos botones, es decir el botón de ingresar y el de registro.

Cada uno de los elementos tiene un ID, con este ID desde el área de controlador o archivos Java se puede hacer referencia al elemento, como por ejemplo obtener información.

Para definir los botones se maneja un atributo conocido como '`onClick`', donde se asigna el nombre de una función, localizada en una carpeta java y se va activar esa función cada vez que se le de click en el botón

VISTA REGISTRO

Al tocar el botón registrar de la vista principal (explicada anteriormente) se activa la función correspondiente ubicada en la carpeta 'java'- 'utils'- 'PrincipalActivity'. Esta función llamada 'eventoRegistro' la cual se encarga de tomar la referencia de una vista que se precarga llamada 'dialog_registro_content'. Esta vista (VISTA REGISTRO) es el formulario de registro que

contiene los campos de texto requeridos. Luego de llenar el registro y aceptar se valida que todos los espacios hayan sido diligenciados y se valida el tipo de usuario seleccionado, por consiguiente se guarda la información, de manera que pueda ser precargada. Dependiendo del tipo de usuario seleccionado, se carga una vista correspondiente al usuario elegido. En la **Figura 11** Relacionada a continuación se puede apreciar de forma gráfica la vista ‘Registro’.



Figura 11. VISTA REGISTRO

VISTA FORMULARIO

Las vistas de formularios son dos, una para cada tipo de usuario (‘Vehículo’ o ‘Plataforma’). Las vistas de formulario son necesarias para registrar los cilindros entregados y recibidos (capturando imágenes) con los respectivos datos de interés tanto del cliente como de los cilindros. El nombre de estas vistas son ‘fragment_formulario_vehiculo.xml’ y ‘fragment_formulario_plataforma.xml’ están definidas ‘java’-’res’-’layout’. El principio de diseño y desarrollo de ambas vistas es prácticamente el mismo, por lo tanto solo se va detallar ‘fragment_formulario_plataforma.xml’.

En la estructura de texto (código) de la vista plataforma se tienen dos campos dirigidos para la captura de las imágenes del cilindro entregado y del recibido. A estas imágenes se les adiciona un ID para lograr identificarlas y relacionarlas con los datos ingresados posteriormente.

En esta estructura se maneja un ‘<ScrollView>’ que permite extender la pantalla y manejar elementos no visibles. En este se definen los textos de entrada de cada uno de los elementos del

formulario. En la estructura también se relacionan elementos tipo ‘<Spinner>’ para manejar listas desplegables, en este caso son útiles para elegir la ciudad y la capacidad del cilindro. En la **Figura 12** y **Figura 13** Relacionada a continuación se puede apreciar de forma gráfica la vista formulario usuario tipo vehículo y la vista formulario usuario tipo plataforma respectivamente.



Figura 12. Vista formulario usuario tipo vehículo



Figura 13. Vista formulario usuario tipo plataforma.

VISTA DE REGISTROS

En la vista de registros se observa todos los registros que se han realizado durante la sesión. Esta ‘vista’ se encuentra definida en ‘java’-’res’-’layout’ y es llamada ‘fragment_lista_registros.xml’. En la estructura de texto (código) de esta ‘vista’ se tiene un elemento ‘ListView’ en donde el contenido se va obteniendo con elementos de ‘item_list_registro.xml’.

En ‘item_list_registro.xml’ se tiene una plantilla que permite visualizar cada uno de los registros guardados de la misma manera. Para este caso los registros se aprecian con dos imágenes contiguas (cilindro entregado y cilindro recibido) y la fecha y hora en que se obtuvo el registro.

VISTA EDITAR REGISTRO

La ‘vista’ editar registro es prácticamente la misma ‘vista’ de los formularios con la diferencia que esta vista se va a precargar con la información del registro que se quiere editar y solo tiene el botón ‘Guardar’. Cuando se cambia información del registro que se está editando y se pulsa ‘Guardar’ lo que sucede es que se sobrescribe la información en el mismo registro. Esta ‘vista’ se encuentra definida en ‘java’-’res’-’layout’ y es llamada ‘fragment_edit_form_vehiculo.xml’ o ‘fragment_edit_form_plataforma.xml’ dependiendo del tipo de usuario que se esté manejando. El principio de diseño y desarrollo de estas dos vistas es el mismo.

Se deben programar los controladores de cada una de las vistas de la aplicación.

CONTROLADORES

Los controladores son las estructuras de texto (código) en donde se definen funcionalidades de la aplicación móvil, es decir, eventos, operaciones con registros, entre otros. Estos controladores están definidos en archivos java ubicados ‘Android’-’app’-’java’-’com.applications.glp’. Aquí se ingresa toda la lógica de programación para controlar cada una de las vistas de la aplicación móvil.

CONTROLADOR LOGIN

El controlador de la vista login es la parte java donde se programa la lógica de los botones de ‘Ingresar’ y ‘Registrar’. Esta programación se desarrolla en el archivo ‘PrincipalActivity’ donde se manejan el método de actividad llamado ‘onCreate()’. Este método permite verificar si hay una sesión abierta. Si hay una sesión abierta se valida mediante la función ‘iniMain()’ y dependiendo

del tipo de usuario que se haya escogido al abrir la sesión, si es tipo 'vehículo' se abre la interfaz con la 'vista' del formulario de 'Vehículo' y si es tipo plataforma se abre una interfaz con la 'vista' del formulario de 'Plataforma'. En el caso de tenerse una sesión abierta se abre la interfaz de la 'vista' principal.

Cuando se abre la interfaz de la 'vista' principal se manejan dos eventos, el evento cuando se toca el botón 'Ingresar' y el evento cuando se toca el botón 'Registrar'. En la función evento registro se ingresan todos los condicionales los cuales se hacen con base al archivo XML de la 'vista' registro. Cuando se toca el botón aceptar se valida que los condicionales se cumplan y no haya espacios vacíos. Luego se verifica si el 'spinner' está con el tipo de usuario 'Vehículo' o 'Plataforma', esta información se almacena en una base de datos local del celular y luego se guarda en datos persistentes de la aplicación, de esta manera la información no se pierde aún si se cierra la aplicación. De esta manera se sabe si existe un usuario registrado para el próximo inicio de la aplicación y por ende, no iniciar con la 'vista' principal sino iniciar en la 'vista' formulario.

En la función evento Ingresar se valida que no hayan campos vacíos y se verifica el tipo de usuario y se inspecciona en la base de datos local para determinar si el usuario ya existe, se guarda en los datos persistentes y se muestra la interfaz de la 'vista' formulario. En el caso de no encontrarse el usuario y pulsar el botón ingresar se muestra el mensaje 'El usuario no está registrado'.

CONTROLADOR REGISTRO

Para controlar los registros se requieren dos controladores, uno para controlar el formulario de registro para usuarios tipo vehículo y el otro para controlar formulario de registro para usuarios tipo plataforma. El nombre para estos controladores es 'EditarFragmentVehiculo' (usuario tipo vehiculo) y 'EditarFragmentPlataforma' (usuario tipo plataforma). El principio de diseño y desarrollo de estos dos controladores es el mismo, por tal motivo para este caso vamos a detallar solo el controlador para usuario de tipo plataforma.

En la estructura de texto (código) del controlador se encuentra el método 'onCreateView' donde se crea la interfaz que se va a mostrar. Con base a la 'vista' de edición de plataforma ya detallada, se crea una nueva interfaz y se cargan cada de uno de los campos de texto y los desplegables de la vista edición de plataforma para que el usuario pueda editar la información y posteriormente guardarla. Se procede a verificar que cada uno de los campos no estén vacíos, en caso de que estén

vacíos al momento de pulsar guardar se genera un mensaje de alerta ‘Todos los campos son requeridos’, de lo contrario, se activa la función guardar. En la función guardar se toman cada uno de los campos e información existente y se guarda en la base de datos local. Las imágenes capturadas son almacenadas en un archivo que se guarda en la memoria interna del celular.

