

## Aplicación móvil para control de maduración de frutillas utilizando algoritmo de procesamiento de imágenes de MATLAB

### [ Mobile application for strawberry maturation control using MATLAB image processing algorithm ]

*Jimmy Sornoza Moreira, Fabricio Medina Palacios, Melina Fuentes Marmolejo, Kevin Labre Hidalgo, and Alfonso Vaca Chóez*

Facultad de Ciencias Matemáticas y Físicas, Universidad de Guayaquil, Guayaquil, Ecuador

Copyright © 2020 ISSR Journals. This is an open access article distributed under the *Creative Commons Attribution License*, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**ABSTRACT:** Taking a strawberry harvesting process is tedious and requires a lot of time for dedication, because small periods of time must be followed for the strawberry harvesting process and these times go hand in hand with revisions of the fruit in its entirety. Situation that falls on the need for a system that helps the collector as the main entity to keep track of said process, that is, software that can differentiate and select between colors and characteristics to be able to collect the strawberry. This problem leads to the development of a system that complies by means of algorithm the whole process to choose the strawberry at its exact point of maturation. The implementation of the algorithm contains a methodology based on a sequence of subprocesses such as: Pre-Processed, Segmentation and Extraction of Characteristics to be able to decide by means of an algorithm that strawberry is minutes for its subsequent collection. The elaboration of the system was used in MATLAB because it has more familiar functions to image processing and many projects derived in this area are made or carried out in this tool. It will have data collection in MySQL and finally, the data will be displayed in an Android application for the internal interested party.

**KEYWORDS:** Pre-Processing, Segmentation, Extraction, Android, DATABASE.

**RESUMEN:** Llevar un proceso de recolección de frutilla resulta tedioso y requiere mucho tiempo de dedicación, debido a que se debe seguir pequeños períodos de tiempo para el proceso de recolección de la frutilla y estos tiempos van de la mano con revisiones de la fruta en su totalidad. Situación que recae en la necesidad de un sistema que ayude al recolector como entidad principal a llevar un seguimiento del dicho proceso, es decir, un software que pueda diferenciar y seleccionar entre colores y características para poder recolectar la frutilla. Esta problemática conduce al desarrollo de un sistema que cumpla por medio de algoritmo todo el proceso para escoger a la frutilla en su punto exacto de maduración. La implementación del algoritmo contiene una metodología basada en una secuencia de subprocessos como: Pre-Procesado, Segmentación y Extracción de Características para poder decidir mediante un algoritmo que frutilla está lista para su posterior recolección. La elaboración del sistema fue empleado en MATLAB debido a que posee funciones más familiares al procesamiento de imágenes y muchos proyectos derivados en este tema son hechos o realizados en esta herramienta. Contará con la recopilación de datos en MySQL y por último, los datos se mostrarán en una aplicación Android para la parte interesada interna.

**PALABRAS CLAVE:** Pre-Procesado, Segmentación, Extracción, Android, DATABASE.

## 1 INTRODUCCIÓN

La frutilla es uno de los frutos que su principal característica es tener una tasa respiratoria sumamente alta, lo que hace que no se conserve a largo tiempo, más bien, posee una efímera vida de almacenamiento. Todo esto a causa de su fina epidermis, mayor proporción de agua y gran metabolismo, volviéndola un punto apetecido para los microorganismos o el entorno en que se encuentre.

Frente al avance tecnológico, se espera cubrir una gran cantidad de necesidades con respecto al área de agricultura, poder disminuir tiempo y costos operacionales y, maximizar el ingreso; un sistema que permita darle seguimiento a los frutos o algún otro cultivo. Captar por imágenes si el cultivo está listo para el consumo o, por el contrario, separar de los demás para evitar problemas e incluso grandes pérdidas. Cada segmento especificará el proceso desde el procesamiento de imágenes en MATLAB hasta finalizar con la implementación de la aplicación en Android.

