



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de Software

Trabajo Fin de Grado

App para móviles de detección de características hortofrutícolas mediante tratamiento de imágenes.



UNIVERSIDAD DE EXTREMADURA

Escuela Politécnica

Grado en Ingeniería Informática en Ingeniería de Software

Trabajo Fin de Grado

App para móviles de detección de características  
hortofrutícolas mediante tratamiento de imágenes.

Autor: César Pizarro Jacinto

Tutor: Adolfo Lozano Tello

## ÍNDICE GENERAL DE CONTENIDOS

|  |           |
|--|-----------|
| <b>ÍNDICE DE FIGURAS.....</b>                | <b>4</b>  |
| <b>RESUMEN .....</b>                         | <b>6</b>  |
| <b>1. INTRODUCCIÓN .....</b>                 | <b>7</b>  |
| <b>2. OBJETIVOS .....</b>                    | <b>8</b>  |
| <b>3. ANTECEDENTES.....</b>                  | <b>9</b>  |
| <b>4. ANÁLISIS Y DISEÑO .....</b>            | <b>10</b> |
| <b>4.1 Obtención del Color .....</b>         | <b>10</b> |
| <b>4.2 Análisis del Color.....</b>           | <b>10</b> |
| <b>4.3 Resultados .....</b>                  | <b>11</b> |
| <b>4.4 Requisitos Funcionales .....</b>      | <b>11</b> |
| <b>4.5 Requisitos No Funcionales.....</b>    | <b>11</b> |
| <b>4.6 Casos de Uso .....</b>                | <b>12</b> |
| <b>4.7 Diagrama de Actividad .....</b>       | <b>13</b> |
| <b>5. MATERIAL Y MÉTODO .....</b>            | <b>14</b> |
| <b>5.1 Investigación .....</b>               | <b>14</b> |
| 5.1.1 Android.....                           | 14        |
| 5.1.2 Colorímetro .....                      | 14        |
| 5.1.3 Fotografía e Imagen .....              | 15        |
| <b>5.2 Desarrollo Android.....</b>           | <b>15</b> |
| 5.2.1 Introducción .....                     | 15        |
| 5.2.2 Android Manifest .....                 | 15        |
| 5.2.3 Librerías .....                        | 16        |
| 5.2.4 Recursos y vistas.....                 | 17        |
| 5.2.5 Lógica de Aplicación .....             | 18        |
| 5.2.5.1 Métodos principales.....             | 19        |
| 5.2.5.2 Métodos adicionales .....            | 27        |
| <b>5.3 Registros de color.....</b>           | <b>34</b> |
| <b>5.4 Características de la fruta .....</b> | <b>35</b> |
| <b>6. RESULTADOS Y DISCUSIÓN.....</b>        | <b>37</b> |
| <b>7. CONCLUSIONES.....</b>                  | <b>38</b> |

|  |           |
|--|-----------|
| <b>REFERENCIAS BIBLIOGRÁFICAS .....</b>  | <b>39</b> |
| <b>BIBLIOGRAFÍA.....</b>   | <b>40</b> |
| <b>ANEXOS.....</b>   | <b>41</b> |
| <b>Anexo 1 – Tomas Colorímetro “Lab” .....</b>                                 | <b>41</b> |
| <b>Anexo 2 – Tomas Cámara Samsung S3 Luz directa a 8 cm “RGB” .....</b>        | <b>44</b> |
| <b>Anexo 3 – Tomas Cámara Samsung S3 Luz Directa a 8cm “Lab” .....</b>         | <b>46</b> |
| <b>Anexo 4 – Fórmulas de Conversión Cámara “Lab” a Colorímetro “Lab” .....</b> | <b>50</b> |

## ÍNDICE DE FIGURAS

|  |    |
|--|----|
| Ilustración 1 - Casos de uso .....                               | 12 |
| Ilustración 2 - Diagrama de Actividad .....                      | 13 |
| Ilustración 3 - Minolta Chroma Meter CR-300.....                 | 14 |
| Ilustración 4 – Android Manifest, Actividad Principal.....       | 15 |
| Ilustración 5 - Android Manifest, Cámara .....                   | 16 |
| Ilustración 6 - Android Manifest, Escritura almacenamiento ..... | 16 |
| Ilustración 7 - Android Manifest, Portrait.....                  | 16 |
| Ilustración 8 - Librerías, SDK.....                              | 16 |
| Ilustración 9 - Librerías, Externas.....                         | 17 |
| Ilustración 10 - Recursos y vistas, Main .....                   | 18 |
| Ilustración 11 - Métodos principales, getAlbumName .....         | 19 |
| Ilustración 12 - Métodos principales, getAlbumStorageDir .....   | 19 |
| Ilustración 13 - Métodos principales, getAlbumDir .....          | 19 |
| Ilustración 14 - Métodos principales, onCreate .....             | 20 |
| Ilustración 15 - Métodos principales, createImageFile .....      | 20 |
| Ilustración 16 - Métodos principales, startCamera .....          | 21 |
| Ilustración 17 - Métodos principales, onActivityResult .....     | 21 |

|   |    |
|---|----|
| Ilustración 18 - Métodos principales, galleryAddPic.....              | 22 |
| Ilustración 19 - Métodos principales, InstanceState .....             | 22 |
| Ilustración 20 - Métodos principales, setPic .....                    | 23 |
| Ilustración 21 - Métodos principales, centralPixelColor 1/2 .....     | 24 |
| Ilustración 22 - Métodos principales, centralPixelColor 2/2 .....     | 25 |
| Ilustración 23 - Métodos principales, algorithmConversionLAB 1/3..... | 25 |
| Ilustración 24 - Métodos principales, algorithmConversionLAB 2/3..... | 25 |
| Ilustración 25 - Métodos principales, algorithmConversionLAB 3/3..... | 26 |
| Ilustración 26 - Métodos principales, LBAtoLCH.....                   | 26 |
| Ilustración 27 - Métodos adicionales, fixOrientation.....             | 27 |
| Ilustración 28 - Métodos adicionales, metodos Bitmap.....             | 27 |
| Ilustración 29 - Métodos adicionales, modificaciones adicionales..... | 28 |
| Ilustración 30 - Métodos adicionales, replaceColor.....               | 29 |
| Ilustración 31 - Métodos adicionales, processEdge .....               | 30 |
| Ilustración 32 - Métodos adicionales, eraseBackground.....            | 31 |
| Ilustración 33 - Métodos adicionales, createHistogram 1/3.....        | 32 |
| Ilustración 34 - Métodos adicionales, createHistogram 2/3.....        | 33 |
| Ilustración 35 - Métodos adicionales, createHistogram 3/3.....        | 33 |
| Ilustración 36 - Plantilla de colores.....                            | 34 |
| Ilustración 37 - Características de la fruta.....                     | 35 |

## RESUMEN

El proyecto desarrolla una aplicación para teléfonos Android. Dicha aplicación permite fotografiar piezas de fruta y analizar sus características propias. Hasta ahora, el método para dicho análisis consta de analizar el color de la fruta a través de un colorímetro, e insertar dichos valores en formulas matemáticas para obtener los resultados requeridos. Para esto, se necesita disponer de dicho dispositivo, y de las habilidades para manejarlo y utilizar los datos obtenidos.

Para facilitar y simplificar todo ese trabajo, la aplicación desarrollada consistirá en una interfaz que simplemente, tras tomar una fotografía de la fruta, indicará la información que se desea obtener.

Las investigaciones sobre modos de color y regresiones polinomiales fueron vitales para comprobar si el proyecto se podría llevar a cabo, ya que la conversión de una fotografía a valores de un colorímetro parecía difícil de aproximar.

La facilidad de uso fue una las metas de este proyecto, que intenta acercar y acelerar el proceso de análisis de parámetros de una fruta. La posibilidad de conseguir dichos análisis sin necesidad de recolectar la fruta ni impedir su crecimiento la hacen una aplicación llamativa e interesante para el usuario final.

## **1. INTRODUCCIÓN**

Los Smartphone están al alcance de la mano. Sus aplicaciones facilita la realización de cualquier tarea diaria, desde lectores de códigos a sensores de sonido. Desde ese enfoque, se piensa en facilitar el análisis de las características de una fruta.

La idea emerge de nuestros colaboradores, los Ingenieros Abelardo García y Luis Lorenzo Paniagua, los cuales desarrollaron un estudio para la conversión del color de una fruta en características de dicha fruta. Para ello utilizaron un colorímetro, el cual obtiene los valores de color de una fruta, para desarrollar las formulas en función de dichos valores.

Nuestro trabajo de investigación comienza desde ese punto, con el estudio de la relación entre el color y las características de una fruta, y con el objetivo de concentrarlo todo en una aplicación para Smartphone.

## 2. OBJETIVOS

La relación entre distintos modos de color y distintos dispositivos de medida suele generar dudas. Un objetivo principal es obtener valores de color con un Smartphone, y que esos valores sean realmente aproximados a los valores de un colorímetro. Distancia de enfoque, horario del día, luz directa solar o flash, son distintos entornos fotográficos.

Otro objetivo, pero dependiente del anterior, es añadir las formulas que conectan los valores de color con las características de la fruta. Las formulas están desarrolladas con la ayuda del colorímetro, por lo que hay que reducir los márgenes de error lo máximo posible.

Paralelamente, otro objetivo principal es crear una aplicación en Android capaz de ejecutar las tareas que requieran el resto de objetivos. Se seleccionó la plataforma Android por la cantidad de dispositivos que tiene en activo, intentando llegar al mayor número de usuarios.

Cumpliendo todos los objetivos, obtenemos una aplicación en Android que es capaz de obtener la tonalidad de una fruta como lo haría un colorímetro, y que, usando los datos obtenidos de la fotografía, calcula características de dicha fruta.



### **3. ANTECEDENTES**

El proyecto comienza entorno a un trabajo de investigación realizado por Abelardo García y Luis Lorenzo Paniagua, Ingenieros de la Universidad de Extremadura. Dicho trabajo de investigación relaciona el color de una uva con sus características.

Disponemos del colorímetro con el que desarrollaron dicho trabajo, y los consejos necesarios para utilizarlo. Tras analizar dicho trabajo, obtenemos las necesidades de nuestro proyecto y el área de investigación que necesitamos enfocar.

Según Torres(2013), se puede relacionar el color y el estado de madurez con las propiedades de dichas frutas, en su caso frutas tropicales, a partir del modo de color “Lab”.

Kit L. Yam(2004) nos demuestra como analizar y medir el color de la superficie de una pieza de fruta, utilizando el modo de color “Lab” por su amplio rango.

En nuestro proyecto, el modo de color “Lab” será un pilar clave del análisis.

## **4. ANÁLISIS Y DISEÑO**

En este capítulo se encuentran las fases de análisis y diseño de las distintas partes en las que se puede particionar este proyecto. Diferenciamos la obtención de color con las aproximaciones del color obtenido con el color del dispositivo utilizado, y su posterior análisis para extraer resultados finales. Se incluyen también los requisitos funcionales y no funcionales, al igual que los casos de uso y diagramas de actividad.

### **4.1 Obtención del Color**

Fotografiar una fruta, esa es la base de nuestro análisis. Como obtener toda la información posible de la captura de una foto a una fruta.

El primer enfoque consiste en obtener todos los píxeles de la fruta en la fotografía. A continuación, con los píxeles obtenidos, se obtiene una gráfica de colores con los tonos más predominantes, o se obtiene un color “medio” a partir de los píxeles obtenidos. El segundo enfoque, el escogido finalmente, consiste en centrar la fruta en la fotografía para obtener el píxel central. Este segundo enfoque nos acerca a las tomas obtenidas del colorímetro, ya que dicho dispositivo obtiene también el píxel central de la toma de color.

El lugar y la distancia para la toma de la fotografía es otro proceso a analizar. Se hacen dos estudios con tomas de prueba, uno con luz solar directa a la fruta, y el otro sin luz directa, ambos sin rango de distancia a la fruta (entre 5 y 30 centímetros). Se repiten dichas tomas de pruebas con un rango fijo de 8 centímetros, el mínimo para conseguir un buen enfoque de la cámara del Smartphone. Los valores menos dispersos se encuentran en las tomas con luz solar directa y rango fijo de 8 centímetros, por lo que se toman dichas medidas para los cálculos siguientes.