CONTROLADOR FORMULARIO

Para controlar los formularios de registros se requieren dos controladores, uno para controlar el formulario de registro para usuarios tipo vehículo y el otro para controlar formulario de registro para usuarios tipo plataforma. El nombre para estos controladores es 'FormularioFragmentVehiculo' (usuario tipo vehículo) y 'FormularioFragmentPlataforma' (usuario tipo plataforma). El principio de diseño y desarrollo de estos dos controladores es el mismo, por tal motivo para este caso vamos a detallar solo el controlador para usuario de tipo plataforma.

Se carga el formulario con los últimos datos ingresados siempre y cuando el usuario haya pulsado el botón ‘Guardar y Agregar’. Si solo se pulsa ‘Guardar’ se carga el formulario sin la última información ingresada. Esto se define en la función ‘Guardar’ de la estructura de texto (código). Cuando se pulsa el botón ‘Guardar’ y ‘Guardar y Agregar’ se valida que no haya espacios o elementos vacíos. Por consiguiente se inserta una nueva fila con la información en la tabla de reportes. Adicionalmente se inserta el ID como complemento al nombre de la imagen.

CONTROLADOR DE REGISTROS

El nombre para este controlador es ‘ListaRegistroActivityFragment’, y su funcionalidad radica en llenar por cada uno de los registros que se hallan en la base de datos una lista con la imagen del cilindro recibido, la imagen del cilindro entregado y la información de cuando se realizó el registro. En el método de actividad ‘onCreateView’ se toma la vista del XML y se especifica que esa va ser la vista del controlador.

En el método de actividad ‘onViewCreate’ se toman todos los registros que existen en la tabla de reportes y por cada uno de ellos se crean un arreglo, el cual tiene objetos con la información necesaria para representar un registro del ingreso de un cilindro en la base de datos local.

La clase ‘ListaRegistrosAdapter’ se encarga de crear una vista por cada elemento. Basándose en el objeto ‘Registro’ se logra determinar el tamaño de la lista y la forma de administrar.

Ahora se da lugar a explicar datos de interés sobre el proceso realizado en la segunda área de desarrollo

7.4 Diagrama del Funcionamiento del Procesamiento de Imágenes

En la **Figura 14** Relacionada a continuación se puede apreciar el diagrama del funcionamiento del procesamiento de imágenes

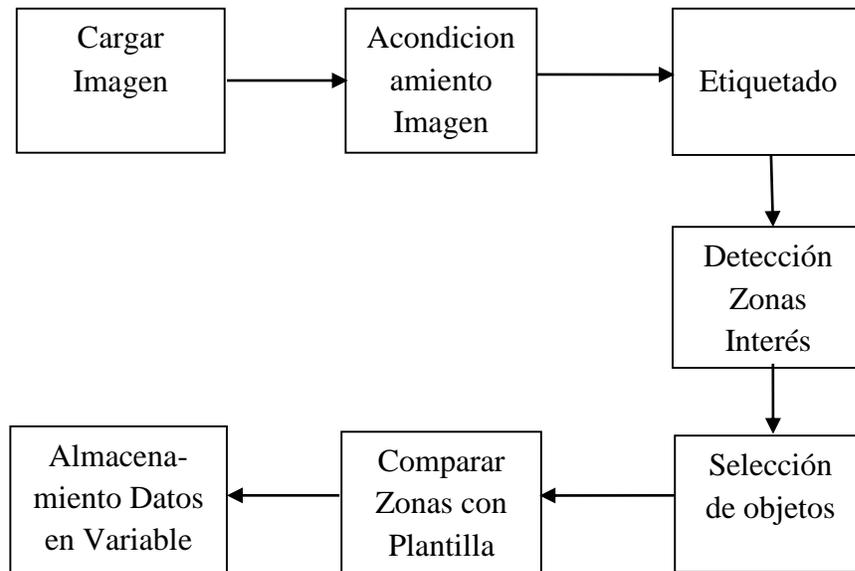


Figura 14. Diagrama del funcionamiento del procesamiento de imágenes.

7.4.1. Descripción Detallada del Funcionamiento del Procesamiento de Imágenes

En esta descripción se tratan temas más técnicos relacionados en con desarrollo de cada proceso resaltado en el diagrama de la **Figura 15**, para lo cual se utiliza el software de Octave.

CARGAR IMAGEN ORIGINAL Y ACONDICIONARLA

En el código de procesamiento de imágenes en primer lugar lo que se hace es cargar la imagen original, esta debe ser acondicionada de manera que logren extraer los caracteres necesarios de la misma. Para acondicionar la imagen, a esta se le deben hacer una serie de procesos, entre los cuales se destacan los siguientes: Adecuar tamaño, convertir a escala de grises, eliminación de ruido, dilatación y erosión, mejoramiento de bordes, binarización, eliminación de líneas no deseadas,

llenado de objetos, filtrado de objetos, etc. A continuación se da muestra gráfica de los procesos mencionados de la **Figura 15** a la **Figura 25**. Cabe destacar que el código de procesamiento de imágenes es desarrollado en el software 'Octave'.



Figura 15. Imagen original

Para cargar la imagen original en el código se utiliza la función 'imread' del paquete de librerías 'image' de 'octave'. Esta función es empleada bajo la estructura '`[IMG, MAP, ALPHA] = imread (FILENAME)`'. La función permite leer una imagen como una matriz para un nombre de archivo específico.



Figura 16. Imagen proceso adecuación del tamaño

Para adecuar el tamaño de la imagen original cargada previamente se utiliza la función 'imresize' del paquete de librerías 'image' de. Esta función es empleada bajo la estructura '`Function File: imresize (IM, [M N])`'. La función toma la imagen IM y retorna la misma con un tamaño M x N. Para este caso el valor especificado de N fueron las siglas 'NaN', ya que de este modo se determina

automáticamente un valor adecuado de forma que se adecúe el tamaño de la imagen preservando la misma relación del aspecto.



Figura 17. Imagen proceso conversión a escala de grises

Para pasar la imagen a escala de grises se utiliza la función 'rgb2gray' del paquete de librerías 'image' de 'octave'. Esta función es empleada bajo la estructura 'GRAY = rgb2gray (RGB)'. La función convierte una imagen RGB o mapa de color a escala de grises. Para este caso se trata de una imagen RGB, por lo tanto, la conversión a escala de grises es ponderada basado en los valores de luminancia.

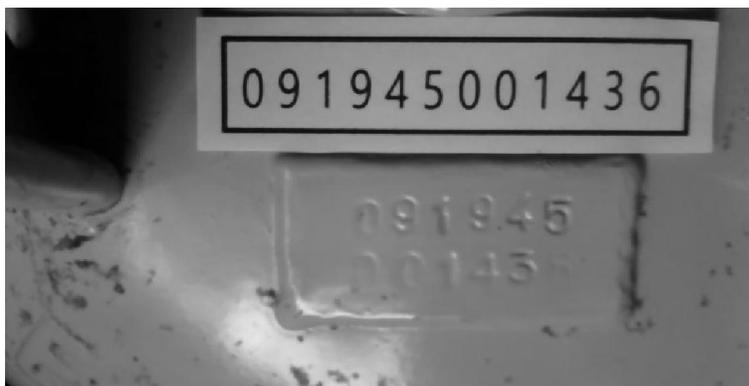


Figura 18. Imagen proceso eliminación de ruido

Para el primer filtrado que se realiza a la imagen se utiliza la función 'medfilt2' del paquete de librerías 'image'. Esta función es empleada bajo la estructura 'Function File: medfilt2 (A)'. La función sustituye elementos de A con la media de sus vecinos definidos por elementos de una matriz lógica de [M x N], que de no especificarse su valor, el valor predeterminado es de [3 x 3].