## 2 METODOLOGÍA DEL ALGORITMO EN MATLAB

### 2.1 PROCESOS

#### 2.1.1 PERIODO DE LECTURA

Para cumplir con la funcionalidad de la obtención de imagen es necesario solicitar el uso de la función `imread` que se encuentra en la figura 1. La cual permite la recepción de una imagen y su posterior lectura. El autor como [1] menciona la imagen a leer debe encontrarse en la carpeta de trabajo de Matlab. Los formatos de imagen soportados por Matlab son: TIFF, JPEG, GIF, BMP, PNG, XWD.

```

3      %% Obtención de imagen original, escala grises y RGB.
4 -    imagen = imread('9.jpg');
5 -    figure
6 -    subplot(2,2,1);
7 -    imshow(imagen);title('Imagen Original');
```

**Fig. 1. Lectura y Presentación de imagen mediante algoritmo**

Una vez leída la imagen se tiene que obtener una imagen con escala de grises y tres vectores que representen al conjunto RGB para obtener las matrices de cada color. Estas matrices tendrán las valoraciones de los pixeles como se muestra en la figura 2. En [1] se señala que una imagen de color RGB se simboliza por tres matrices bidimensionales, dados a los planos R, G y B.

```

8 -    imagen_grises = rgb2gray(imagen);
9 -    imR = double(imagen(:,:,1));
10 -    imG = double(imagen(:,:,2));
11 -    imB = double(imagen(:,:,3));
```

**Fig. 2. Obtención de imagen gris y vectores RGB por algoritmo**

#### 2.1.2 PRE-PROCESADO

Terminado el período de lectura, el proceso a llevar a cabo es del Pre – Procesado. En la figura 3 se aprecia el algoritmo del pre-proceso que comienza con obtener la capa que predomina más en una frutilla que es la capa de color rojo. Posterior a la obtención de la capa roja de la imagen se procede a realizar una transformación binaria de esta imagen. Para la eliminación de ruido fue de preferencia utilizar la función `medfilt2` que representa un filtro medio y dicha imagen filtrada es presenta en pantalla.

```

13     %% Pre-Procesado: Eliminación de ruido y filtro.
14 -    imagenC = imR - imG - imB;
15 -    imagen_binaria = imagenC>20;
16 -    imagen_binaria_filtrada = medfilt2(imagen_binaria);
17 -    subplot(2,2,2);
18 -    imshow(imagen_binaria_filtrada);title('Imagen binaria filtrada');
```

**Fig. 3. Algoritmo del Pre-Proceso**

### 2.1.3 SEGMENTACIÓN

Culminado el pre-proceso es momento de seguir con la segmentación. En la segmentación se realiza la eliminación de cosas con menor importancia y agregación de cosas relevantes. Por lo cual se realiza un relleno estructural mediante la función `strel` para rellenar huecos y otro para el mejoramiento del borde.

La imagen filtrada es dilatada en este proceso con la función `imfill`. Y dicha dilatación para por un proceso de relleno de pixeles que se encuentran en negro por su valoración de 0 y posterior se realiza la eliminación de pixeles aislados estos dos últimos por medio de la función `bwmorph` la cual utiliza `fill` para el relleno y `clean` para la eliminación.

Terminado el anterior proceso se obtiene un borde preliminar por medio de la función `sobel`. Para ayuda de mejoramiento del borde se procede a dilatar el borde con la función antes mencionada para este proceso. Parte del proceso de segmentación se refleja en algoritmo en la figura 4.

```

20 %% Proceso de Segmentacion: Histogramas, Dilatación
21 %%Relleno Interno de Pixeles, Eliminación de Pixeles Aislados, Borde.
22 - SE= strel('square',50);
23 - se = strel('octagon',3);
24 - dilatacion= imdilate(imagen_binaria_filtrada,SE);
25 - relleno = imfill(dilatacion,'holes');
26 - relle = bwmorph(relleno,'fill');
27 - rell = bwmorph(rele,'clean');
28 - subplot(2,2,3);
29 - imshow(rell);title('Imagen dilatada y con relleno');
30 - Borde= edge(rell,'sobel');
31 - d2=imdilate(Borde,se);
32 - subplot(2,2,4);
33 - imshow(d2);title('Borde');
    
```

Fig. 4. Proceso de Segmentación

En la figura 5 se observa la imagen original sobre la cual se trabaja para la prueba del sistema, también se aprecia la imagen filtrada, la imagen con relleno y dilatación es visible en la figura 5 y por último se refleja la imagen del borde de la frutilla.