### **4.2 Análisis del Color**

A partir del píxel de color obtenido, con una toma de luz solar directa y una distancia de 8cm a la fruta, se requiere unas formulas de conversión para conseguir los resultados más similares a los del colorímetro.

Se utilizan funciones de conversión para pasar del modo de color “RGB”, que se obtiene de la cámara del Smartphone, al modo de color “Lab”, el cual se obtiene del colorímetro. También se utilizan funciones de regresión para conseguir el menor error posible entre tomas de prueba de los dispositivos.

Ambos pasos consiguen que, tras obtener una fotografía con el Smartphone, obtengamos unos valores del modo de color “Lab” similares a las tomas con el colorímetro.

### **4.3 Resultados**

Tras obtener unos resultados óptimos con el Smartphone, adaptamos el modo de color de las fotografías obtenidas a las formulas de resultados. Dichas formulas se encargan de transformar el color de una fruta en características de dicha fruta, tales como acidez o pH.

Los resultados se muestran en la pantalla del Smartphone junto con la foto obtenida y el color analizado.

### **4.4 Requisitos Funcionales**

1. Debe tener un botón de fácil acceso para hacer una fotografía.
2. Debe mostrar la fotografía tomada y, opcionalmente, mostrar un recuadro con el color seleccionado.
3. Debe mostrar las características de la fruta tras obtener la fotografía, organizadas y todas visibles en la pantalla.

### **4.5 Requisitos No Funcionales**

1. Diseño usable e intuitivo
2. Respuesta y análisis rápido del sistema.
3. Menor margen de error posible en la conversión de modos de color.

## 4.6 Casos de Uso

Un caso de uso es una descripción de los pasos o actividades que deberán realizarse para llevar a cabo algún proceso.

En este caso, el único proceso a realizar es la opción de hacer una fotografía.

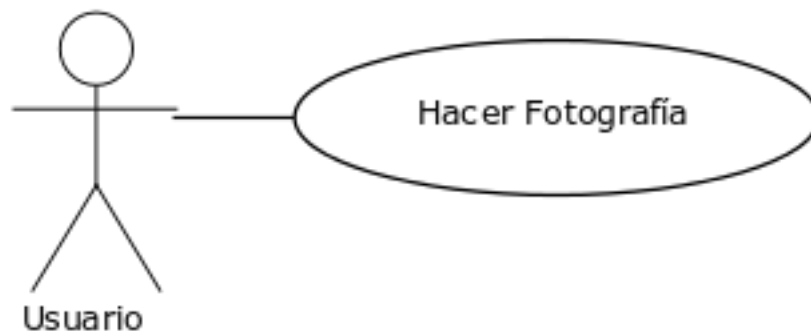


Ilustración 1 - Casos de uso

|                       |   |
|-----------------------|---|
| <b>Nombre:</b>        | Hacer Fotografía.   |
| <b>Actores:</b>       | Usuario.  |
| <b>Función:</b>       | Obtener una fotografía.   |
| <b>Descripción:</b>   | Se realiza una fotografía a través de la aplicación nativa de “Cámara” del dispositivo. |
| <b>Precondición:</b>  | El dispositivo(Smartphone) debe disponer de cámara.                                     |
| <b>Postcondición:</b> | Se obtiene una imagen, la cual se usará para obtener píxeles de color para su análisis. |

## 4.7 Diagrama de Actividad

Un diagrama de actividad muestra un proceso de software como un flujo de trabajo a través de una serie de acciones.

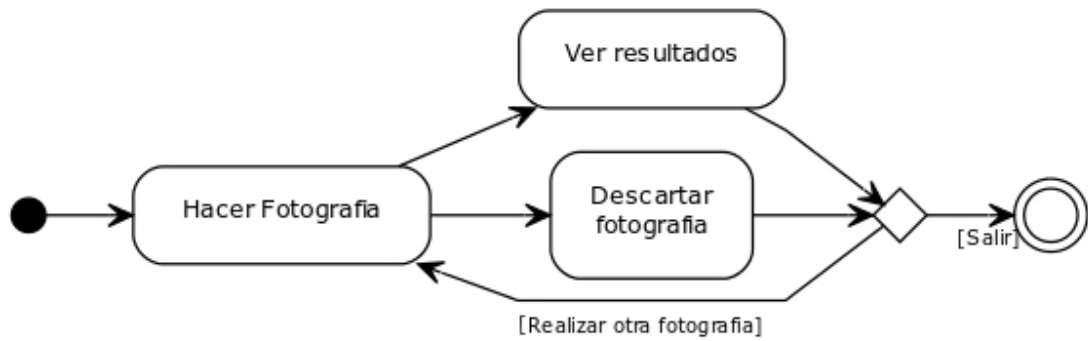


Ilustración 2 - Diagrama de Actividad

En este caso, vemos que, tras hacer una fotografía podemos ver los resultados o descartar la fotografía. Tras ello podremos realizar otra fotografía o salir de la aplicación.

## 5. MATERIAL Y MÉTODO

### 5.1 Investigación

#### 5.1.1 Android

El proyecto comienza con la creación de una App de Android, para la cual es necesario investigar como comenzar desde cero. Android dispone de una web para desarrolladores [1] que ayuda a encauzar la creación desde la base.

Lo primero a estudiar en un sistema nuevo es su estructura y programas de desarrollo. Android esta basado en Linux Kernel, y sus aplicaciones son desarrolladas en lenguaje Java, por lo que utilizaremos Android Studio[2], IDE oficial de la plataforma Android. Como dispositivo para la ejecución de la aplicación, utilizaremos un Samsung Galaxy S3 [3] con la versión de Android 4.3 Jelly Bean (API level 18).

#### 5.1.2 Colorímetro

El colorímetro utilizado para el proyecto es Minolta Chroma Meter CR-300 [4]. Obtiene el pixel central del área de acción que tiene en el extremo, y devuelve los resultados en distintos formatos como “Lab”, “LCH” o “XYZ” entre otros. El análisis de colores que vamos a utilizar se obtiene en formato “Lab”.



Ilustración 3 - Minolta Chroma Meter CR-300

### 5.1.3 Fotografía e Imagen

La cámara del dispositivo Smartphone va a ser clave para el desarrollo. Con 8Megapixels, el Samsung Galaxy S3 nos proporcionará la calidad necesaria para el análisis.

La aplicación Android nos proporcionara los valores RGB de la fotografía obtenida. El colorímetro, por otro lado, recogemos los datos en formato “Lab”, por lo que necesitamos formulas de conversión de RGB de la cámara del Smartphone a “Lab” del colorímetro.

## 5.2 Desarrollo Android

### 5.2.1 Introducción

Toda aplicación Android se compone de una estructura básica, que se compone por el Android Manifest, Librerías, Recursos y vistas, y Lógica de aplicación. A continuación explicaremos cada una de ellas y analizaremos el código de los archivos principales.

### 5.2.2 Android Manifest

“AndroidManifest.xml” es un archivo de configuración que se encarga de las características básicas de una aplicación Android. Esta situado en la raíz de la aplicación. En dicho archivo se declaran obligatoriamente todas las actividades de la aplicación, junto con los permisos opcionales que necesitemos.

La aplicación que hemos desarrollado utiliza una única actividad, la cual será llamada al iniciar la aplicación.

```
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

Ilustración 4 – Android Manifest, Actividad Principal

Se añaden dos permisos para la utilización de la aplicación.

El primero se encarga de habilitar la utilización de la cámara del dispositivo. Utilizaremos la cámara para capturar las fotografías necesarias para el análisis.

```
<uses-feature android:name="android.hardware.camera"
            android:required="true" />
```

**Ilustración 5 - Android Manifest, Cámara**

El segundo se encarga de dar permiso de escritura en el almacenamiento del dispositivo. El permiso de escritura nos proporciona automáticamente el permiso de lectura. Utilizaremos el almacenamiento para guardar las fotografías tomadas por la aplicación.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

**Ilustración 6 - Android Manifest, Escritura almacenamiento**

También se añade una configuración a la actividad Main, por la cual mantendremos la aplicación en todo momento en "Portrait"(modo vertical).

```
android:screenOrientation="nosensor"
android:configChanges="orientation"
```

**Ilustración 7 - Android Manifest, Portrait**

### 5.2.3 Librerías

En nuestra aplicación vamos a encontrar dos grupos de librerías: Las internas y las externas.

Las librerías internas vienen asociadas a la versión SDK utilizada para compilar la aplicación. Nuestra aplicación utiliza la versión SDK 23.

```
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.1"
    defaultConfig {
        applicationId "es.unex.cesarpizarrojacinto.proyectotfg"
        minSdkVersion 15
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
}
```

**Ilustración 8 - Librerías, SDK**

Las librerías externas son las que incluiremos adicionalmente para ampliar nuestros recursos. Añadimos dos:



- GpuImage de cyberagent [5], la cual nos ayudara con métodos de análisis fotográfico.
- Android Support v4 [6], la cual permite ejecutar en versiones antiguas de Android APIs mas recientes.

```
dependencias {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    compile 'jp.co.cyberagent.android.gpuimage:gpuimage-library:1.3.0'  
    compile 'com.android.support:support-v4:24.0.0-alpha1'  
}
```

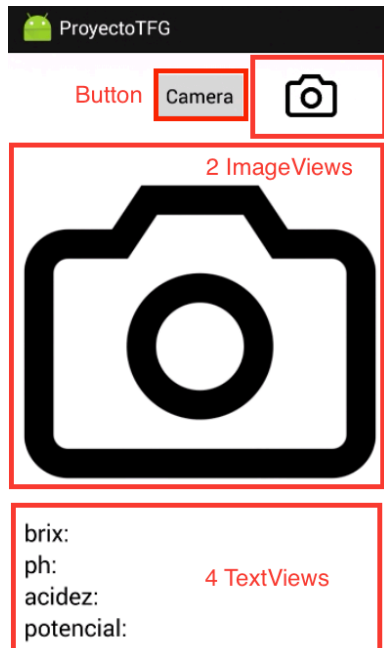
**Ilustración 9 - Librerías, Externas**

### 5.2.4 Recursos y vistas

En esta sección encontramos las imágenes y estilos que usa la aplicación. A continuación detallaremos las partes por las que se compone la vista de una única actividad de que disponemos.

La vista de la actividad esta organizada con un Relative Layout, que trata los elementos como hermanos en una posición relativa. Cada elemento se relaciona con uno conectado a él(“encima de”, “debajo de”, “a la derecha de”...) y con el elemento padre, en este caso el Relative Layout. El funcionamiento es en árbol, es decir, un elemento dentro de un Relative Layout podría ser otro Relative Layout y tener sus propios elementos internos.

Finalmente, basado en un Relative Layout, tenemos 7 elementos en la vista: Un “Button”, dos “ImageView” y cuatro “TextView”.



**Ilustración 10 - Recursos y vistas, Main**

El “Button”(Botón) tiene el texto “Camera”, que se encarga de llamar a la actividad de la cámara, la cual se explicará mas adelante.

Los dos “ImageView” muestran una imagen de una cámara inicialmente y, tras hacer la fotografía, el “ImageView” central muestra la fotografía original; mientras que el “ImageView” situado a la derecha del botón muestra el color seleccionado para el análisis. Dicho color esta seleccionado como el pixel central de la fotografía original.

Y para finalizar, los cuatro “TextView” muestran las características obtenidas según el color seleccionado. Las características a analizar son brix, pH, acidez y potencial.

### 5.2.5 Lógica de Aplicación

Se desarrolla dentro del directorio SRC del proyecto. En este caso, concentramos la lógica en un archivo llamado MainActivity.java, la lógica de aplicación de la actividad principal. A continuación se explicarán los métodos principales utilizados en el desarrollo, y unos métodos adicionales. Dichos métodos adicionales, aun sin estar incluidos en el funcionamiento de la aplicación, fueron desarrollados para maximizar los análisis de color, y permanecerán en la aplicación para posibles futuros avances.

### 5.2.5.1 Métodos principales

Método encargado de retornar un String con el nombre del álbum en el que se almacena la fotografía.