Figura 19. Imagen proceso de dilatación

El proceso de dilatación se interpreta como un crecimiento de píxeles. Para aplicar el proceso de dilatación a la imagen se emplean dos funciones del paquete de librerías 'image'. La función 'strel' y la función 'imdilate'. La función 'strel' es empleada bajo la estructura 'Function File: strel ("disk", RADIUS, N)'. Esta función crea un elemento estructural en forma de disco, es decir, define la forma y el tamaño de la vecindad del píxel que será analizado, para posteriormente alterar su valor de composición. El RADIUS debe ser un entero positivo y el valor de N es opcional. Fue elegida la forma 'disk' debido a que luego de varias pruebas presentó mayor efectividad respecto al resultado esperado. La función 'imdilate' es empleada bajo la estructura 'Function File: imdilate (IM, SE)'. La imagen IM debe ser una matriz numérica de cualquier dimensión y SE es el elemento estructural mencionado anteriormente. Lo que se obtiene es la imagen dilatada.



Figura 20. Imagen proceso de erosión

Al erosionar la imagen se ensanchan y realzan los objetos detectados en la imagen. El proceso de erosión de la imagen se ejecuta mediante la función 'imerode' del paquete de librerías 'image'. Esta función es empleada bajo la estructura 'Function File: imerode (IM, SE)'. La imagen IM debe ser una matriz numérica de cualquier dimensión y SE es el elemento estructural.



Figura 21. Imagen proceso mejoramiento de bordes

Para el mejoramiento de bordes se realizan tres procesos de tratamiento de imágenes con el fin de obtener el gradiente morfológico, hacer una conversión a clase doble y aplicar convolución. Los procesos mencionados se ejecutan mediante las funciones 'imsubtract', 'mat2gray' y 'conv2' respectivamente, del paquete de librerías 'image'. La función 'imsubtract' es empleada bajo la estructura 'OUT = imsubtract (A, B)' y permite obtener el gradiente morfológico (diferencia entre la dilatación y la erosión) efectuando la resta de B con A. Este proceso es muy útil debido a que se logra que cada valor de píxel indique la intensidad de contraste en la vecindad de ese píxel. La función 'mat2gray' es empleada bajo la estructura 'I = mat2gray (M)' y la matriz que devuelve es una imagen en escala de grises, de clase doble y en el rango de valores [0, 1]. La función 'conv2' es empleada bajo la estructura 'Built-in Function: conv2 (V1, M)' y se realiza convolución entre la imagen con conversión de clase doble y una matriz [1 1; 1 1], logrando así iluminar los bordes de los objetos de la imagen.



Figura 22. Imagen proceso de binarización

Para binarizar la imagen deseada se ejecutan dos funciones del paquete de librerías 'image', la función 'imadjust' y la función 'logical'. La función 'imadjust' es empleada bajo la estructura 'J = imadjust (... , GAMMA)' y permite ajustar la intensidad de colores para el rango específico. Tomando en este caso un GAMMA de 0.1, logrando así un mapeo de valores más altos (brillantes). La función 'logical' es empleada bajo la estructura 'Function: logical (X)' y convierte el objeto numérico X a un objeto de tipo lógico.

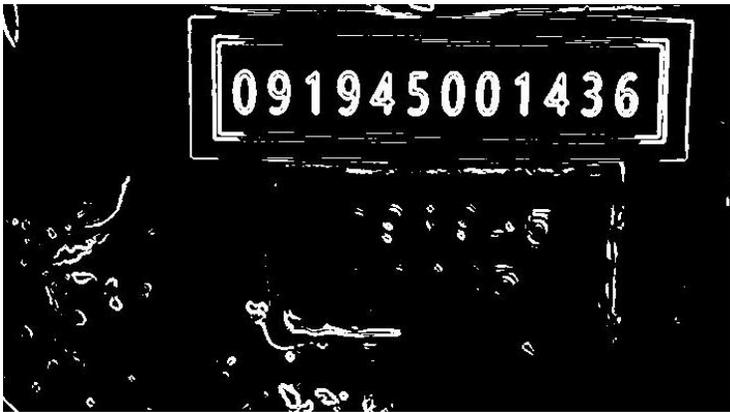


Figura 23. Imagen proceso eliminación de líneas no deseadas

La eliminación de líneas no deseadas se logra con dos procesos. En el primero se detectan algunas líneas que se quieren eliminar. El segundo se trata de quitar esas líneas detectadas de la imagen. Para detectar las líneas se emplea la función 'imerode' del paquete de librerías 'image'. Esta función es empleada bajo la estructura 'Function File: imerode (IM, SE)'. En este caso SE es un elemento estructural de tipo 'line'. Por consiguiente, se ejecuta la función 'imsubtract' del paquete de librerías

'image' que se emplea bajo la estructura 'OUT = imsubtract (A, B)'. B son las líneas detectadas, logrando así eliminar estas de la imagen A al sustraerlas de la misma.



Figura 24. Imagen proceso llenado de objetos

Para lograr el llenado de regiones u objetos de la imagen se ejecuta la función 'bwfill' del paquete de librerías 'image'. Esta función es empleada bajo la estructura '[BW2, IDX] = bwfill (BW1, "holes", N)'. De este modo se encuentran agujeros e interiores de objetos de la imagen BW1 y se llenan, esto con el fin de definir más ópticamente el objeto a comparar con la plantilla luego de ser detectado.

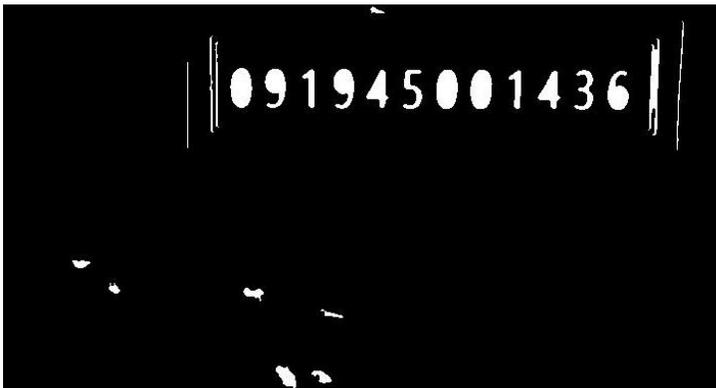


Figura 25. Imagen proceso filtrado de objetos

Cuando las fotografías se capturan basándose en ciertas especificaciones, por ejemplo, una distancia promedio de 15 cm, es posible eliminar algunos objetos no deseados empleando la función 'bwareaopen' del paquete de librerías 'image'. Esta función se emplea bajo la estructura 'Function File: bwareaopen (BW, LIM)'. En LIM se indica en número de píxeles mínimo que deben

contener los objetos de la imagen BW. Para este caso LIM tiene un valor de 50, es decir, en la imagen BW solo quedan objetos que están conformados por igual o más de 50 píxeles.