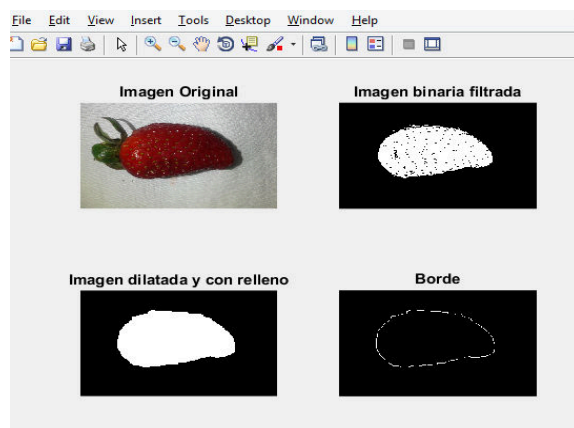


Fig. 5. Representación del proceso de segmentación y pre-proceso

Para mayor interpretación se implementa la visualización de histograma para la escala de grises, y los colores RGB. Dicha visualización se la realiza por algoritmo en la figura 6. Donde se obtienen los vectores RGB para sus respectivos histogramas, con su respectivo título de identificación y su representación en pantalla.

```

35 %% Proceso de graficación de histogramas RGB e Imagen Original.
36 - figure
37 - subplot(2,2,1);
38 - imhist(imagen_grises);
39 - title ('Escala grises');
40 - subplot(2,2,2);
41 - imhist(imagen(:,:,1));
42 - title ('Escala rojo');
43 - subplot(2,2,3);
44 - imhist(imagen(:,:,2));
45 - title ('Escala verde');
46 - subplot(2,2,4);
47 - imhist(imagen(:,:,3));
48 - title ('Escala azul');

```

Fig. 6. Algoritmo de histogramas RGB

En la figura 7 se aprecia la ejecución de este proceso para visualizar al detalle cual es el comportamiento que presentan los histogramas en sus respectivas capas y se logra apreciar que capa predomina sobre la otra.

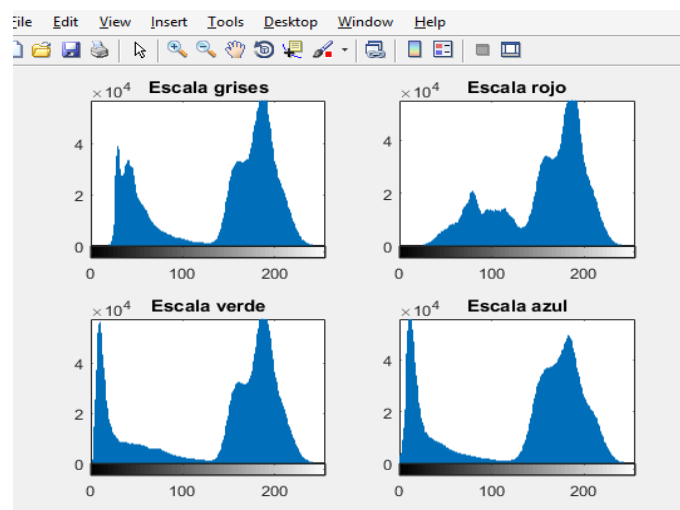


Fig. 7. Representación de histogramas

En [2] se dice que el histograma de una imagen  $f$  es una gráfica que representa los niveles de intensidad del color de  $f$  con respecto al número de píxeles presentes en  $f$  con cada intensidad de color. También dice en [2] el comando Matlab para realizar histograma de una imagen es `imhist`. El comando `imhist` debe usarse solamente con imágenes en escala de grises.

#### 2.1.4 EXTRACCIÓN DE CARACTERÍSTICAS

Por último se lleva a cabo el proceso de extracción de característica. La función para obtener una caracterización y reconocimiento del objeto como tal es `regionprops` y la función `bwlabel`. Esta última recibe la imagen "rell" para proceder a identificar el objeto o los objetos que se presenten en la imagen binaria. Mientras `regionprops` se encarga de obtener y visualizar las propiedades del objeto como se ve en la figura 8.

```

51     %% Proceso de Extracción de Características.
52     imagen_C = bwlabel(rell);
53     figure;
54     imshow(label2rgb(imagen_C));title('Identificación del objeto');
55     propiedades = regionprops(imagen_C);
56     obj= size(propiedades);
57     for j=1:obj;
58         disp('Propiedades del objeto identificado');
59         disp(propiedades(1));
60     end
    
```

Fig. 8. Algoritmo para la caracterización del objeto

En la figura 9 se aprecia el objeto representado con color que fue obtenido en la imagen binaria y en la ventana de comando se visualiza los datos que arrojan las propiedades del objeto que se reflejan en la figura 10.