```
private String getAlbumName() {  
    return getString(R.string.album_name);  
}
```

Ilustración 11 - Métodos principales, getAlbumName

Método encargado de retornar un tipo File con la dirección de almacenamiento. Dicha dirección esta relacionada con el nombre del Álbum que entra por parámetros.

```
private File getAlbumStorageDir(String albumName) {  
    return new File (  
        Environment.getExternalStorageDirectory()  
        + CAMERA_DIR  
        + albumName  
    );  
}
```

Ilustración 12 - Métodos principales, getAlbumStorageDir

Método encargado de comprobar si esta creada la carpeta para almacenar las fotos, y si se tiene acceso a ella.

```
private File getAlbumDir() {  
    File storageDir = null;  
  
    if (Environment.MEDIA_MOUNTED.equals(Environment.getExternalStorageState())) {  
        storageDir = getAlbumStorageDir(getAlbumName());  
  
        if (!storageDir.mkdirs()) {  
            if (!storageDir.exists()) {  
                Log.e(TAG, "failed to create directory");  
                return null;  
            }  
        }  
    } else {  
        Log.e(TAG, "External storage is not mounted READ/WRITE.");  
    }  
  
    return storageDir;  
}
```

Ilustración 13 - Métodos principales, getAlbumDir

Método propio de una aplicación Android, sobrescrito para orientar la aplicación en modo vertical, inicializar las variables e insertar la conexión al clicar (onClick) del botón “Camera”. La presión de dicho botón lanzará el método “startCamera()”.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    mImageView = (ImageView) findViewById(R.id.imageView1);
    mImageColor = (ImageView) findViewById(R.id.imageView2);
    mImageBitmap = null;
    mImageColorBitmap = null;

    mIsError = false;

    // Declare and setup Implicit Activation button
    Button cameraButton = (Button) findViewById(R.id.cameraButton);
    cameraButton.setOnClickListener(new View.OnClickListener() {

        // Call startImplicitActivation() when pressed
        @Override
        public void onClick(View v) {

            Log.v(TAG, "Boton Presionado. Iniciando startCamera()");

            startCamera();

        }
    });
}
```

Ilustración 14 - Métodos principales, onCreate

Método encargado de crear el fichero para almacenar la fotografía. Para ello, utiliza la fecha y hora actual para el nombre, y la dirección donde almacenar la fotografía.

```
private File createImageFile() throws IOException {

    String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());
    String imageFileName = "IMG_" + timeStamp + "_";
    Log.d(TAG, imageFileName);
    File albumF = getAlbumDir();
    File imageF = File.createTempFile(imageFileName, ".jpg", albumF);
    mCurrentPhotoPath = imageF.getAbsolutePath();
    return imageF;
}
```

Ilustración 15 - Métodos principales, createImageFile

Método encargado de ejecutar la cámara del dispositivo. Se usa un Intent para invocar una aplicación ya existente en el dispositivo encargada de realizar una captura de imagen. Se crea el fichero en el que se almacenara la imagen y se lanza la actividad esperando los resultados de ella (“startActivityForResult”). Dichos resultados se tratan en el método “onActivityResult”.

```
private void startCamera(){  
  
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
  
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {  
  
        // Create the File where the photo should go  
        try {  
            Log.d(TAG, "Llamando a createImageFile()");  
            File photoFile = createImageFile();  
            mCurrentPhotoPath = photoFile.getAbsolutePath();  
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, Uri.fromFile(photoFile));  
        } catch (IOException ex) {  
            // Error occurred while creating the File  
  
        }  
  
        // Continue only if the File was successfully created  
  
        Log.d(TAG, "Insertando ruta para guardar la foto");  
        startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);  
  
    }  
  
}
```

Ilustración 16 - Métodos principales, startCamera

Método propio de una aplicación Android, sobrescrito para tratar los resultados obtenidos de la captura de la fotografía. Se comprueba que existe una imagen que guardar, que hay una dirección de almacenamiento para guardarla, y a continuación se añade la imagen a la galería del dispositivo y se lanza el método “setPic”, encargado de trabajar con dicha imagen.

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
  
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {  
  
        if (mCurrentPhotoPath != null) {  
            mLastPhotoPath = mCurrentPhotoPath;  
            setPic();  
            galleryAddPic();  
            mCurrentPhotoPath = null;  
        }  
  
    }  
  
}
```

Ilustración 17 - Métodos principales, onActivityResult

App para móviles de detección de características hortofrutícolas mediante tratamiento de imágenes

Método encargado de añadir a la galería la imagen obtenida, y crear un vínculo entre su posición en la galería y su localización de almacenamiento.

```
private void galleryAddPic() {
    Intent mediaScanIntent = new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE);
    File f = new File(mCurrentPhotoPath);
    Uri contentUri = Uri.fromFile(f);
    mediaScanIntent.setData(contentUri);
    this.sendBroadcast(mediaScanIntent);
}
```

Ilustración 18 - Métodos principales, galleryAddPic

Métodos encargados de almacenar y recuperar valores. El uso para nuestra aplicación es el almacenamiento de la dirección de la última fotografía tomada y sus medidas. Nos ayuda a recuperar la información en caso de cambiar la orientación del dispositivo.

```
// Some lifecycle callbacks so that the image can survive orientation change
@Override
protected void onSaveInstanceState(Bundle savedInstanceState) {
    //savedInstanceState.putParcelable(BITMAP_STORAGE_KEY, mImageBitmap);
    savedInstanceState.putString(PHOTO_PATH_STORAGE_KEY, mLastPhotoPath);
    savedInstanceState.putInt(TARGET_H_STORAGE_KEY, targetH);
    savedInstanceState.putInt(TARGET_W_STORAGE_KEY, targetW);
    super.onSaveInstanceState(savedInstanceState);
}

@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
    //mImageBitmap = savedInstanceState.getParcelable(BITMAP_STORAGE_KEY);
    mLastPhotoPath = savedInstanceState.getString(PHOTO_PATH_STORAGE_KEY);
    targetH = savedInstanceState.getInt(TARGET_H_STORAGE_KEY);
    targetW = savedInstanceState.getInt(TARGET_W_STORAGE_KEY);
    if(mLastPhotoPath==null) {
        Log.e(TAG, "mLastPhotoPath NULL");

        //mImageView.setImageBitmap(mImageBitmap);
        mCurrentPhotoPath = mLastPhotoPath;
        setPic();
        mCurrentPhotoPath = null;
    }
}
```

Ilustración 19 - Métodos principales, InstanceState

Método encargado de escalar la imagen al tamaño de la pantalla e insertarla en su imageView. Tras esto, el método “centralPixelColor” se encargará de continuar con la obtención del pixel central.

```
private void setPic() {  
  
    /* There isn't enough memory to open up more than a couple camera photos */  
    /* So pre-scale the target bitmap into which the file is decoded */  
  
    /* Get the size of the ImageView */  
    if(mImageView.getWidth() != 0 && mImageView.getHeight() != 0) {  
        targetW = mImageView.getWidth();  
        targetH = mImageView.getHeight();  
    }  
  
    /* Get the size of the image */  
    BitmapFactory.Options bmOptions = new BitmapFactory.Options();  
    bmOptions.inJustDecodeBounds = true;  
    BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);  
    int photoW = bmOptions.outWidth;  
    int photoH = bmOptions.outHeight;  
  
    /* Figure out which way needs to be reduced less */  
    int scaleFactor = 8;  
    if ((targetW > 0) || (targetH > 0)) {  
        scaleFactor = Math.max(photoW / targetW, photoH / targetH);  
    }  
  
    /* Set bitmap options to scale the image decode target */  
    bmOptions.inJustDecodeBounds = false;  
    bmOptions.inSampleSize = scaleFactor;  
    bmOptions.inPurgeable = true;  
  
    /* Decode the JPEG file into a Bitmap */  
    mImageBitmap = BitmapFactory.decodeFile(mCurrentPhotoPath, bmOptions);  
  
    //Central Pixel Color  
    centralPixelColor();  
  
    /* Associate the Bitmap to the ImageView */  
    mImageView.setImageBitmap(mImageBitmap);  
}
```

Ilustración 20 - Métodos principales, setPic

Método encargado de obtener el pixel central de la imagen. El color de dicho pixel se utilizara como fondo de imagen(background) en el imageView situado a la derecha del botón “Camera”. Se obtienen los valores “RGB” de la imagen, y se transforman en los valores “LCH” necesarios para las características de la fruta en tres pasos:

- Se convierten los valores “RGB” a “LAB” con el método “RGBtoLAB” de la librería “ColorUtils”.
- Se utilizan los algoritmos de conversión de “LAB” de la cámara a “LAB” del colorímetro.
- Se convierten los valores “LAB” obtenidos en valores “LCH” con el método “LABtoLCH”

```
private void centralPixelColor() {  
    int widthImage = mImageBitmap.getWidth();  
    int heightImage = mImageBitmap.getHeight();  
    int[] pixelsImage = new int[widthImage * heightImage];  
    mImageBitmap.getPixels(pixelsImage, 0, widthImage, 0, 0, widthImage, heightImage);  
  
    int central = ((heightImage/2) * widthImage) + (widthImage/2);  
  
    //Color of central pixel of the pic  
    int colorPixel = pixelsImage[central];  
  
    //Fill the small image with the color of the central pixel  
    mImageColor.setBackgroundColor(colorPixel);  
  
    int redValue = Color.red(colorPixel);  
    int greenValue = Color.green(colorPixel);  
    int blueValue = Color.blue(colorPixel);  
  
    //RGB to LAB color  
    double[] outLAB = new double[3];  
    ColorUtils.RGBToLAB(redValue, greenValue, blueValue, outLAB);  
  
    //Conversion CAMERA PHONE to COLORIMETRO  
    double[] outLAB2 = new double[3];  
  
    outLAB2[0] = algorithmConversionLAB(1, outLAB[0], outLAB[1], outLAB[2]);  
    outLAB2[1] = algorithmConversionLAB(2, outLAB[0], outLAB[1], outLAB[2]);  
    outLAB2[2] = algorithmConversionLAB(3, outLAB[0], outLAB[1], outLAB[2]);  
}
```

Ilustración 21 - Métodos principales, centralPixelColor 1/2

Dichos valores “LCH” son utilizados para calcular las características “brix”, “pH”, “acidez” y “potencial” de la fruta.

Finalmente, las características obtenidas se insertan en los textViews para que aparezcan en la pantalla del dispositivo.



```
//LBA to LCH color
double[] outLCH = LBAtoLCH(outLAB2);

//Formula of conversion
double brix = 1.7413 * ((180-outLCH[2])/(outLCH[0] + outLCH[1])) + 10.894;
double ph = 0.1822 * ((180-outLCH[2])/(outLCH[0] + outLCH[1])) + 2.5931;
double acidez = -2.5272 * ((180-outLCH[2])/(outLCH[0] + outLCH[1])) + 27.201;
double potencial = 80.73 * ((180-outLCH[2])/(outLCH[0] * outLCH[1])) + 92.886;

Log.e(TAG, "centralPixelColor LAB: " + outLAB2[0] + " " + outLAB2[1] + " " + outLAB2[2]);

TextView textView5 = (TextView) findViewById(R.id.textView5);
TextView textView6 = (TextView) findViewById(R.id.textView6);
TextView textView7 = (TextView) findViewById(R.id.textView7);
TextView textView8 = (TextView) findViewById(R.id.textView8);

textView5.setText(String.format("brix: %.2f", brix));
textView6.setText(String.format("ph: %.2f", ph));
textView7.setText(String.format("acidez: %.2f", acidez));
textView8.setText(String.format("potencial: %.2f", potencial));
```

Ilustración 22 - Métodos principales, centralPixelColor 2/2

Método encargado de convertir los valores “LAB” obtenidos por la cámara en valores “LAB” similares a los obtenidos por el colorímetro. Se retorna el valor “L”, “A” o “B” dependiendo del valor “i” insertado por parámetros.