ETIQUETADO

Se realiza el proceso de etiquetado para lograr identificar objetos dentro de una imagen determinada. A continuación se da muestra de gráfica en la **Figura 26** de una imagen etiquetada:

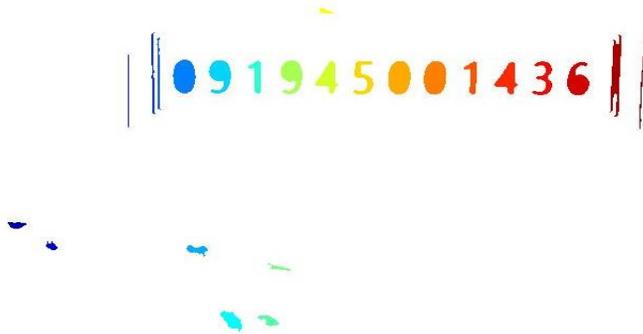


Figura 26. Imagen con proceso de etiquetación

Cuando se quieren identificar los objetos de una imagen es necesario emplear el proceso de etiquetado de objetos. Este se logra al ejecutar la función 'bwlabel' del paquete de librerías 'image'. Esta función se emplea bajo la estructura '[L, NUM] = bwlabel (BW)'. La función permite etiquetar objetos de la imagen binarizada BW en primer plano. La salida L es una matriz, donde 0 indica un píxel de fondo, 1 indica que el píxel pertenece al objeto número 1, 2 que el píxel pertenece objeto número 2, etc. El número total de objetos es NUM.

DETECCIÓN ZONAS DE INTERÉS

Se deben detectar zonas de interés buscando con esto minimizar el margen de error del procesamiento de imágenes. A continuación Se da muestra gráfica en la **Figura 27** de una imagen con detección de zonas de interés:

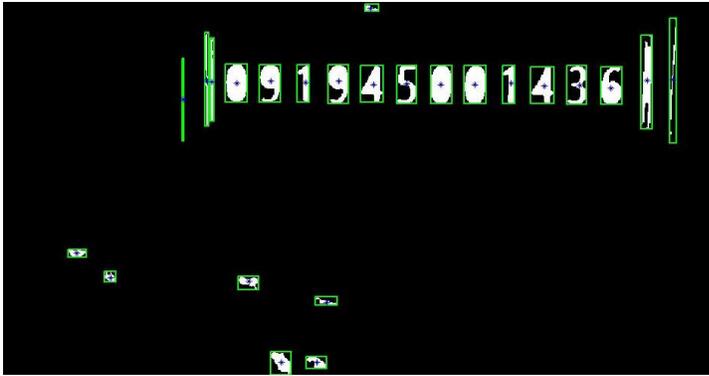


Figura 27. Imagen con detección de zonas de interés

Al tener los objetos de la imagen etiquetados se procede a detectar los mismos. En este orden de ideas, se ejecuta la función 'regionprops' del paquete de librerías 'image'. Esta función se emplea bajo la estructura 'PROPS = regionprops (BW, PROPERTIES,...)'. Es así como se logran calcular diferentes características de los objetos individualmente detectados en la imagen binaria BW. Las propiedades que se emplean son las 'BoundingBox' e 'Image'. De este modo se tiene como resultado los cuadros delimitadores de cada uno de los objetos etiquetados.

Para seleccionar todos los cuadros delimitadores se complementa el proceso al ejecutarse la función 'cat' bajo la estructura 'Function: cat (DIM, ARRAY1, ARRAY2, ..., ARRAYN)'. En este caso concatena los datos obtenidos de cada uno de los objetos etiquetados entregando una matriz con todos los elementos.

SELECCIÓN DE OBJETOS

Al identificar y obtener las zonas de interés de la imagen previamente tratada se deben seleccionar los objetos de interés para compararlos con los objetos de la plantilla. Para esto se realizan dos funciones enfocadas en la detección de los indicativos de los objetos de interés y tratamiento de los mismos.

Se desarrolla la función 'OB_IN' que determina la gama de indicativos de las cajas de interés. Esta función entrega un vector con los indicativos de los cuadros delimitadores de interés, es decir, entrega doce datos de identificación de objetos debido a que los objetos para comparar de la imagen son doce (12).

Para obtener los indicativos se emplean diferentes técnicas de procesamiento de imágenes y tratamiento de datos enfocadas a: Obtener el histograma del ancho de la dimensión de todos los cuadros delimitadores, obtener indicativos correspondientes a la frecuencia doce (12), ordenar las dimensiones del ancho, búsqueda de doce (12) caracteres (objetos) consecutivos, calcular el tamaño del contenedor, entregar datos indicadores de la caja o recuadro seleccionado y calificado como de interés.

La función desarrollada 'IND_REC' se encuentra anidada en la función 'OB_IN' para complementar el mismo proceso. Esta inicializa variables, entrega datos y coordenadas de los objetos consecutivos y permite la selección de objetos de interés.

COMPARACIÓN CON PLANTILLA

Para obtener el número de una imagen es fundamental hacer una comparación entre los objetos extraídos de la imagen y una plantilla. Los objetos extraídos deben cumplir con unas condiciones básicas, como por ejemplo, un tamaño específico, luego de esto, se hace una comparación con los objetos contenidos en la plantilla para así lograr identificar caracteres y almacenarlos apropiadamente. A continuación se da muestra gráfica en la **Figura 28** la imagen de un objeto obtenido previo a la comparación. Por consiguiente, se relaciona en la **Figura 29** la imagen de un objeto contenido en la plantilla.



Figura 28. Imagen objetos obtenidos antes de la comparación

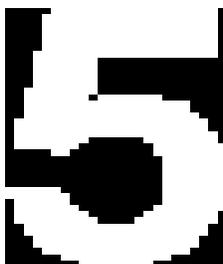


Figura 29. Imagen objetos contenidos en la plantilla

Siempre y cuando se obtengan índices de los objetos de interés, se procede a ejecutar una de las propiedades de 'regionprops' ('image') (empleada anteriormente) con el fin de cargar una matriz del mismo tamaño del cuadro delimitador que contiene los píxeles originales. Por consiguiente, se inicializa la variable 'NU_CIL' para almacenar la cadena de números detectados.

Luego se extrae la imagen binaria (objeto) con respecto a los índices obtenidos. Se desarrolla la función 'Leer_numero' para calcular la correlación entre la imagen de entrada (objeto) y las imágenes de diferentes caracteres contenidos en la plantilla, entregando el dato para almacenar en la variable.

El dato obtenido es almacenado en una variable específica para luego poder ser enviado a la base de datos. Ejemplo:

```
Num_Cil = '121945672398'
```

De manera opcional se puede verificar que el dato obtenido sea correcto. Para esto se ejecutan dos funciones esenciales, la función 'fopen' que permite abrir archivos de bajo nivel y la función 'system' que permite dar muestra del dato en un archivo de bloc de notas.