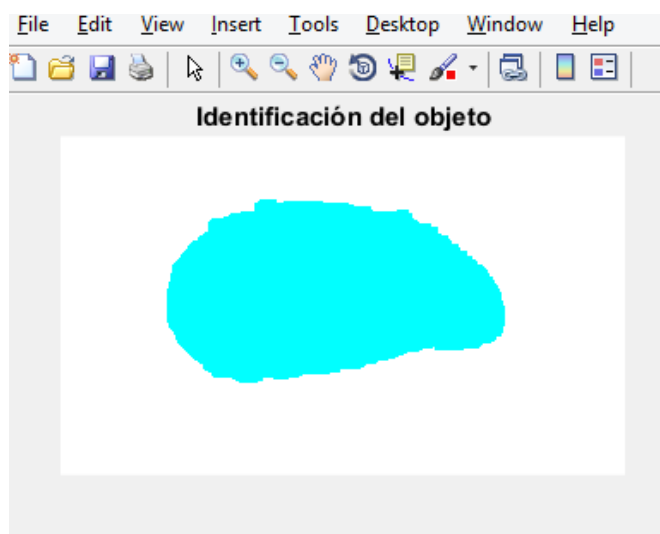


Fig. 9. Característica del objeto encontrado en la imagen binaria

```

Propiedades del objeto identificado
      Area: 964650
      Centroid: [1.2018e+03 705.3109]
      BoundingBox: [482.5000 287.5000 1534 830]
    
```

Fig. 10. Propiedades del objeto

### 3 RECOPIACIÓN DE DATOS

Como ya sabemos, todos los datos deben ser guardados esto para poder acceder luego a ellos de una manera rápida y precisa.

#### 3.1 VARIABLES

En [3] este caso se debe guardar cada etapa de la fruta en cuestión, con el dato a tomar en consideración, en este caso la más importante sería el color de la fruta, esto nos permitirá saber si la fruta está lista para su recolección o no.

Pero también existen otras variables a tomar en cuenta tales como:

- Color
- Tamaño

- Forma
- Tiempo de plantación

Los datos almacenados como el color, clasifican las frutillas, esto quiere decir que dependiendo si la frutilla presenta un tono aceptable está disponible para la recolección, en cambio si presenta otro tipo de tonalidad el sistema deberá evaluarlo. Con estos datos y a través del código se determina si la frutilla está lista para su recolección.

### **3.2 BASE DE DATOS**

Para guardar los datos se usará 2 herramientas: XAMPP y MySQL.

En lo que respecta a XAMPP es un sistema de gestión de bases de datos en las cuales se encuentran MySQL con el servidor web apache en esta herramienta también se encuentran los intérpretes de lenguajes de script (conjunto de instrucciones) que son PHP y Perl. XAMPP es un acrónimo de sus participantes colocando la X al inicio para representar que puede funcionar en cualquier sistema operativo y con la inicial de cada uno Apache, MariaDB/MySQL, PHP y Perl.

Cuando se habla en [4] de MySQL se refiere a un sistema gestor de base de datos relacionales, MySQL no es desarrollado por toda comunidad, así como lo es apache, este es administrado por una empresa privada, esta posee su validación de propiedad privada o copyright en gran parte de su código, es utilizada sobre todo en aplicativos webs, tales como:

- Joomla
- WordPress
- Drupal
- phpBB,

#### **3.2.1 CONEXIÓN MYSQL Y MATLAB**

Para [5] enlazar la base de datos de MySQL en Matlab primero se debe crear una base de datos que posea un nombre característico para el proyecto, esto para facilitar la búsqueda de la base de datos creada. Se debe crear tablas que posea un nombre el cual identifique que posee la tabla, se crea las tablas dependiendo de cuantas se necesiten, estas tablas deben estar relacionadas entre sí. Una manera rápida de realizar las tablas es por el método gráfico el cual permite crear tablas y relacionarlas creando el diagrama entidad relación. Pero cuando se crea de manera gráfica se debe convertir las tablas para ingresarlas en nuestra base de datos.