```
private double algorithmConversionLAB(int i, double x1, double x2, double x3) {
    double value = 0;

    double LValue = -5.6433542 * Math.pow(10,-5) * Math.pow(x1,4) - 2.729373046 * Math.pow(10,-6) * Math.pow(x1,3) * x2
    - 2.542036479 * Math.pow(10,-5) * Math.pow(x1,3) * x3 - 8.692019067 * Math.pow(10,-6) * Math.pow(x1,2) * Math.pow(x2,2)
    - 7.260781609 * Math.pow(10,-5) * Math.pow(x1,2) * x2 * x3
    + 2.125378615 * Math.pow(10,-5) * Math.pow(x1,2) * Math.pow(x3,2) - 3.140729563 * Math.pow(10,-6) * x1 * Math.pow(x2,3)
    + 1.15675544 * Math.pow(10,-5) * x1 * Math.pow(x2,2) * x3 + 3.764521033 * Math.pow(10,-5) * x1 * x2 * Math.pow(x3,2)
    - 1.296331606 * Math.pow(10,-5) * x1 * Math.pow(x3,3) + 2.189807162 * Math.pow(10,-7) * Math.pow(x2,4)
    + 5.424451046 * Math.pow(10,-7) * Math.pow(x2,3) * x3 + 9.213962171 * Math.pow(10,-6) * Math.pow(x2,2) * Math.pow(x3,2)
    - 7.53054916 * Math.pow(10,-6) * x2 * Math.pow(x3,3) + 2.151807588 * Math.pow(10,-6) * Math.pow(x3,4)
    + 8.964967317 * Math.pow(10,-3) * Math.pow(x1,3) + 1.601962232 * Math.pow(10,-3) * Math.pow(x1,2) * x2
    + 4.036057663 * Math.pow(10,-3) * Math.pow(x1,2) * x3 - 8.747822934 * Math.pow(10,-5) * x1 * Math.pow(x2,2)
    + 5.353660233 * Math.pow(10,-3) * x1 * x2 * x3 - 2.678962354 * Math.pow(10,-3) * x1 * Math.pow(x3,2)
    + 1.293994576 * Math.pow(10,-4) * Math.pow(x2,3) - 8.673035123 * Math.pow(10,-4) * Math.pow(x2,2) * x3
    - 1.864793144 * Math.pow(10,-3) * x2 * Math.pow(x3,2) + 7.448971447 * Math.pow(10,-4) * Math.pow(x3,3)
    - 4.76337062 * Math.pow(10,-1) * Math.pow(x1,2) - 1.029182322 * Math.pow(10,-1) * x1 * x2
    - 1.960291418 * Math.pow(10,-1) * x1 * x3 + 2.642941511 * Math.pow(10,-2) * Math.pow(x2,2)
    - 8.160914945 * Math.pow(10,-2) * x2 * x3 + 7.881707427 * Math.pow(10,-2) * Math.pow(x3,2) + 11.09559613 * x1
    + 1.438869318 * x2 + 2.882452371 * x3 - 71.04844428;
```

Ilustración 23 - Métodos principales, algorithmConversionLAB 1/3

```
double AValue = 4.002696867 * Math.pow(10,-5) * Math.pow(x1,4) - 7.791370788 * Math.pow(10,-5) * Math.pow(x1,3) * x2
- 1.610935814 * Math.pow(10,-5) * Math.pow(x1,3) * x3 + 4.972142022 * Math.pow(10,-6) * Math.pow(x1,2) * Math.pow(x2,2)
+ 1.325464287 * Math.pow(10,-4) * Math.pow(x1,2) * x2 * x3 + 1.692922586 * Math.pow(10,-5) * Math.pow(x1,2) * Math.pow(x3,2)
+ 5.429026582 * Math.pow(10,-8) * x1 * Math.pow(x2,3) - 1.54822404 * Math.pow(10,-5) * x1 * Math.pow(x2,2) * x3
- 3.439066986 * Math.pow(10,-5) * x1 * x2 * Math.pow(x3,2) - 1.965814637 * Math.pow(10,-5) * x1 * Math.pow(x3,3)
+ 1.001576885 * Math.pow(10,-6) * Math.pow(x2,4) + 5.961740568 * Math.pow(10,-6) * Math.pow(x2,3) * x3
+ 4.536510191 * Math.pow(10,-6) * Math.pow(x2,2) * Math.pow(x3,2) + 4.301046432 * Math.pow(10,-6) * x2 * Math.pow(x3,3)
+ 4.721310113 * Math.pow(10,-6) * Math.pow(x3,4) - 7.622999388 * Math.pow(10,-3) * Math.pow(x1,3)
+ 6.700679003 * Math.pow(10,-3) * Math.pow(x1,2) * x2 + 2.966467148 * Math.pow(10,-3) * Math.pow(x1,2) * x3
- 1.644213219 * Math.pow(10,-4) * x1 * Math.pow(x2,2) - 9.47190348 * Math.pow(10,-3) * x1 * x2 * x3
- 1.943958629 * Math.pow(10,-3) * x1 * Math.pow(x3,2) - 3.201487341 * Math.pow(10,-4) * Math.pow(x2,3)
+ 5.586738105 * Math.pow(10,-4) * Math.pow(x2,2) * x3 + 8.176201035 * Math.pow(10,-4) * x2 * Math.pow(x3,2)
+ 9.973249911 * Math.pow(10,-4) * Math.pow(x3,3) + 5.176919871 * Math.pow(10,-1) * Math.pow(x1,2)
- 1.468717118 * Math.pow(10,-1) * x1 * x2 - 1.465238811 * Math.pow(10,-1) * x1 * x3
+ 6.478583279 * Math.pow(10,-4) * Math.pow(x2,2) + 1.577483565 * Math.pow(10,-1) * x2 * x3
+ 4.815754326 * Math.pow(10,-2) * Math.pow(x3,2) - 14.82144951 * x1 + 1.599301155 * x2 + 1.752270734 * x3
+ 150.8798357;
```

Ilustración 24 - Métodos principales, algorithmConversionLAB 2/3

```

double BValue = -3.987282478 * Math.pow(10,-5) * Math.pow(x1,4) + 3.50449784 * Math.pow(10,-5) * Math.pow(x1,3) * x2
+ 1.16105166 * Math.pow(10,-5) * Math.pow(x1,3) * x3 + 2.856734291 * Math.pow(10,-5) * Math.pow(x1,2) * Math.pow(x2,2)
- 7.845264056 * Math.pow(10,-5) * Math.pow(x1,2) * x2 * x3 - 3.830178777 * Math.pow(10,-5) * Math.pow(x1,2) * Math.pow(x2,2)
+ 1.006685516 * Math.pow(10,-5) * x1 * Math.pow(x2,3) + 3.84620005 * Math.pow(10,-6) * x1 * Math.pow(x2,2) * x3
+ 2.187218112 * Math.pow(10,-5) * x1 * x2 * Math.pow(x3,2) + 5.747316452 * Math.pow(10,-5) * x1 * Math.pow(x3,3)
+ 1.397349313 * Math.pow(10,-6) * Math.pow(x2,4) - 5.529649842 * Math.pow(10,-6) * Math.pow(x2,3) * x3
- 4.105789396 * Math.pow(10,-6) * Math.pow(x2,2) * Math.pow(x3,2) - 1.460175974 * Math.pow(10,-6) * x2 * Math.pow(x3,3)
- 1.999337441 * Math.pow(10,-5) * Math.pow(x3,4) + 7.923684658 * Math.pow(10,-3) * Math.pow(x1,3)
- 2.077820723 * Math.pow(10,-3) * Math.pow(x1,2) * x2 - 4.784477637 * Math.pow(10,-3) * Math.pow(x1,2) * x3
- 2.206411379 * Math.pow(10,-3) * x1 * Math.pow(x2,2) + 4.350142517 * Math.pow(10,-3) * x1 * x2 * x3
+ 5.245497757 * Math.pow(10,-3) * x1 * Math.pow(x3,2) - 1.689368 * Math.pow(10,-4) * Math.pow(x2,3)
- 2.939370635 * Math.pow(10,-4) * Math.pow(x2,2) * x3 + 1.410184069 * Math.pow(10,-4) * x2 * Math.pow(x3,2)
- 2.854511738 * Math.pow(10,-3) * Math.pow(x3,3) - 5.535855334 * Math.pow(10,-1) * Math.pow(x1,2)
- 1.909512648 * Math.pow(10,-2) * x1 * x2 + 2.949787353 * Math.pow(10,-1) * x1 * x3
+ 3.189695502 * Math.pow(10,-2) * Math.pow(x2,2) - 4.277717828 * Math.pow(10,-2) * x2 * x3
- 1.298304627 * Math.pow(10,-1) * Math.pow(x3,2) + 15.90785368 * x1 + 1.433060643 * x2 - 3.375125386 * x3 - 153.8029171;

switch (i){
    case 1 :    value = LValue;
                break;
    case 2 :    value = AValue;
                break;
    default:    value = BValue;
                break;
}

return value;

```

Ilustración 25 - Métodos principales, algorithmConversionLAB 3/3

Método encargado de convertir los valores “LAB” en valores “LCH”.

```

private double[] LBAtoLCH(double[] outLAB) {

    double[] outLCH = new double[3];

    //L = L
    outLCH[0] = outLAB[0];

    //C = (a^2 + b^2) ^ (1/2)
    outLCH[1] = Math.pow(Math.pow(outLAB[1], 2) + Math.pow(outLAB[2], 2), 0.5);

    //H = 1/(tan(b/a)) in radians, and we need degrees.
    outLCH[2] = Math.toDegrees(Math.atan2(outLAB[2],outLAB[1]));

    while (outLCH[2] >= 360) outLCH[2] = outLCH[2] - 360;

    while (outLCH[2] < 0) outLCH[2] = outLCH[2] + 360;

    return outLCH;
}

```

Ilustración 26 - Métodos principales, LBAtoLCH

### 5.2.5.2 Métodos adicionales

Método encargado de orientar la fotografía tomada en imageView central. Soluciona un pequeño problema que tienen algunos dispositivos, que no almacenan la foto rotada como se tomó, sino que almacenan la rotación en los datos Exif de la foto.

```
private void fixOrientation() {
    try {
        ExifInterface ei = new ExifInterface(mCurrentPhotoPath);
        int orientation = ei.getAttributeInt(ExifInterface.TAG_ORIENTATION, ExifInterface.ORIENTATION_NORMAL);

        switch(orientation) {
            case ExifInterface.ORIENTATION_ROTATE_90:
                mImageView.setRotation(90);
                break;
            case ExifInterface.ORIENTATION_ROTATE_180:
                mImageView.setRotation(180);
                break;
            case ExifInterface.ORIENTATION_ROTATE_270:
                mImageView.setRotation(270);
                break;
            default: break;
        }
        //mImageView.setVisibility(View.VISIBLE);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

Ilustración 27 - Métodos adicionales, fixOrientation

Métodos encargados de comprobar y liberar los bitmaps cargados y sobrescritos por nuevos bitmaps. Comprueba que no están en uso y los recicla para la fluidez de la aplicación.

```
public void ImageProcessor(final Bitmap image) {
    mImage = image.copy(image.getConfig(), image.isMutable());
    if(mImage == null) {
        mIsError = true;
    }
}

public boolean isError() {
    return mIsError;
}

public void setImage(final Bitmap image) {
    mImage = image.copy(image.getConfig(), image.isMutable());
    if(mImage == null) {
        mIsError = true;
    } else {
        mIsError = false;
    }
}

public Bitmap getImage() {
    if(mImage == null){
        return null;
    }
    return mImage.copy(mImage.getConfig(), mImage.isMutable());
}

public void free() {
    if(mImage != null && !mImage.isRecycled()) {
        mImage.recycle();
        mImage = null;
    }
}
```

Ilustración 28 - Métodos adicionales, metodos Bitmap

Para el uso de los métodos que veremos a continuación, se requiere realizar la fotografía de la fruta, únicamente situándola sobre un folio en blanco. Se obtendrá el contorno y se analizarán todos los píxeles que componen la fruta.

Como se explicó anteriormente, estos métodos no son utilizados en la aplicación, por lo que están comentados.