Ahora se da lugar a explicar datos de interés sobre el proceso realizado en la tercera área de desarrollo

7.5 Descripción Funcionamiento de la Base de Datos

La base de datos fue diseñada en el programa Excel de Microsoft. Esta base de datos recibe la información organizada enviada por el código de procesamiento de datos e imágenes. A continuación se puede apreciar en la **Tabla 1** una pequeña parte de la base de datos con algunos datos almacenados:

| | | | | | |
|-----------------------|----------|-----------|------|--------------------|-----------------|
| BASE DE DATOS | CÓDIGO: | | | | |
| GESTIÓN CILINDROS GLP | VERSIÓN: | | | | |
| COMPAÑÍA: | FECHA: | | | | |
| N | CIUDAD | FECHA | HORA | VEHÍCULO | NOMBRE_CLIENTE |
| 8 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 9 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 10 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 11 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 12 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 13 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |

Tabla 1. Resultado obtenido en la base de datos

La base de datos desarrollada en Excel de Microsoft contiene una serie de fórmulas que ayudan a complementar la información y permite tener una trazabilidad apropiada de los cilindros a comercializar. La base de datos en Excel está compuesta por dos hojas de cálculo, la primera es la encargada de recibir la información o datos entregados por el sistema de procesamiento de imágenes y datos, y la segunda adopta esta misma información y la complementa con datos de interés para las compañías de envasado y distribución de GLP. Estos datos se obtienen al ejecutar unas fórmulas que se encuentran programadas en la base de datos. Es necesario que la segunda hoja de cálculo sea manipulada por el usuario indicado para que se ejecuten las fórmulas y se obtenga el resultado esperado. A continuación se da muestra en la **Tabla 2** y **Tabla 3** de las hojas de cálculo de la base de datos (Solo una parte de todo el contenido).

| | | | | | |
|-------------------|----------|-----------|------|--------------------|-----------------|
| BASE DE DATOS | CÓDIGO: | | | | |
| GESTIÓN CILINDROS | VERSIÓN: | | | | |
| COMPAÑÍA: | FECHA: | | | | |
| N | CIUDAD | FECHA | HORA | VEHÍCULO | NOMBRE_CLIENTE |
| 8 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 9 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 10 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |
| 11 | Pereira | 25/3/2016 | 17:1 | plataforma pereira | simon castañeda |

Tabla 2. Hoja de cálculo uno (1) de la base de datos

| PESO_REAL | ERROR | ESTADO | VALOR(\$) | N_RECARGA | PESO_ESPERADO | C_C_E_ERROR |
|-----------|--------|---------|-----------|-----------|---------------|-------------|
| 28,5 | 0,0049 | BUENO | 0 | 0 | 28,50 | 0,074 |
| 28,5 | 0,0049 | BUENO | 0 | 0 | 28,50 | 0,074 |
| 28,5 | 0,0049 | BUENO | 0 | 0 | 28,50 | 0,074 |
| 28,5 | 0,0049 | BUENO | 0 | 0 | 28,50 | 0,074 |
| 28,5 | 0,0049 | BUENO | 0 | 0 | 28,50 | 0,074 |
| 33,5 | 0,0049 | REGULAR | 0 | 0 | 33,50 | 0,088 |

Tabla 3. Hoja de cálculo dos (2) de la base de datos

En la hoja de cálculo dos (2) de la base de datos que se puede apreciar en la **Tabla 3** cabe resaltar que las fórmulas están programadas en las últimas columnas de la hoja.

Después de todo el proceso desarrollado, en el siguiente capítulo se puede observar el producto obtenido y todos los datos que de interés que se obtienen a la largo del proceso.

8. RESULTADOS

8.1. Sistema Obtenido

Se obtuvo un sistema que permite dar cumplimiento por parte de las compañías de envasado y distribución de GLP en un alto porcentaje a la normatividad vigente (mostrada puntualmente en la sesión de motivación) impuesta por diferentes entidades del estado.

Es importante aclarar que para que sistema diseñado funcione de la mejor manera se deben contar con los cilindros GLP etiquetados de una forma apropiada y con dispositivos que cumplan con unas especificaciones técnicas mínimas.

En el siguiente esquema de la **Figura 31** se busca mostrar el modelo de operación y funcionamiento del sistema obtenido por parte de las compañías. Por consiguiente, se da una explicación de manera textual:

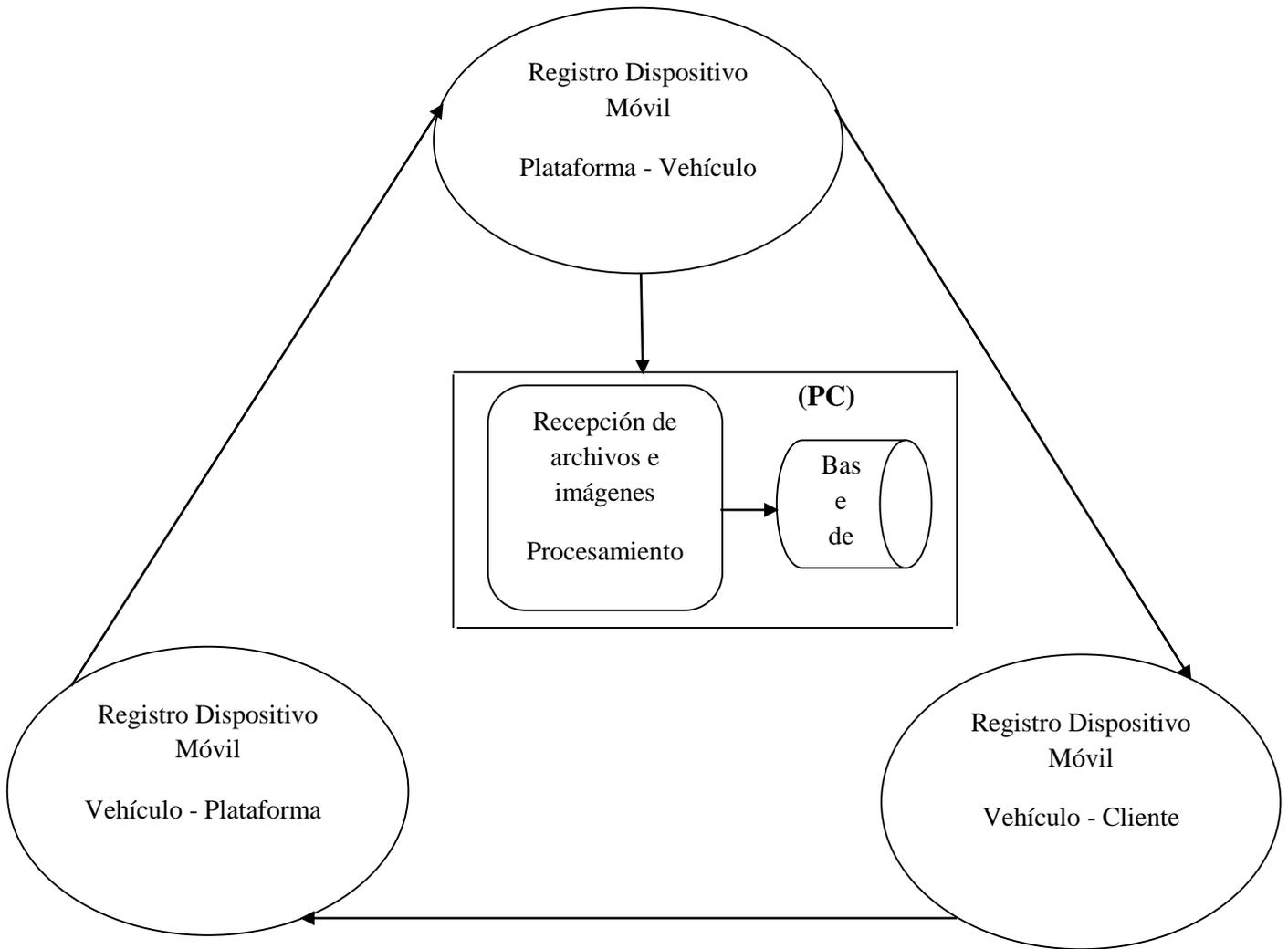


Figura 30. Zonas de transacciones de cilindros GLP (registro y procesamiento de datos)

En el esquema de la **Figura 30** se muestra el modelo de operación a seguir por parte de las compañías del sistema desarrollado, en donde la primera transacción de cilindros se da en la plataforma de envasamiento del Gas Licuado del Petróleo (GLP), en este punto se envasan los cilindros y, por consiguiente, se le entregan al personal de distribución. En la plataforma debe haber una persona encargada de registrar los cilindros llenos que van a ser entregados al distribuidor y aquellos que el anteriormente mencionado trae en su camión vacíos, Para ello debe hacer uso de un dispositivo móvil el cual debe tener cargada la aplicación 'GESTIÓN GLP. La persona encargada captura las imágenes del cilindro a entregar y del cilindro a recibir, Por consiguiente, ingresa los datos requeridos y guarda la información en el dispositivo. Es importante destacar que

la App móvil debe estar en el usuario tipo plataforma, ya que este requiere que sean ingresados unos datos específicos indispensables para alimentar la base de datos de la manera correcta.