En [6] después de la creación de las tablas se debe seleccionar la conexión, colocar su usuario para acceder; en MySQL, el proceso es automáticamente seleccionado, con esto se puede avanzar con la selección de siguiente, la base de datos creada debe ser revisada en caso de que exista un error.

##### **3.2.1.1 CONECTOR ODBC**

Para realizar la conexión de Matlab a MySQL se necesita poseer un conector de nombre MySQL conector ODBC, este es un conector de MySQL, al momento de su instalación las opciones a seleccionar están designadas por defecto, para habilitar se debe ir a herramientas administrativas, se busca los orígenes de datos, este proceso envía a la sección del conector en el que se procede a llenar los campos estos son:

- Nombre
- Descripción
- Nombre de usuario
- Nombre de usuario de MySQL
- Contraseña

Después de llenar todo los campos se debe seleccionar la base de datos a la que se desea que Matlab realice la conexión, una vez la prueba finalice ya se puede trabajar con esta base.

Ahora para seguir con el proceso, se debe ingresar a Matlab y se ejecuta el comando dexplore en el comand window. Esto abre una ventana conect to a data source, en esta ventana se va a seleccionar la conexión creada con el conector ODBC, después se ingresa el nombre de usuario y contraseña. Esto abre la ventana donde se encuentra la base de datos, se debe crear un programa por cada base de datos. Ahora se escribe el código guide este abrirá una ventana para crear una nueva GUI (interfaz gráfica de usuario).

Para obtener una guía de un fondo con imagen se usa el botón Axes, esto se debe guardar , y se creara automáticamente un archivo .m donde se puede apreciar el código, se busca la parte del código que contenga el mismo nombre que nuestro

archivo .m , se borran los comentarios, cuando esto se realice se creara una variable con el nombre imagen esta guardara lo que se ingrese dentro del paréntesis en este caso se coloca el nombre de la imagen o texto que se desea , para poder ingresar una imagen, esta debe estar en la misma carpeta que el archivo .m, se debe desactivar axes para que al momento de ejecutar el programa presente la variable guardada.

Después se deben crear 3 (puede requerir más) aplicaciones para el ingreso de datos, para esto una vez creada las 3 aplicaciones, después se debe dirigir al archivo del menú principal , este archivo posee una extensión .fig, al dar clic derecho se busca la opción open un guide, esto permitirá que aparezca la ventana del archivo .fig, ingresando a la opción menú editor se debe seleccionar new menú, al crearse este menú se lo puede configurar si el menú se encuentra como se desea se presiona la opción view para ser enviado automáticamente la parte del código, se eliminan los comentarios y se escribe las siguientes líneas de código, close para cerrar y luego el nombre del archivo que se desea abrir, al ejecutar esta parte nos presentara un menú en la parte superior con el nombre que se le coloco esto re-direccionará a la opción que se colocó después del close.

Después de este proceso se retorna al archivo .fig y se le da clic derecho se selecciona la opción open in guide, esto para colocar los campos para la base de datos , esto se consigue con un static text para poder ingresar el título y un edit text por cada campo que posee una la tabla, para subir lo datos se puede usar un push butoon pero antes de subir los datos se deben visualizar para conseguir esto se construye una tabla creada con el botón table una vez creada la tabla se debe asegurar que hay las suficientes secciones en caso de que no sea así podemos aumentar columnas dando clic derecho sobre la tabla y elegir editor de propiedades de tabla , cada tabla debe tener su tamaño y su nombre.

Para la edición de los botones ya creados se puede acceder a las opciones dando clic derecho y dirigiéndose a view callbarks y callback, esto abre la sección al código que le corresponde al botón, para el código se crea una variable con un nombre dato 1 esto será igual a lo que se ingrese en una variable tag, esta es una variable creada por cada campo creada por el id , una vez que se sepa el tag se lo reemplaza en el código, esta operación se repite para ordenar los valores que se van a ingresar.