Como podemos observar, en análisis de la imagen consta de 5 pasos:

- Se utiliza un filtro de la librería GPUImage con detección de bordes para remarcar el contorno de la fruta.
- Se reemplaza los colores a blanco y negro puro para obtener el contorno.
- Se procesa el contorno obteniendo los puntos de dicho contorno.
- Sobre la fotografía inicial, se transforman todos los píxeles exteriores al contorno en píxel blanco.
- Finalmente se crea un histograma con los píxeles de la fruta, despreciando los píxeles blancos de la imagen.

```
/*
//EDGEDETECTOR
GPUImage mGPUImage = new GPUImage(this);
mGPUImage.setFilter(new GPUImageSobelEdgeDetection());
mImageBitmapProcess = mGPUImage.getBitmapWithFilterApplied(mImageBitmap);

ImageProcessor(mImageBitmapProcess);
mImageBitmapProcess = replaceColor (Color.BLACK, Color.WHITE);
//free();

processEdge(Color.BLACK, Color.WHITE);

mImageBitmapProcess = eraseBackground();

mImageBitmapProcess = createHistogram();
//createHistogram();

free();
*/
```

Ilustración 29 - Métodos adicionales, modificaciones adicionales

Tras utilizar un filtro de bordes en la imagen, se obtiene una imagen en escala de grises con los contornos un poco remarcados. Para facilitar la obtención del

contorno, el método se encarga de transformar en blanco los pixeles con tonos mas blancos, y en negro los pixeles mas oscuros.

```
public Bitmap replaceColor(int fromColor, int targetColor) {  
    Log.e(TAG, "replaceColor");  
  
    if(mImage == null) {  
        return null;  
    }  
  
    int width = mImage.getWidth();  
    int height = mImage.getHeight();  
    int[] pixels = new int[width * height];  
    mImage.getPixels(pixels, 0, width, 0, 0, width, height);  
  
    Log.e(TAG, String.valueOf(pixels.length));  
  
    for(int x = 0; x < pixels.length; ++x) {  
        if(x<10) Log.e(TAG, String.valueOf(pixels[x]));  
        pixels[x] = (pixels[x] < fromColor+2500000) ? fromColor : targetColor;  
        if(x<10) Log.e(TAG, String.valueOf(pixels[x]));  
    }  
  
    Bitmap newImage = Bitmap.createBitmap(width, height, mImage.getConfig());  
    newImage.setPixels(pixels, 0, width, 0, 0, width, height);  
  
    return newImage;  
}
```

Ilustración 30 - Métodos adicionales, replaceColor

Método encargado de recolectar los puntos del contorno de la fruta. Para ello, analiza la fotografía horizontalmente fila a fila en dos pasos:

- De izquierda a derecha hasta encontrar el primer punto de contorno.
- De derecha a izquierda hasta encontrar el primer punto de contorno.

Dichos puntos se almacenan en una matriz llamada “edges”.

```
private void processEdge(int fromColor, int targetColor){
    Log.e(TAG, "processEdge");

    boolean primerBlanco;

    int width = mImageBitmapProcess.getWidth();
    int height = mImageBitmapProcess.getHeight();
    int[] pixels = new int[width * height];
    mImageBitmapProcess.getPixels(pixels, 0, width, 0, 0, width, height);
    int pos=0;
    boolean first = false;
    boolean last = false;
    int y =0;
    for(int x = 0; x < height; x++) {
        first=false;
        last=false;

        y = 0;
        while ((!first)&&(y < width)){
            pos = (x * width) + y;
            if(pixels[pos] == targetColor){
                edges[x][0]=y;
                first=true;
            }
            y++;
        }
        //Si recorre toda la fila sin encontrar un punto blanco, entonces no hace falta recorrerla a la inversa.
        if(!first){
            edges[x][0] = -1;
            edges[x][1] = -1;
        }
        else{
            y = width;
            while ((!last)&&(y > 0)){
                y--;
                pos = (x * width) + y;
                if(pixels[pos] == targetColor){
                    edges[x][1]=y;
                    last=true;
                }
            }
        }
    }
}
```

Ilustración 31 - Métodos adicionales, processEdge

Tras obtener la matriz de los puntos del contorno de la fruta, este método se encarga de volver a la foto original y convertir en pixel blanco todos los pixeles fuera del contorno de la fruta.

```
private Bitmap eraseBackground(){
    Log.e(TAG, "eraseBackground");

    int width = mImageBitmap.getWidth();
    int height = mImageBitmap.getHeight();
    int[] pixels = new int[width * height];
    mImageBitmap.getPixels(pixels, 0, width, 0, 0, width, height);

    int pos = 0;

    for(int x = 0; x < height; x++) {
        for (int y = 0; y < width; y++) {
            if((y < edges[x][0]) || (y > edges[x][1])){
                pos = (x * width) + y;
                pixels[pos] = Color.WHITE;
            }
        }
    }

    Bitmap newImage = Bitmap.createBitmap(width, height, mImage.getConfig());
    newImage.setPixels(pixels, 0, width, 0, 0, width, height);

    return newImage;
}
```

Ilustración 32 - Métodos adicionales, eraseBackground

Método encargado de crear un histograma con la imagen final. Omitiendo los píxeles blancos, se obtiene el valor entero máximo y mínimo de los píxeles de la imagen.

```
private Bitmap createHistogram(){
    int widthImage = mImageBitmapProcess.getWidth();
    int heightImage = mImageBitmapProcess.getHeight();
    int[] pixelsImage = new int[widthImage * heightImage];
    mImageBitmapProcess.getPixels(pixelsImage, 0, widthImage, 0, 0, widthImage, heightImage);

    int numParts = 25;

    int[] columns = new int[numParts];
    for (int x=0; x<numParts; x++){
        columns[x] = 0;
    }

    int menor = 0;
    int mayor = 0;

    for(int x = 0; x < pixelsImage.length; ++x) {
        if(pixelsImage[x] != Color.WHITE){
            if(menor == 0){
                menor = pixelsImage[x];
                mayor = pixelsImage[x];
            }
            else{
                if(pixelsImage[x] < menor){
                    menor = pixelsImage[x];
                }

                if(pixelsImage[x] > mayor){
                    mayor = pixelsImage[x];
                }
            }
        }
    }
}
```

Ilustración 33 - Métodos adicionales, createHistogram 1/3



Se obtiene el intervalo entre el valor entero mayor y menor y se vuelve a particionar en intervalos mas pequeños para la gráfica de barras. Se analizan de nuevo todos los pixeles de la imagen, contabilizando los pixeles de cada intervalo.

```
int interval = mayor - menor;
int smallInterval = interval/numParts;

int intervalnumber = 0;
int intervalPart = 0;
int cont = 0;
for(int x = 0; x < pixelsImage.length; ++x) {

    if(pixelsImage[x] != Color.WHITE){
        intervalnumber = pixelsImage[x]-mayor;
        intervalPart = intervalnumber / -(smallInterval);
        if(intervalPart==numParts) intervalPart--;
        columns[intervalPart]++;
    }
}
int maxCol = 0;
for (int x=0; x<numParts; x++){
    Log.e(TAG, "Columna "+x+": "+String.valueOf(columns[x]));
    if (columns[x] > maxCol){
        maxCol = columns[x];
    }
}

int widthHistogram = targetW;
int heightHistogram = targetH;
int[] pixelsHistogram = new int[widthHistogram * heightHistogram];

int maxPosible = maxCol / widthHistogram;
if (maxPosible < 1) maxPosible = 1;
int columnasPintadas = (heightHistogram / numParts) - 1;

int colorColumn = mayor - smallInterval/2;
```

Ilustración 34 - Métodos adicionales, createHistogram 2/3

Finalmente se muestra la cantidad de cada intervalo de pixeles en una gráfica de barras, cuyo color de barra viene dado por la media de los colores de los pixeles que representa.

```
int pos = 0;
for (int x=0; x<numParts; x++){
    for (int y = 0; y<columns[x]/maxPosible; y++){

        for (int z = 0; z<columnasPintadas; z++){
            pos = ((x*columnasPintadas+z) * widthHistogram) + y;
            pixelsHistogram[pos] = colorColumn;
        }
    }
    colorColumn = colorColumn - smallInterval;
}

Bitmap.Config conf = Bitmap.Config.ARGB_8888;
Bitmap newImage = Bitmap.createBitmap(widthHistogram, heightHistogram, conf);
newImage.setPixels(pixelsHistogram, 0, widthHistogram, 0, 0, widthHistogram, heightHistogram);

return newImage;
```

Ilustración 35 - Métodos adicionales, createHistogram 3/3

### 5.3 Registros de color

Necesitamos obtener unas formulas de conversión de los valores de la cámara utilizada por la aplicación y los valores del colorímetro. Para ello, utilizamos un misma plantilla de colores.

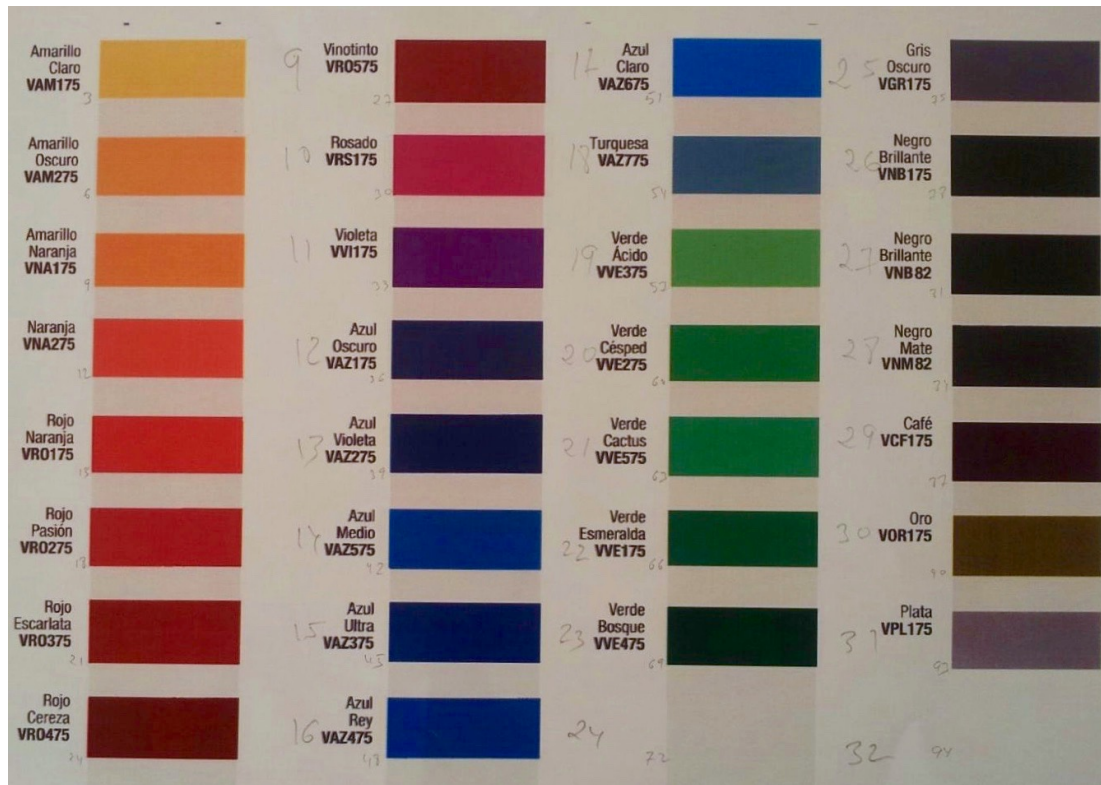


Ilustración 36 - Plantilla de colores

Obtenemos primero los valores con el colorímetro. Se realizan 3 tomas de cada color de la plantilla, comenzando por la esquina superior izquierda, siguiendo un orden vertical hacia abajo, seguido de un orden horizontal hacia la derecha(ANEXO 1).

El siguiente paso es obtener los valores de la cámara de la aplicación. Para que las tomas de los valores tengan unas características genéricas, se toman recibiendo luz directa solar y a 8 cm del objetivo, para facilitar el enfoque.

En primera instancia se obtienen los valores “RGB” de la plantilla de colores(ANEXO 2). Los distintos modos de color de las tomas (“RGB” y “Lab”) nos plantean la idea de convertir primero los valores “RGB” a “Lab”. Tras convertir los valores, se realiza una toma de valores de la plantilla de colores, esta vez con 5 tomas por color, para aumentar la aproximación(ANEXO 3).