La segunda transacción de cilindros se da cuando el distribuidor va hacer entrega de un determinado cilindro a un cliente, En este punto, El distribuidor debe registrar el cilindro lleno el cual va a entregar y el cilindro vacío que recibe de parte del cliente, para lo anterior, debe hacer uso del dispositivo móvil el cual debe tener cargada la App móvil 'GESTIÓN GLP'. El distribuidor captura las imágenes del cilindro a entregar y del cilindro a recibir, por consiguiente, ingresa los datos requeridos y guarda la información en el dispositivo. Es importante destacar que la App móvil debe estar en el usuario tipo vehículo, ya que este requiere que sean ingresados unos datos específicos indispensables para alimentar la base de datos de la manera correcta.

La tercera transacción de cilindros se da cuando el distribuidor ingresa a la planta de envasamiento para recargar los cilindros vacíos, en este punto, nuevamente la persona encargada en plataforma realiza el registro del mismo modo como se explicó anteriormente.

Luego los registros obtenidos en los dispositivos móviles tanto de los usuarios tipo plataforma como vehículos, deben ser descargados por una persona designada al servidor o computador principal, luego ejecuta el código de procesamiento de imágenes y organización de datos y toda la información es dirigida y almacenada en la base de datos. Por consiguiente, la persona designada ejecuta las fórmulas de la hoja de cálculo 2 de la base de datos para complementar totalmente la información de la misma. Es indispensable que en este punto se borre la información contenida en los dispositivos móviles para evitar posibles confusiones futuras.

Todo el proceso que se describió anteriormente está previsto para que se ejecute en un día laboral.

8.1.2. Pruebas y resultados del sistema obtenido

Las pruebas realizadas están enfocadas en dar muestra de que tan efectivo es el sistema desarrollado. Las capturas de imágenes se realizan con una distancia promedio de 15 cm y estas se harán en siete (7) escenarios diferentes. Para el escenario uno (1) y el escenario siete (7) son capturadas un total de 100 imágenes. Para los demás escenarios se capturan un total de 50 imágenes.

El escenario uno (1) simula el proceso de entrega y recibido de cilindros GLP por parte de la plataforma de envasado hacia los distribuidores. Del escenario dos (2) al escenario seis (6) se

simula el proceso de entrega y recibido de cilindros GLP por parte de los distribuidores hacia los clientes. El escenario siete (7) simula el proceso de entrega y recibido de cilindros GLP por parte de los distribuidores hacia la plataforma de envasado.

Cabe destacar, que para cada escenario se tienen diferentes entornos, es decir, diferentes tonos de iluminación.

Los resultados para cada escenario se mostrarán en una tabla en donde se relacionan algunas imágenes capturadas en la práctica y los números obtenidos luego de procesarlas con el código de procesamiento de imágenes. Por último, se muestra el resultado obtenido en la base de datos luego de ejecutar el código completo de procesamiento de imágenes y datos. De la **Tabla 4** a la **Tabla 10** Relacionadas a continuación se pueden apreciar algunas imágenes capturadas para los diferentes escenarios descritos con anterioridad.

Escenario uno (1)

| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '187191164824' | '311945765823' |
|  |  | '667191164824' | '129915366811' |
|  |  | '987191164829' | '829915366811' |

Tabla 4. Algunas imágenes capturadas escenario uno (1) entregado - recibido

Escenario dos (2)

| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '187191164824' | '101995765829' |
|  |  | '667191164824' | '701356765823' |
|  |  | '987191164829' | '801955765828' |

Tabla 5. Algunas imágenes capturadas escenario dos (2) entregado - recibido

Escenario tres (3)

| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '087141164829' | '101944765823' |
|  |  | '191945301436' | '401945765823' |
|  |  | '771945001436' | '501945765825' |

Tabla 6. Algunas imágenes capturadas escenario tres (3) entregado - recibido

Escenario cuatro (4)

| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '091848001436' | '101975765827' |
|  |  | '091965001439' | '101849765823' |
|  |  | '181945125893' | '001045765821' |

Tabla 7. Algunas imágenes capturadas escenario cuatro (4) entregado - recibido

Escenario cinco (5)

| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '088945125593' | '111747775833' |
|  |  | '081245225893' | '111955775835' |
|  |  | '081905125890' | '171945765823' |

Tabla 8. Algunas imágenes capturadas escenario cinco (5) entregado - recibido

Escenario seis (6)

| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '229915366812' | '701965765826' |
|  |  | '121915366111' | '105945765523' |
|  |  | '129415466811' | '211945775833' |

Tabla 9. Algunas imágenes capturadas escenario seis (6) entregado - recibido

Escenario siete (7)

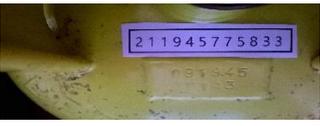
| Imagen cilindro entregado (C_E) | Imagen cilindro recibido (C_R) | Número obtenido C_E | Número obtenido C_R |
|---|---|---------------------|---------------------|
|  |  | '089191164924' | '211945775833' |
|  |  | '087393164824' | '911945775839' |
|  |  | '087101164820' | '113945775333' |

Tabla 10. Algunas imágenes capturadas escenario siete (7) entregado – recibido

Las etiquetas empleadas para la prueba del proyecto son un prototipo que cumple con las condiciones necesarias para la ejecución del proceso. Sin embargo, se recomienda para una puesta en práctica empresarial que cumplan con las mismas características pero que estas sean de tipo

adhesivo, con un material resistente al agua y que refleje la mínima luz posible. En la **Figura 31** Relacionada a continuación se puede apreciar el modelo de la etiqueta a emplear.



Figura 31. Modelo etiqueta a emplear

Las etiquetas deben cumplir las siguientes especificaciones: Debe tener 12 dígitos escritos con un tipo de letra ‘Gadugi’ tamaño ‘30’ de ‘Microsoft Word’. Debe tener una fuente de espaciamiento del 80%, dejando entre cada dígito un espacio. Los 12 dígitos deben estar dentro de un recuadro resaltado. La etiqueta debe ser de color blanco y tanto los dígitos como el recuadro deben ser de color negro. Debe ser adhesiva, de material resistente al agua y poco reflectivo.

Luego de las pruebas se puede determinar qué siempre y cuando se cumplan con unas condiciones para capturar la imagen, el sistema presenta un margen de error aproximado de 8 %. Para este caso, se capturaron 150 imágenes de las cuales 12 presentaron problemas en el momento de detectar el número.

Las imágenes se deben capturar a una distancia aproximada de 15 cm, lo más nivelada posible y evitando la iluminación excesiva o, por el contrario, poca iluminación que afecta la apreciación visual de la etiqueta. Se debe capturar en modo detenido tanto el cilindro con su etiqueta de identificación como el dispositivo móvil empleado para la toma fotográfica.