Pasando estos pasos se crea una variable que será igual al nombre de la base de datos a la que se realizó la conexión anteriormente, después se debe ingresar el nombre de la base de datos, el nombre de usuario y contraseña, se crea una variable para contener los valores de los datos y una variable para guardar que será igual al nombre de las columnas, en esta parte se tendrá el envío y guardado de datos, para saber que los datos han sido enviados o guardados exitosamente se puede crear un mensaje, con msgbox que enviara un mensaje dependiendo si se guardan o envían.

Los datos guardados son almacenados en mysql, estos datos pueden ser visualizados en la tabla, solo se debe importar, se realiza de la siguiente manera: desde el botón creado para mostrar se da clic derecho y se selecciona view callback , callback. Esto envía a la sección del botón mostrar, en data base explore se selecciona la tabla a la cual se desee importar los datos , una vez realizado esto se selecciona importar (generar scrip) esto crea un archivo .m este posee el código de importación, este es copiado y pegado en la sección del botón mostrar, esto con esto se coloca el tag donde se desea que se muestren los datos , junto con esto se coloca la variable de donde se importarán las datos permitiendo visualizar los datos que se ingresan a la base.

## **4 APLICATIVO MÓVIL (ANDROID)**

En [7] señala a Android como un sistema operativo que trabaja bajo la plataforma Linux, la cual es de código abierto, brindando un amplio abanico en creación de funciones con lenguajes de programación distintas. En Android, se admite desarrollar en un entorno de trabajo (framework) de lenguaje JAVA. El lado beneficioso de Android es permitir la creación de un sin número de aplicaciones para lanzarlas al mercado o, utilizar dicha herramienta para alterar el sistema operativo.

### **4.1 VERSIONES**

Desde su creación en el año 2005 hasta la actualidad, Android ha lanzado 17 versiones con nombres muy peculiares basados en deliciosos postres. Cabe recalcar que la primera versión tuvo un nombre acorde llamado “Android Beta” lanzado en el 5 de noviembre del 2007 pero, no fue hasta el año 2008 donde consiguió una versión más estable llamada “Android 1.0”, donde [8] indica que el primer teléfono con este sistema operativo fue HTC Dream / T-Mobile G1.

A partir del año 2008 con su versión estable 1.0 cuyo nombre de código era “Banana Bread” hasta en la actualidad, Android se convirtió en uno de los sistemas operativos mayormente usados en celulares, tanto para la marca Samsung, Xiaomi, entre otros. Desde aquí parten sus versiones con los distintivos nombres de postres, los cuales se detallan a continuación:

Tabla 1. Tabla de versiones de Android

NOMBRE (CÓDIGO)	NÚMERO DE VERSIÓN	FECHA DE LANZAMIENTO	NIVEL DE API
Apple Pie1	1.0	23 de septiembre de 2008	1
Banana Bread1	1.1	9 de febrero de 2009	2
Cupcake	1.5	25 de abril de 2009	3
Donut	1.6	15 de septiembre de 2009	4
Eclair	2.0 – 2.1	26 de octubre de 2009	5 – 7
Froyo	2.2 – 2.2.3	20 de mayo de 2010	8
Gingerbread	2.3 – 2.3.7	6 de diciembre de 2010	9 – 10
Honeycomb2	3.0 – 3.2.6	22 de febrero de 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.5	18 de octubre de 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 de julio de 2012	16 – 18
KitKat	4.4 – 4.4.4	31 de octubre de 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 de noviembre de 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 de octubre de 2015	23
Nougat	7.0 – 7.1.2	15 de junio de 2016	24 – 25
Oreo	8.0 – 8.1	21 de agosto de 2017	26 – 27
Pie	9.0	6 de agosto de 2018	28
Android 10 3	10.0	3 de septiembre de 2019	29

Con lo que corresponde a API en la tabla 1, sus siglas significan **Application Programming Interface (Interfaz de Programación de Aplicaciones)**, según [9] es una especificación de la comunicación de dos componentes de software totalmente diferentes y así, poder receptor y enviar información entre ambos. Se tiene como ejemplo la API de la cámara, haciendo que el desarrollador evite consumir tiempo en la creación de una aplicación que le admita el acceso a la cámara, sino, solamente crear una aplicación a nivel general que permita acceder a lo que capta el sensor de la lente de cámara y tomar las fotos.