Finalmente, tenemos los valores “Lab” del colorímetro y de la cámara de la aplicación. A continuación utilizamos una herramienta online[7] para obtener las formulas de conversión a través de una regresión polinomial múltiple[8]. El coeficiente de determinación[9] determina la calidad del modelo para replicar resultados. Dichos coeficientes oscilan entre 0 y 1. Para cada variable del modo de color “Lab” se genera una formula con unos coeficientes de determinación(ANEXO 4).

#### 5.4 Características de la fruta

El último paso a tomar para obtener las características de la fruta es insertar las formulas desarrolladas por los ingenieros, como explicamos en los antecedentes. Dichas formulas necesitan valores de color en el modo de color “LCH”, por lo que se convierten los valores de “LAB” obtenidos desde la cámara del dispositivo.

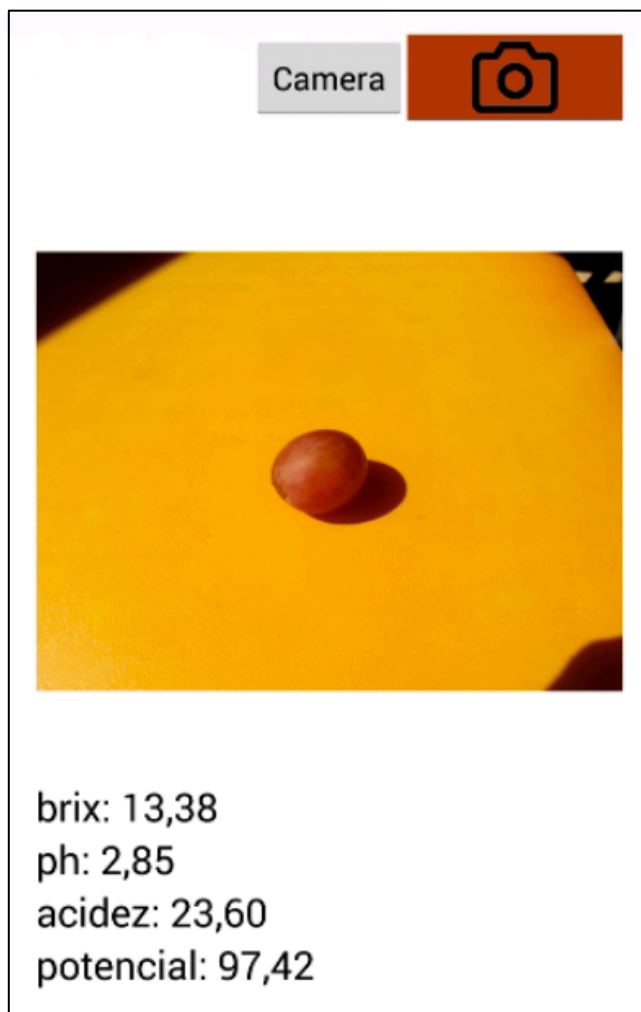


Ilustración 37 - Características de la fruta

Las características obtenidas son Brix, pH, acidez y potencial (Ilustración 10). Dichas características se calculan a través del color del pixel central de la fotografía tomada. El tono de color original obtenido por la cámara se puede observar en el “ImageView” situado a la derecha del botón “Camera” (Ilustración 10).

## **6. RESULTADOS Y DISCUSIÓN**

Podemos centrar nuestros resultados en la satisfactoria generación de formulas de conversión. Como podemos observar(ANEXO 4), los coeficientes de determinación, cuyo propósito es predecir los valores generados por las nuevas formulas, son aproximados a 1. Esto nos indica la calidad de dichas formulas, y que la aproximación obtenida es mayor a la esperada en los objetivos.

Las configuraciones en la toma de fotografías aumenta la aproximación, pero limita su ejecución a días soleados con un dispositivo determinado. La variedad de dispositivos Android y sus distintas cámaras y enfoques podría dificultar la obtención de una única formula de conversión. Aun así, se puede afirmar que los resultados son satisfactorios.

## **7. CONCLUSIONES**

Un proyecto complejo, y con muchos frentes abiertos, como son una aplicación Android, el análisis fotográfico de frutas, y los valores generados por un colorímetro.

Android es un sistema con una progresión exponencial. Proporciona unas habilidades al alcance de la mano que permiten alcanzar cualquier meta. La tecnología crece de tal forma que la mayoría de usuarios de Smartphone desconoce el potencial del que disponen.

Los distintos modos de color utilizados fue un reto a conquistar como hijo de fotógrafo que soy. Esa variedad de opciones a la hora de definir el color se asemeja a la variedad de idiomas a la hora de pronunciar una palabra.

La gratificación de los resultados obtenidos ayuda a entender en la conversión entre valores. Todo es transformable y adaptable con un ligero margen de error. Un dispositivo(el colorímetro), que yo considero como “caro”, se puede transformar en un teléfono de bolsillo para dar un servicio mas rápido y específico a un usuario final.

## REFERENCIAS BIBLIOGRÁFICAS

Kit L. Yam, Spyridon E. Papadakis, A simple digital imaging method for measuring and analyzing color of food surfaces, *Journal of Food Engineering*, (2004): 137-142.

Torres, Ramiro, et al. "Relación del color y del estado de madurez con las propiedades fisicoquímicas de frutas tropicales." *Información tecnológica* 24.3 (2013): 51-56.

## BIBLIOGRAFÍA

- [1] <https://developer.android.com/index.html>
- [2] <https://developer.android.com/studio/index.html>
- [3] [http://www.gsmarena.com/samsung\\_i9300\\_galaxy\\_s\\_iii-4238.php](http://www.gsmarena.com/samsung_i9300_galaxy_s_iii-4238.php)
- [4] <http://www.konicaminolta.com.cn/instruments/download/manual/pdf/CR-300.pdf>
- [5] <https://github.com/CyberAgent/android-gpuimage/tree/master>
- [6] <https://developer.android.com/reference/android/support/v4/app/package-summary.html>
- [7] <http://www.xuru.org/rt/MPR.asp>
- [8] [https://es.wikipedia.org/wiki/Regresion\\_no\\_lineal](https://es.wikipedia.org/wiki/Regresion_no_lineal)
- [9] [https://es.wikipedia.org/wiki/Coeficiente\\_de\\_determinacion](https://es.wikipedia.org/wiki/Coeficiente_de_determinacion)
- [10] <http://yuml.me/>



## ANEXOS

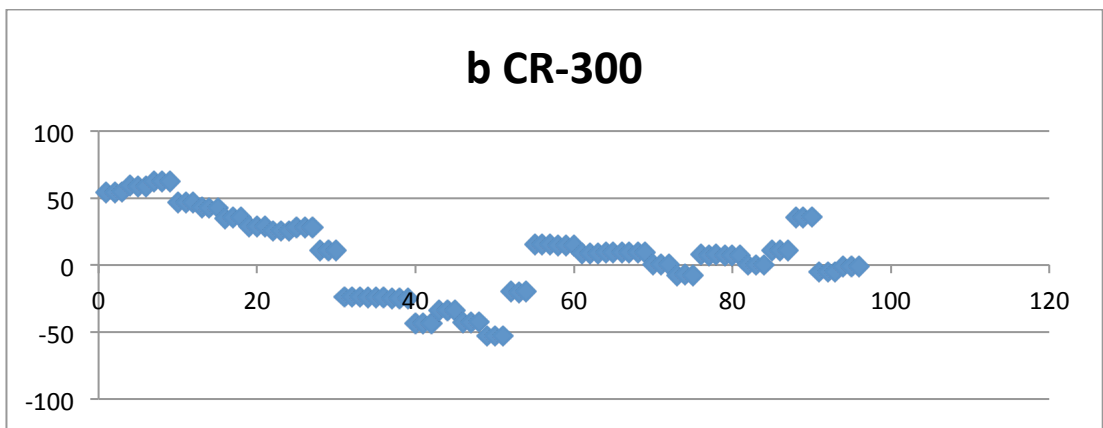
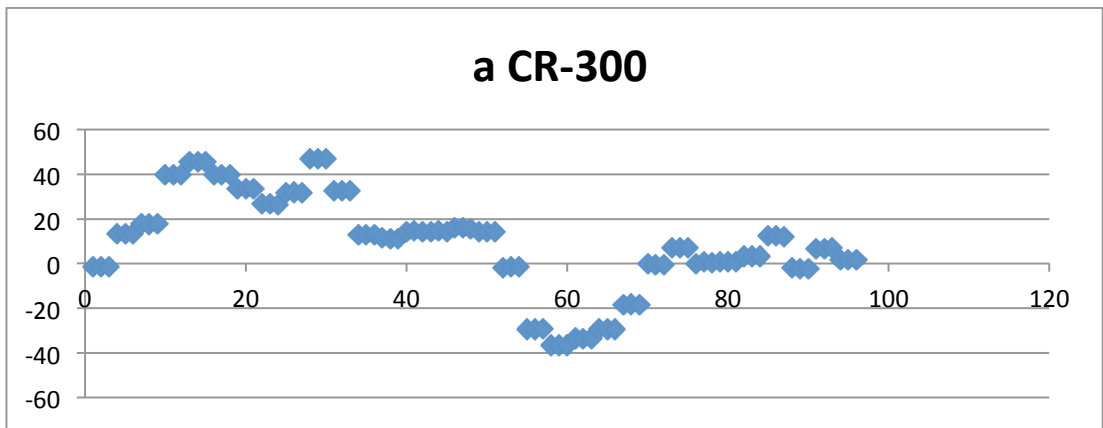
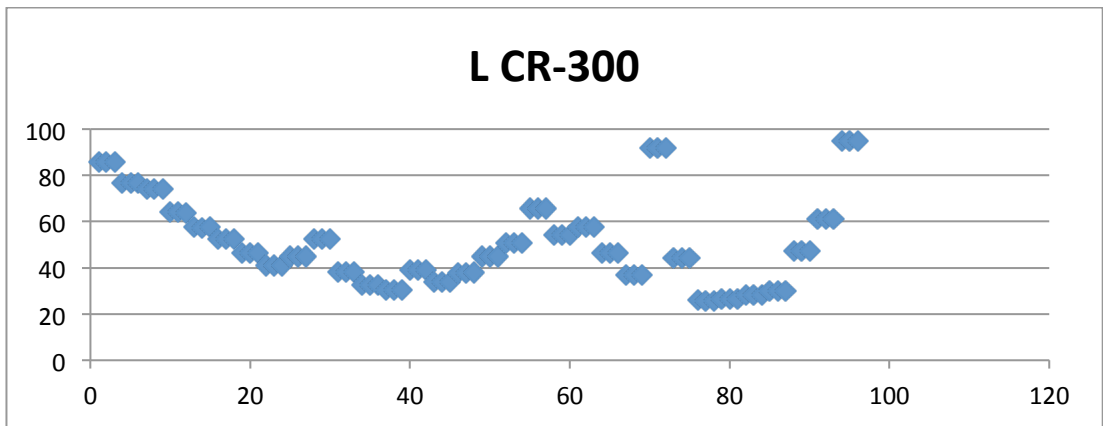
### Anexo 1 – Tomas Colorímetro “Lab”