A continuación se da muestra en la **Tabla 11** de algunas imágenes que presentaron fallas con una breve descripción del posible motivo:

| Imagen Capturada | Descripción |
|---|--|
|  | <p>Para algunos números detectados el código entrega el dato incorrecto.</p> |

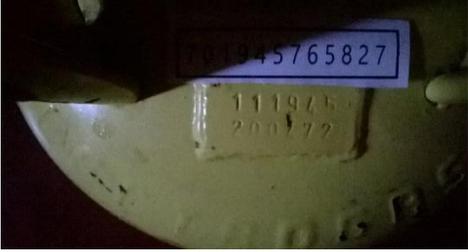
| | |
|---|---|
|  | <p>No se obtiene ningún dato debido a la poca iluminación al lado izquierdo de la imagen.</p> |
|  | <p>No se obtiene ningún dato debido al grado de inclinación de la imagen.</p> |
|  | <p>Los números entregados por el código de procesamiento son en su gran mayoría incorrectos debido a poca calidad de la imagen.</p> |
|  | <p>No se obtiene ningún dato debido a que es una imagen demasiado movida.</p> |
|  | <p>Se obtienen datos incorrectos debido a que se trata de una imagen muy iluminada.</p> |

Tabla 11. Imágenes que presentan fallas para detectar sus números

La base de datos (solo una parte del contenido) obtenida está relacionada en la **Tabla 12** descrita a continuación.

| CILINDRO_RECIBIDO | CAP_CIL_REC | CILINDRO_ENTREGADO | CAP_CIL_ENT | TARA_CIL_ENT | PESO_REAL |
|-------------------|-------------|--------------------|-------------|--------------|-----------|
| 311945765823 | 15 | 187191164824 | 15 | 13,5 | 28,5 |
| 129915366811 | 15 | 667191164824 | 15 | 13,5 | 28,5 |
| 829915366811 | 15 | 987191164829 | 15 | 13,5 | 28,5 |
| 629915366816 | 15 | 83191164324 | 15 | 13,5 | 28,5 |
| 126915366611 | 15 | 87691664824 | 15 | 13,5 | 28,5 |
| 129019366811 | 18 | 87141164829 | 18 | 15,5 | 33,5 |
| 129975366817 | 18 | 191945301436 | 18 | 15,5 | 33,5 |

Tabla 12. Base de datos

8.2. Conclusiones

Se procede a dar muestra al desenlace del sistema obtenido:

8.2.1. Logros y Reflexiones

* Con el código de procesamiento de imágenes se logra un amplio porcentaje de efectividad al obtener el número de identificación de los cilindros GLP, se considera que la dificultad para obtener una efectividad cercana al 100 % radica en que la manipulación del dispositivo móvil la realiza el factor humano, motivo por el cual, es muy difícil regular la forma en que se deben tomar las fotografías. También hay que destacar, que las condiciones climáticas son demasiado variables, lo cual afecta el entorno y ambiente apropiado para la captura de las mismas. A continuación se da muestra de algunas formas inadecuadas de capturar las imágenes.

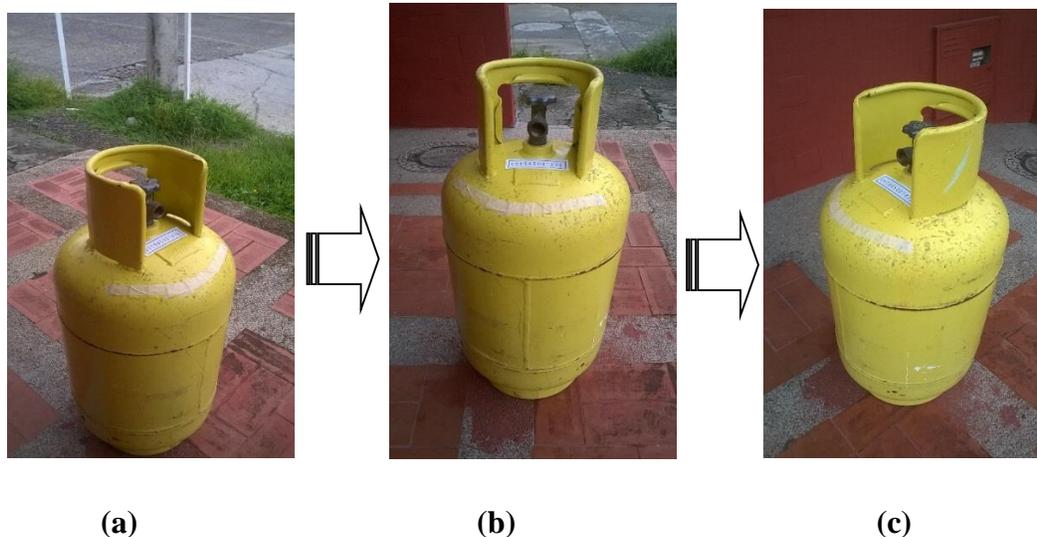


Figura 32. Formas no adecuada de capturar las imágenes.

* Almacenar la información en la base de datos fue un logro muy acertado para lograr que las compañías de envasado y distribución de Gas Licuado de Petróleo (GLP) puedan tener un registro y control de la comercialización de cilindros efectuada, sin embargo, se recomienda estar haciendo copias de seguridad de esta base de datos para evitar que diversos inconvenientes por factores externos que se pudiesen presentar en el servidor o computador principal puedan afectar la información. Hay que destacar que la base de datos puede ser modificada por el usuario, por tal motivo, debe ser manipulada por una persona fiable y con una buena ética laboral.

* Con el código de procesamiento de imágenes se logra Extraer los caracteres o números de la imagen con un alto porcentaje de efectividad, sin embargo, en la primeras pruebas prácticas realizadas no se obtenían los resultados esperados, el motivo de lo anterior se debía principalmente a dos factores, el primero radica en que el carácter extraído de la imagen no estaba conformado plenamente, es decir, faltaba aplicar el proceso de relleno para su conformación, el segundo factor radica en que no se tenían los caracteres de comparación apropiados, es decir, los objetos de comparación almacenados en la plantilla no cumplían con las condiciones necesarias, cuando se aplicaba el proceso de relleno a los caracteres detectados estos cambian notoriamente y en la plantilla no se contaba con el mismo modelo. Por ejemplo, el número seis (6) quedaba con su círculo inferior totalmente lleno.

* La interoperabilidad entre los datos ingresados en la App móvil y la ejecución de las fórmulas en la base de datos es correcta. Es de destacar lo siguiente: Cuando se está diseñando tanto la App móvil como la base de datos, se debe tener muy presente el tipo de indicador decimal que se va a utilizar, ya que cuando difieren entre las dos plataformas la interoperabilidad entre ellas no es correcta. Por ejemplo, si en la App móvil se maneja un indicador decimal tipo coma (,) y en la base de datos se maneja un indicador decimal tipo punto (.) no es posible la ejecución de fórmulas matemáticas en la base de datos con datos entregados por la App móvil.

* Cuando se ejecutan procesos de eliminación de ruido y filtrado de objetos no deseados en procesamiento de imágenes, es fundamental definir una distancia promedio para la captura de la imagen entre el dispositivo móvil y el objeto al cual se le va a tomar la fotografía. Cuando esta distancia varía se corre con el riesgo de eliminar objetos deseados cuando se aplican los procesos, es decir, eliminar posibles caracteres de interés antes de realizar la comparación con los objetos de la plantilla.