#### 4.2 ESTRUCTURA DE LA APLICACIÓN

Esta aplicación ayudará al encargado del huerto de las frutillas minimizando el tiempo de inspección de las frutas, todo esto con la unión de la identificación de procesamiento de imágenes de MATLAB y su recopilación de datos en MySQL conectada a una aplicación Android.

Esta aplicación contará con las siguientes pantallas:



Fig. 11. Inicio y Registro de usuario



En la figura 11, el usuario al momento de abrir la aplicación le saldrá la pantalla de registro, en el cual si ya cuenta con uno y es aprobado por el administrador, sencillamente se ubicará a la opción de iniciar sesión.

Contará con los siguientes perfiles:

- **Perfil de administrador.** Aquí únicamente dueños del huerto podrán acceder a la información, sólo aquellos autorizados podrán aceptar el registro de nuevo personal.

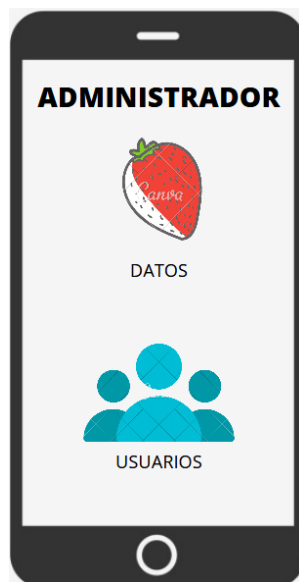


Fig. 12. Perfil principal de Administrador



Fig. 13. Pantalla de aceptación de usuarios

En la figura 12, el administrador tendrá dos opciones: visualizar los datos de su huerto de frutillas para prepararse a la recolección de tal o visualizar los nuevos usuarios que desean ingresar. En la figura 13 el administrador tendrá dividido en dos partes: usuarios registrados que son los que ya fueron aceptados y, los usuarios pendientes, que son aquellos nuevos registros que no han sido aprobados para ingresar al sistema de monitoreo de las frutillas. Esto se ideó de tal forma que únicamente los usuarios de mayor confianza o personas desconocidas no tengan acceso fácilmente al sistema, permitiendo un mejor control y mejorando la seguridad. Por el contrario, si una persona externa si observa detenidamente la información podrá sacar cuentas de los días donde la fruta esté lista y proceder a sustraer la misma con mayor precisión.

- **Perfil de usuario.** El nuevo personal podrá crear una cuenta, siempre y cuando el administrador lo habilite.



Fig. 14. Pantalla de aceptación de usuarios

En la figura 14, tanto el administrador como el usuario en general, cabe recalcar un usuario aprobado, podrán visualizar los datos actualizados de cada hilera, sea esta de 1 columna o doble, la aplicación se modificaría acorde a las especificaciones de la parte interesada en esta, haciendo que lleve el control de las frutillas antes de que pasen al proceso de putrefacción.

#### 4.2.1 CONEXIÓN MYSQL CON ANDROID

Una vez teniendo claras las definiciones de la base de datos MySQL y Android, se procede al siguiente paso que corresponde a la conexión de ambas para lograr tener como resultado una aplicación que permita mostrar la cantidad de frutillas en sus 3 diferentes estados de maduración: verde, madura y podrida.

Los pasos a seguir para esta conexión con un gestor de base de datos externo, en este caso MySQL con una aplicación Android son los siguientes:

1. Para poder llevar a cabo la conexión se debe tener un servidor local para las correspondientes pruebas.
2. Agregar la librería 'mysql-connector-java-5.1.3X-bin.jar', para las distintas funciones que hay de conexión y el CRUD a crear, como lo que es la lectura, escritura o eliminación de algún elemento de la base de datos.
3. Al momento de escribir el código se debe tener en cuenta el import para las clases `java.sql.Connection` y `java.sql.DriverManager`, el driver correspondiente al conector JDBC.