| Nº Lectura | Color nº | Rep | L CR-300 | a CR-300 | b CR-300 |
|------------|----------|-----|----------|----------|----------|
| 1          | 1        | 1   | 85,58    | -1,66    | 54,35    |
| 2          | 1        | 2   | 85,57    | -1,67    | 54,28    |
| 3          | 1        | 3   | 85,58    | -1,74    | 54,46    |
| 4          | 2        | 1   | 76,83    | 13,28    | 59,29    |
| 5          | 2        | 2   | 76,79    | 13,38    | 59,11    |
| 6          | 2        | 3   | 76,81    | 13,39    | 59,03    |
| 7          | 3        | 1   | 74,22    | 17,72    | 62,17    |
| 8          | 3        | 2   | 74,22    | 17,71    | 62,17    |
| 9          | 3        | 3   | 74,22    | 17,83    | 62,15    |
| 10         | 4        | 1   | 63,87    | 39,71    | 46,67    |
| 11         | 4        | 2   | 63,87    | 39,53    | 46,99    |
| 12         | 4        | 3   | 63,72    | 39,58    | 46,68    |
| 13         | 5        | 1   | 57,54    | 45,36    | 42,98    |
| 14         | 5        | 2   | 57,48    | 45,43    | 42,61    |
| 15         | 5        | 3   | 57,49    | 45,39    | 42,69    |
| 16         | 6        | 1   | 52,44    | 39,78    | 35,12    |
| 17         | 6        | 2   | 52,51    | 39,54    | 35,45    |
| 18         | 6        | 3   | 52,46    | 39,65    | 35,68    |
| 19         | 7        | 1   | 46,22    | 33,54    | 28,62    |
| 20         | 7        | 2   | 46,29    | 33,31    | 28,8     |
| 21         | 7        | 3   | 46,28    | 33,3     | 28,98    |
| 22         | 8        | 1   | 41,01    | 26,87    | 25,46    |
| 23         | 8        | 2   | 41,07    | 26,71    | 25,37    |
| 24         | 8        | 3   | 41,1     | 26,47    | 25,63    |
| 25         | 9        | 1   | 44,81    | 31,74    | 28       |
| 26         | 9        | 2   | 44,76    | 31,81    | 27,75    |
| 27         | 9        | 3   | 44,8     | 31,68    | 27,78    |
| 28         | 10       | 1   | 52,39    | 46,82    | 11       |
| 29         | 10       | 2   | 52,38    | 46,86    | 11,11    |
| 30         | 10       | 3   | 52,42    | 46,7     | 11,33    |
| 31         | 11       | 1   | 38,3     | 32,68    | -23,65   |
| 32         | 11       | 2   | 38,3     | 32,79    | -23,65   |
| 33         | 11       | 3   | 38,33    | 32,67    | -23,65   |
| 34         | 12       | 1   | 32,59    | 12,91    | -24,25   |
| 35         | 12       | 2   | 32,51    | 13,09    | -24,27   |
| 36         | 12       | 3   | 32,59    | 13,06    | -24,26   |
| 37         | 13       | 1   | 30,55    | 11,63    | -25,14   |
| 38         | 13       | 2   | 30,54    | 11,37    | -24,75   |
| 39         | 13       | 3   | 30,55    | 11,37    | -24,9    |
| 40         | 14       | 1   | 39,01    | 14,02    | -43,15   |
| 41         | 14       | 2   | 39,07    | 14,57    | -43,34   |
| 42         | 14       | 3   | 39,12    | 14,16    | -43,14   |
| 43         | 15       | 1   | 33,78    | 14,36    | -33,86   |
| 44         | 15       | 2   | 33,71    | 14,59    | -34,02   |

App para móviles de detección de características hortofrutícolas mediante tratamiento de imágenes

|    |    |   |       |        |        |
|----|----|---|-------|--------|--------|
| 45 | 15 | 3 | 33,76 | 14,19  | -33,86 |
| 46 | 16 | 1 | 37,8  | 15,78  | -42,83 |
| 47 | 16 | 2 | 37,8  | 15,84  | -42,88 |
| 48 | 16 | 3 | 37,84 | 15,51  | -42,6  |
| 49 | 17 | 1 | 44,86 | 14,33  | -52,73 |
| 50 | 17 | 2 | 44,96 | 14,33  | -52,85 |
| 51 | 17 | 3 | 44,99 | 14,13  | -52,76 |
| 52 | 18 | 1 | 50,74 | -1,8   | -20,13 |
| 53 | 18 | 2 | 50,75 | -1,71  | -20,26 |
| 54 | 18 | 3 | 50,73 | -1,65  | -20,16 |
| 55 | 19 | 1 | 65,79 | -29,48 | 15,64  |
| 56 | 19 | 2 | 65,75 | -29,34 | 15,6   |
| 57 | 19 | 3 | 65,74 | -29,27 | 15,44  |
| 58 | 20 | 1 | 54,01 | -36,52 | 14,65  |
| 59 | 20 | 2 | 53,97 | -36,56 | 14,62  |
| 60 | 20 | 3 | 53,99 | -36,55 | 14,52  |
| 61 | 21 | 1 | 57,62 | -33,45 | 8,53   |
| 62 | 21 | 2 | 57,63 | -33,66 | 8,68   |
| 63 | 21 | 3 | 57,62 | -33,5  | 8,53   |
| 64 | 22 | 1 | 46,5  | -29,26 | 9,75   |
| 65 | 22 | 2 | 46,54 | -29,45 | 9,9    |
| 66 | 22 | 3 | 46,54 | -29,43 | 9,9    |
| 67 | 23 | 1 | 36,82 | -18,42 | 9,36   |
| 68 | 23 | 2 | 36,82 | -18,38 | 9,22   |
| 69 | 23 | 3 | 36,85 | -18,44 | 9,5    |
| 70 | 24 | 1 | 91,92 | -0,32  | 0,6    |
| 71 | 24 | 2 | 91,91 | -0,37  | 0,65   |
| 72 | 24 | 3 | 91,91 | -0,41  | 0,64   |
| 73 | 25 | 1 | 44,26 | 7,21   | -7,44  |
| 74 | 25 | 2 | 44,29 | 7,12   | -7,27  |
| 75 | 25 | 3 | 44,23 | 7,26   | -7,44  |
| 76 | 26 | 1 | 25,93 | 0,11   | 7,96   |
| 77 | 26 | 2 | 25,79 | 0,55   | 7,49   |
| 78 | 26 | 3 | 25,86 | 0,27   | 7,59   |
| 79 | 27 | 1 | 26,62 | 0,77   | 7,49   |
| 80 | 27 | 2 | 26,59 | 0,89   | 7,41   |
| 81 | 27 | 3 | 26,57 | 0,88   | 7,18   |
| 82 | 28 | 1 | 28,24 | 3,35   | 0,01   |
| 83 | 28 | 2 | 28,26 | 3,35   | 0,1    |
| 84 | 28 | 3 | 28,22 | 3,33   | 0,27   |
| 85 | 29 | 1 | 29,83 | 12,32  | 11,01  |
| 86 | 29 | 2 | 29,81 | 12,47  | 11,13  |
| 87 | 29 | 3 | 29,97 | 11,81  | 11,34  |
| 88 | 30 | 1 | 47,43 | -1,99  | 35,69  |
| 89 | 30 | 2 | 47,45 | -2,09  | 35,68  |
| 90 | 30 | 3 | 47,43 | -2,19  | 35,99  |
| 91 | 31 | 1 | 61,32 | 6,69   | -4,96  |
| 92 | 31 | 2 | 61,33 | 6,72   | -5,02  |
| 93 | 31 | 3 | 61,32 | 6,83   | -5,12  |
| 94 | 32 | 1 | 95,01 | 1,56   | -0,74  |

|    |    |   |       |      |       |
|----|----|---|-------|------|-------|
| 95 | 32 | 2 | 95,02 | 1,54 | -0,81 |
| 96 | 32 | 3 | 95,01 | 1,69 | -0,8  |



## Anexo 2 – Tomas Cámara Samsung S3 Luz directa a 8 cm “RGB”

| Nº Lectura | Color nº | Rep | R(foto) | G(foto) | B(foto) |
|------------|----------|-----|---------|---------|---------|
| 1          | 1        | 1   | 178     | 133     | 44      |
| 2          | 1        | 2   | 187     | 142     | 59      |
| 3          | 1        | 3   | 170     | 124     | 38      |
| 4          | 2        | 1   | 199     | 119     | 46      |
| 5          | 2        | 2   | 206     | 127     | 50      |
| 6          | 2        | 3   | 198     | 118     | 45      |
| 7          | 3        | 1   | 222     | 126     | 50      |
| 8          | 3        | 2   | 206     | 110     | 34      |
| 9          | 3        | 3   | 216     | 123     | 48      |
| 10         | 4        | 1   | 204     | 62      | 52      |
| 11         | 4        | 2   | 203     | 61      | 51      |
| 12         | 4        | 3   | 205     | 63      | 51      |
| 13         | 5        | 1   | 206     | 43      | 54      |
| 14         | 5        | 2   | 203     | 40      | 53      |
| 15         | 5        | 3   | 200     | 34      | 46      |
| 16         | 6        | 1   | 187     | 40      | 50      |
| 17         | 6        | 2   | 195     | 46      | 58      |
| 18         | 6        | 3   | 190     | 39      | 50      |
| 19         | 7        | 1   | 173     | 40      | 51      |
| 20         | 7        | 2   | 175     | 43      | 59      |
| 21         | 7        | 3   | 171     | 39      | 52      |
| 22         | 8        | 1   | 120     | 23      | 30      |
| 23         | 8        | 2   | 123     | 24      | 31      |
| 24         | 8        | 3   | 127     | 26      | 34      |
| 25         | 9        | 1   | 142     | 27      | 35      |
| 26         | 9        | 2   | 144     | 29      | 37      |
| 27         | 9        | 3   | 138     | 23      | 31      |
| 28         | 10       | 1   | 190     | 32      | 83      |
| 29         | 10       | 2   | 195     | 40      | 88      |
| 30         | 10       | 3   | 200     | 45      | 95      |
| 31         | 11       | 1   | 89      | 38      | 105     |
| 32         | 11       | 2   | 86      | 39      | 107     |
| 33         | 11       | 3   | 92      | 45      | 113     |
| 34         | 12       | 1   | 16      | 42      | 93      |
| 35         | 12       | 2   | 17      | 41      | 89      |
| 36         | 12       | 3   | 19      | 41      | 91      |
| 37         | 13       | 1   | 6       | 47      | 103     |
| 38         | 13       | 2   | 10      | 49      | 104     |
| 39         | 13       | 3   | 2       | 43      | 97      |
| 40         | 14       | 1   | 5       | 96      | 174     |
| 41         | 14       | 2   | 0       | 90      | 168     |
| 42         | 14       | 3   | 0       | 92      | 171     |
| 43         | 15       | 1   | 4       | 59      | 126     |
| 44         | 15       | 2   | 14      | 68      | 142     |
| 45         | 15       | 3   | 3       | 61      | 132     |
| 46         | 16       | 1   | 2       | 81      | 158     |

App para móviles de detección de características hortofrutícolas mediante tratamiento de imágenes

|    |    |   |     |     |     |
|----|----|---|-----|-----|-----|
| 47 | 16 | 2 | 1   | 78  | 154 |
| 48 | 16 | 3 | 2   | 79  | 155 |
| 49 | 17 | 1 | 0   | 96  | 179 |
| 50 | 17 | 2 | 0   | 92  | 169 |
| 51 | 17 | 3 | 0   | 88  | 167 |
| 52 | 18 | 1 | 38  | 89  | 120 |
| 53 | 18 | 2 | 54  | 100 | 133 |
| 54 | 18 | 3 | 32  | 83  | 114 |
| 55 | 19 | 1 | 67  | 135 | 96  |
| 56 | 19 | 2 | 78  | 145 | 102 |
| 57 | 19 | 3 | 65  | 133 | 94  |
| 58 | 20 | 1 | 9   | 111 | 63  |
| 59 | 20 | 2 | 13  | 112 | 66  |
| 60 | 20 | 3 | 7   | 106 | 60  |
| 61 | 21 | 1 | 12  | 117 | 86  |
| 62 | 21 | 2 | 5   | 112 | 80  |
| 63 | 21 | 3 | 19  | 126 | 84  |
| 64 | 22 | 1 | 11  | 95  | 59  |
| 65 | 22 | 2 | 0   | 81  | 48  |
| 66 | 22 | 3 | 0   | 78  | 43  |
| 67 | 23 | 1 | 10  | 54  | 41  |
| 68 | 23 | 2 | 7   | 51  | 38  |
| 69 | 23 | 3 | 9   | 55  | 42  |
| 70 | 24 | 1 | 166 | 165 | 161 |
| 71 | 24 | 2 | 163 | 162 | 157 |
| 72 | 24 | 3 | 166 | 163 | 158 |
| 73 | 25 | 1 | 65  | 58  | 74  |
| 74 | 25 | 2 | 66  | 63  | 80  |
| 75 | 25 | 3 | 65  | 62  | 79  |
| 76 | 26 | 1 | 15  | 20  | 24  |
| 77 | 26 | 2 | 14  | 22  | 25  |
| 78 | 26 | 3 | 16  | 21  | 27  |
| 79 | 27 | 1 | 19  | 22  | 27  |
| 80 | 27 | 2 | 16  | 20  | 23  |
| 81 | 27 | 3 | 20  | 24  | 27  |
| 82 | 28 | 1 | 29  | 29  | 41  |
| 83 | 28 | 2 | 41  | 43  | 55  |
| 84 | 28 | 3 | 33  | 37  | 48  |
| 85 | 29 | 1 | 53  | 23  | 33  |
| 86 | 29 | 2 | 49  | 19  | 29  |
| 87 | 29 | 3 | 48  | 21  | 30  |
| 88 | 30 | 1 | 89  | 60  | 16  |
| 89 | 30 | 2 | 109 | 81  | 34  |
| 90 | 30 | 3 | 103 | 74  | 30  |
| 91 | 31 | 1 | 112 | 101 | 117 |
| 92 | 31 | 2 | 113 | 110 | 121 |
| 93 | 31 | 3 | 105 | 103 | 114 |
| 94 | 32 | 1 | 161 | 162 | 157 |
| 95 | 32 | 2 | 167 | 166 | 164 |
| 96 | 32 | 3 | 169 | 169 | 167 |