* Este sistema de procesamiento de imágenes presenta algunas fallas cuando la imagen no está lo suficientemente nivelada, generalmente el código opera pero es mayor la probabilidad de que entregue algunos datos de caracteres incorrectos, esto se debe muy probablemente a no tener una plantilla con todos los modelos de los caracteres en diferentes ángulos. Para este caso, se optó por enmarcar las etiquetas con un recuadro ya que este ayuda a mejorar la percepción de la imagen respecto a su ubicación nivelada cuando se va hacer la captura de la misma. Por consiguiente, se implementó el código con un proceso que permite eliminar algunas líneas no deseadas.

* Para recorrer el Archivo XML en el código de procesamiento y lograr obtener la información entregada por la App móvil se emplearon diferentes técnicas, luego de una serie de pruebas, se determinó que una de las más efectivas para este caso era recorrer la información del archivo manejando e identificando las posiciones de interés e inmediatamente después, enviar la información a la base de datos, este proceso se hace por registro detectado en el XML, es decir si el XML contiene 10 registros, el proceso se repite 10 veces.

8.2.2 Inconvenientes Presentados en el Desarrollo del proyecto

* Instalar 'Android Studio'

‘Android Studio’ Es un software o plataforma muy completa que permite desarrollar aplicaciones móviles, para su proceso de instalación se requieren de algunos tips fundamentales para hacerlo de manera correcta. Es de destacar, que es un software de gran magnitud computacionalmente hablando, es decir, se requiere del tiempo suficiente y del espacio necesario en el disco Duro de la computadora. En el momento de omitir algún paso en la instalación, este no funcionara correctamente y requiere ser desinstalado, lo cual es bastante demorado también.

* Identificar las fotografías y etiquetarlas de manera estratégica en la App móvil

Los datos ingresados a la App móvil deben ser complementados con las capturas de imágenes de los cilindros que son comercializados, es decir, la imagen del cilindro que se le entrega al cliente y la imagen del cilindro que se recibe del mismo. Estas imágenes se almacenan todas en una carpeta específica en donde se hace indispensable identificarlas de manera que se puedan relacionar con los datos correctos de la App móvil. Se hizo necesario crear un ‘ID’ o identificador que fuera incrementando cuando se realizará un registro en la App móvil, este ‘ID’ se utiliza para

complementar la etiqueta o nombre de las diferentes imágenes capturadas. Por ejemplo, si el 'ID' es cuatro (4) las imágenes capturadas del cilindro recibido y del cilindro entregado serán etiquetadas como CR_4 Y CE_4 respectivamente.

* Obtener el archivo XML

Para generar el archivo XML desde la aplicación, primero se pensó en cómo organizar los datos de forma que se cargaran uno a uno en las diferentes líneas del formato. Para ello todos los datos se guardaron en forma de 'String', de esta forma solo falta generar un ciclo que valida si hay cambio en cada línea, y ciclo a ciclo se escribe en cada uno de los campos del formato XML.

* Lograr instalar los paquetes necesarios en el software 'Octave'

Para el desarrollo de este sistema es necesario desarrollar un código de procesamiento de imágenes, para desarrollar y ejecutar este código se utiliza el software libre 'Octave'. En 'Octave' se deben instalar paquetes o librerías que facilitan la ejecución de funciones para desarrollar diferentes procesos. 'Octave' también se utiliza como sistema de concatenación entre las dos áreas de desarrollo principales. Para la instalación de los paquetes se presentan algunos problemas los cuales no se hacen fáciles darles solución debido a la falta de información que existe al respecto. Luego de varios ensayos, se logran identificar los pasos a seguir y los comandos que se deben emplear para lograr la instalación correcta.

* Ejecutar la función 'imresize' en 'Octave' con imágenes binarizadas

La función 'imresize' de 'Octave' no permite que se ejecute con imágenes las cuales hayan pasado por el proceso de binarización. Esta función se diseña nuevamente con el nombre 'mi_imresize', la cual permite acondicionar el tamaño de la imagen indiferente de los procesos que se le hayan aplicado previamente.

* Recorrer la imagen para detectar objetos

En primer lugar, se diseña una función que fue llamada 'líneas', esta función recorría la imagen logrando identificar líneas o renglones que tuvieran objetos, para que funcionara el procedimiento anterior es necesario limitar la cantidad de objetos detectados por línea, es decir, para este caso debían ser 12 objetos consecutivos. Esto conlleva a un problema mayor debido a que si en la línea

detectada existen objetos de interés pero a su alrededor se encuentran objetos no deseados, no se pasa al proceso de comparación con la plantilla, lo cual no permitía obtener el resultado deseado. Por consiguiente, se desarrolla una función que recorra la imagen y de los objetos detectados entrega diferentes índices y posiciones de las mismas, logrando así, compararlos con la plantilla y obtener el resultado deseado.

* Ejecutar las funciones del paquete io de ‘Octave’ necesarias para cargar el archivo XML y enviar información a la base de datos

Con las funciones ‘xlswrite’, ‘xlsread’, ‘xmlwrite’ y ‘xmlread’ contenidas en el paquete io de ‘Octave’ se debe tener cierto cuidado para su ejecución, esto debido a que no existen mucha información y ejemplos que expliquen la forma de emplearlas. Se debe ser muy específico con ellas y tener conceptos claros sobre los archivos de tipo xls y xml. Por ejemplo, las funciones de tipo write y read no manejan los mismos argumentos.

8.3. Trabajo Futuro

* Emplear en el código de procesamiento de imágenes un algoritmo de rotación que nivele la imagen para no hacer tan limitada la forma de tomar la misma.

. * Emplear algoritmos más potentes de procesamiento en el código de manera que se pueda disminuir el margen de error cuando se tienen fotografías demasiado iluminadas.

* Lograr que la App móvil entregue sugerencias de personalización al usuario respecto a los datos que muy posiblemente ya ha ingresado con anterioridad.

* Hacer un cuadro delimitador en la App móvil que permita capturar la imagen más nivelada.

* Conseguir inversores para contratar un servidor web y subir la aplicación móvil.

* Optimizar el código para mejorar la velocidad de procesamiento.

* Montar la base de datos en un software libre.

BIBLIOGRAFÍA

- [1] CREG, "Resolución n° 023. Por la cual se establece el reglamento de distribución minorista de gas licuado de petróleo", *Diario Oficial n°46930 de Colombia*, 2008.
- [2] WLPGA, "LPG Exceptional Energy - Historia del GLP", *WLPGA [online]* 2015.
- [3] CREG, "Historia del GLP en Colombia", *CREG [online]* 2016
- [4] A.M. Redacción, "El negocio de la Apps móviles ¿Panacea o rentable solo para unos pocos?", *PuroMarketing [online]* 2014.
- [5] E.R. DOUGHERTY, "Electronic imaging technology" Bellingham USA, *The international society for optical engineering*, ISBN- 0-8194-3037-4, 1999.
- [6] A. Bovid, "Handbook of image and video processing. 2 ed.", Austin USA, *Elsevier academic press*, ISBN- 0-12-119792-1, 2010.
- [7] A.H. LUIS, "Programa original de la asignatura: Programación orientada a objetos en Java", *Programa de informática, Ciudad de México: Universidad Nacional Autónoma de México*, 2016.
- [8] R.KLETTE, *Concise computer vision: an introduction into theory and algorithms*. Springer Science & Business Media, 2014.
- [9] T. ACHARYA and A. K. RAY, *Image processing: principles and applications*. John Wiley & Sons, 2005.
- [10] C. J. DATE and Others, *Introducción a los sistemas de base de datos*. Pearson Educación, 2001.
- [11] J. PASTOR, "Desarrollo de aplicaciones móviles", así está el mercado", *Xataka Móvil [online]*, 2014.