Según [10] indica que las siglas JDBC significan Java Database Connectivity, la cual es una API que dará paso a las ejecuciones de las operaciones de las bases de datos, todo esto desde JAVA, sin importar el sistema operativo desde el que se está ejecutando.

4. Una vez cargado el driver, sólo queda establecer la conexión con el servidor de la base de datos escogida desde el inicio, es decir, MySQL. Se especifica todos los parámetros: usuario, contraseña, la url y la base de datos que irá conectada.
5. Como dato importante, es preferible cerrar sesión en cada acción que se haga en la base de datos.
6. Por último, en lo que corresponde la lógica de la conexión de la base de datos, todos es manejado desde la clase `MainActivity`.

## **5 DISCUSIÓN**

En este documento se puede visualizar las ventajas del uso de MATLAB y MySQL, con estos resultado se muestra las diferentes ventajas de la base de datos escogida que junto con el programa cumplen la función deseada, esto se debe a la variedad de lenguajes que puede aceptar, puede trabajar con la base en diferentes aplicativos. Con esto la base de datos puede ser la misma en el aplicativo.

## **6 RESULTADO**

Se tiene como resultado una aplicación Android, donde se visualizará por hileras, siendo estas de una o doble columna, las frutillas en su estado de maduración, arrojando la cantidad de cada etapa, es decir, si se encuentra en una hilera 10 frutillas en etapa temprana de maduración, se actualizará la base de datos y la aplicación marcará tal cantidad. Mismo proceso con las siguientes etapas. En sí, se ha enfocado más en la parte de software, para mayor resultado y minimizar costos operacionales como lo son la mano de obra, un buen mecanismo para esto es la creación de un robot incorporado con el algoritmo de MATLAB y a su vez pueda servir de recolector de las frutillas.

## **7 CONCLUSIÓN**

Con la utilización de este aplicativo se podrá trabajar de una manera eficiente, esto reduciendo el tiempo de trabajo ya que conociendo el estado de la fruta los recolectores actuarían directamente en las frutillas maduras dejando frutas en mal estado atrás o también lograr retirarlas, esto para que no dañe al resto.

## **REFERENCIAS**

- [1] B. M. Garfia, «Procesamiento de imágenes con MATLAB,» vol. 1, nº 1-20, p. 20, 2014.
- [2] A. Manna, Escritor, *Introducción al Procesamiento de Imágenes con MATLAB*. [Performance]. Departamento de Computación Facultad de Ciencias Exactas y Naturales, 2016.
- [3] Paucar, Juanazo, and Freddy Alex. "Desarrollo de una herramienta informática basada en visión artificial para el reconocimiento del banano para la exportación aplicando procesamiento de imágenes digitales y su simulación en Matlab mediante redes neuronales". BS thesis. Universidad de Guayaquil Facultad de Ciencias Matemáticas y Físicas Carrera de Ingeniería en Sistemas Computacionales, 2019.
- [4] Santillán, Luis Alberto Casillas, Marc Gibert Ginestà, and Óscar Pérez Mora. "Bases de datos en MySQL." Universitat Oberta de Catalunya, 2014.
- [5] Vasquez, Y., et al. "Procesamiento de las Imágenes con Matlab (Venas de los dedos)."
- [6] Pérez López, César. *MYSQL para Windows y Linux*, 2004
- [7] M. Báez, A. Borrego, J. Cordero, L. Cruz, M. González, F. Hernández, D. Palomero, J. Rodríguez, D. Sanz, M. Saucedo, P. Torraldo, A. Zapata *Introducción a Android 1ra Ed. EME Madrid*, España, p. 2, 2012.
- [8] Iván Ramírez. *Historia y evolución de Android: cómo un sistema operativo para cámaras digitales acabó conquistando los móviles*.  
[Online] Available: <https://www.xatakandroid.com/sistema-operativo/historia-y-evolucion-de-android-como-un-sistema-operativo-para-cameras-digitales-acabo-conquistando-los-moviles> (January 7, 2018)
- [9] Iván Linares. *Explicación de las API*.  
[Online] Available: <https://elandroidelibre.lespanol.com/2017/09/que-es-api-software-android.html> (September 4, 2017)
- [10] R. Chica. *Aplicación Android para la gestión de la compra*. p. 16, 2014.