### Anexo 3 – Tomas Cámara Samsung S3 Luz Directa a 8cm “Lab”

| Nº Lectura | Color nº | Rep | L(foto)     | a(foto)     | b(foto)     |
|------------|----------|-----|-------------|-------------|-------------|
| 1          | 1        | 1   | 62,61836756 | 6,806061321 | 50,10125619 |
| 2          | 1        | 2   | 62,99037596 | 6,755498825 | 50,07504337 |
| 3          | 1        | 3   | 65,58469551 | 6,652438295 | 49,04839265 |
| 4          | 1        | 4   | 65,58469551 | 6,416776086 | 49,87387161 |
| 5          | 1        | 5   | 61,70676791 | 7,789942165 | 50,90226253 |
| 6          | 2        | 1   | 60,50043367 | 22,36099131 | 49,81035358 |
| 7          | 2        | 2   | 63,22346988 | 23,79500151 | 48,8411456  |
| 8          | 2        | 3   | 58,89687975 | 23,48088702 | 51,2221694  |
| 9          | 2        | 4   | 58,5287171  | 23,55598534 | 51,25464103 |
| 10         | 2        | 5   | 58,87677899 | 22,08170533 | 51,11837429 |
| 11         | 3        | 1   | 60,17869613 | 27,61735493 | 53,79011798 |
| 12         | 3        | 2   | 61,64690993 | 27,39086205 | 52,79318718 |
| 13         | 3        | 3   | 59,46556187 | 27,85246902 | 53,00778376 |
| 14         | 3        | 4   | 60,67592069 | 26,79326216 | 51,6442986  |
| 15         | 3        | 5   | 60,15780177 | 26,32284999 | 52,81698584 |
| 16         | 4        | 1   | 59,16521119 | 46,48077168 | 30,26444503 |
| 17         | 4        | 2   | 55,42885494 | 46,4339981  | 32,55511301 |
| 18         | 4        | 3   | 53,69847015 | 45,91699351 | 32,47599303 |
| 19         | 4        | 4   | 56,13671629 | 46,3297401  | 32,37144874 |
| 20         | 4        | 5   | 53,34348053 | 45,96897405 | 32,57176805 |
| 21         | 5        | 1   | 43,05346305 | 60,2189642  | 35,40700455 |
| 22         | 5        | 2   | 42,77208875 | 60,23842767 | 34,45363249 |
| 23         | 5        | 3   | 42,57276678 | 59,95284125 | 34,17559436 |
| 24         | 5        | 4   | 42,87950429 | 60,0089812  | 33,98566653 |
| 25         | 5        | 5   | 43,28152381 | 58,87533626 | 34,43826035 |
| 26         | 6        | 1   | 39,79621467 | 55,93126187 | 29,65986636 |
| 27         | 6        | 2   | 38,45155345 | 54,31586706 | 29,60940274 |
| 28         | 6        | 3   | 39,73999034 | 54,46571458 | 28,88758848 |
| 29         | 6        | 4   | 40,36093217 | 54,43080697 | 29,71761939 |
| 30         | 6        | 5   | 40,11931249 | 55,02284215 | 30,03815025 |
| 31         | 7        | 1   | 37,31210292 | 51,28370697 | 23,53612902 |
| 32         | 7        | 2   | 38,26588441 | 51,18692846 | 24,84857142 |
| 33         | 7        | 3   | 36,6541278  | 48,91560672 | 27,51777566 |
| 34         | 7        | 4   | 36,40722403 | 49,21268781 | 24,07635954 |
| 35         | 7        | 5   | 38,56607618 | 48,23019467 | 22,61515675 |
| 36         | 8        | 1   | 31,17454232 | 40,22046481 | 16,99611455 |
| 37         | 8        | 2   | 31,84830386 | 38,51670178 | 17,22646137 |
| 38         | 8        | 3   | 30,7552161  | 39,38263562 | 17,6764565  |
| 39         | 8        | 4   | 29,6678374  | 39,49173706 | 17,46203173 |
| 40         | 8        | 5   | 27,02184914 | 38,07312466 | 17,66149569 |
| 41         | 9        | 1   | 27,02184914 | 38,07312466 | 17,66149569 |
| 42         | 9        | 2   | 33,08307701 | 43,01450324 | 18,51344041 |
| 43         | 9        | 3   | 31,14478626 | 39,43634465 | 16,92098999 |
| 44         | 9        | 4   | 28,87475999 | 41,32361655 | 18,41094444 |
| 45         | 9        | 5   | 29,5927179  | 41,33822781 | 18,09766061 |
| 46         | 10       | 1   | 43,90197714 | 58,82675748 | 8,095718978 |
| 47         | 10       | 2   | 43,09789317 | 57,88943865 | 10,00246289 |

|    |    |   |             |              |              |
|----|----|---|-------------|--------------|--------------|
| 48 | 10 | 3 | 43,79397481 | 57,67114158  | 7,268070385  |
| 49 | 10 | 4 | 43,00286391 | 56,33681842  | 7,289928803  |
| 50 | 10 | 5 | 44,12697134 | 57,70388179  | 7,144306646  |
| 51 | 11 | 1 | 32,19522176 | 26,72248639  | -24,78349923 |
| 52 | 11 | 2 | 31,66927405 | 29,07538259  | -26,23042753 |
| 53 | 11 | 3 | 32,11283992 | 30,24963186  | -26,13591216 |
| 54 | 11 | 4 | 30,87821985 | 30,37848384  | -26,21166002 |
| 55 | 11 | 5 | 31,41171327 | 29,29173934  | -24,73154554 |
| 56 | 12 | 1 | 24,91512124 | 4,843137345  | -27,08595607 |
| 57 | 12 | 2 | 23,76755384 | 6,331788113  | -29,57429011 |
| 58 | 12 | 3 | 26,25920673 | 7,563549217  | -30,72500401 |
| 59 | 12 | 4 | 26,48002976 | 6,669862793  | -31,02073125 |
| 60 | 12 | 5 | 25,50269732 | 9,964631765  | -30,63286798 |
| 61 | 13 | 1 | 22,67065035 | 2,701033212  | -26,1240353  |
| 62 | 13 | 2 | 22,42654197 | 4,062522129  | -27,82413795 |
| 63 | 13 | 3 | 25,31733402 | 2,463442621  | -26,46377932 |
| 64 | 13 | 4 | 25,08539405 | 3,5370363    | -26,82544099 |
| 65 | 13 | 5 | 24,38029414 | 4,525259466  | -27,2968362  |
| 66 | 14 | 1 | 35,25695252 | 5,295986073  | -40,62738043 |
| 67 | 14 | 2 | 36,14246845 | 3,008835042  | -38,6058907  |
| 68 | 14 | 3 | 35,42135181 | 3,738400651  | -39,15331411 |
| 69 | 14 | 4 | 36,94117307 | 2,919981472  | -39,144361   |
| 70 | 14 | 5 | 36,5745282  | 3,325173818  | -39,73279365 |
| 71 | 15 | 1 | 30,52705937 | 6,324031189  | -36,53175077 |
| 72 | 15 | 2 | 29,72498305 | 6,978320548  | -37,82638629 |
| 73 | 15 | 3 | 28,48499474 | 7,387064766  | -37,95089081 |
| 74 | 15 | 4 | 27,92103302 | 6,629145194  | -36,97936585 |
| 75 | 15 | 5 | 28,44219691 | 7,062398953  | -37,39265083 |
| 76 | 16 | 1 | 34,29088757 | 6,642713427  | -40,34435302 |
| 77 | 16 | 2 | 38,73219294 | 3,897785812  | -41,08401863 |
| 78 | 16 | 3 | 36,407943   | 4,869796275  | -41,20406073 |
| 79 | 16 | 4 | 39,70757328 | 4,879490391  | -42,50292545 |
| 80 | 16 | 5 | 37,30861121 | 5,200839552  | -41,56244237 |
| 81 | 17 | 1 | 39,90347502 | 4,536510182  | -45,17954992 |
| 82 | 17 | 2 | 41,32323656 | 3,281404955  | -44,67862815 |
| 83 | 17 | 3 | 42,17018841 | 3,477996038  | -45,08664523 |
| 84 | 17 | 4 | 43,16645611 | 2,107345249  | -44,08073991 |
| 85 | 17 | 5 | 42,08052746 | 2,947355045  | -44,6438981  |
| 86 | 18 | 1 | 42,30179994 | -7,416387553 | -20,39251238 |
| 87 | 18 | 2 | 40,42902585 | -6,43945712  | -20,8780243  |
| 88 | 18 | 3 | 41,18567893 | -6,404725953 | -22,73860252 |
| 89 | 18 | 4 | 40,69666189 | -7,035967468 | -21,6864633  |
| 90 | 18 | 5 | 39,58053545 | -6,205913103 | -22,81996182 |
| 91 | 19 | 1 | 55,16997955 | -29,65950998 | 16,78277396  |
| 92 | 19 | 2 | 52,98854603 | -31,5361768  | 16,53942903  |
| 93 | 19 | 3 | 54,17550027 | -30,94795158 | 14,21823021  |
| 94 | 19 | 4 | 52,32719896 | -30,67209477 | 15,05676573  |
| 95 | 19 | 5 | 55,23637159 | -31,82829208 | 17,39883473  |
| 96 | 20 | 1 | 44,52432844 | -33,98871362 | 14,53094802  |
| 97 | 20 | 2 | 43,86361624 | -33,12180376 | 14,80528988  |

|     |    |   |             |              |              |
|-----|----|---|-------------|--------------|--------------|
| 98  | 20 | 3 | 44,79003783 | -32,00262729 | 14,35071999  |
| 99  | 20 | 4 | 43,88459523 | -32,89949642 | 14,24915524  |
| 100 | 20 | 5 | 44,64093583 | -33,14064593 | 14,70933296  |
| 101 | 21 | 1 | 49,22801544 | -36,08434372 | 12,32004971  |
| 102 | 21 | 2 | 48,88583905 | -36,01162183 | 14,16402638  |
| 103 | 21 | 3 | 47,66523572 | -33,38535993 | 11,90787457  |
| 104 | 21 | 4 | 48,6421577  | -34,64005533 | 11,5152016   |
| 105 | 21 | 5 | 47,5803551  | -37,16525275 | 14,05778235  |
| 106 | 22 | 1 | 37,19189623 | -29,05926728 | 10,33613574  |
| 107 | 22 | 2 | 37,25579233 | -28,6756129  | 11,0446821   |
| 108 | 22 | 3 | 38,34211669 | -29,59726615 | 11,33818733  |
| 109 | 22 | 4 | 37,37341823 | -27,76221604 | 10,61586153  |
| 110 | 22 | 5 | 39,32415908 | -27,99650917 | 9,107402881  |
| 111 | 23 | 1 | 26,36370435 | -13,82019443 | 3,949772432  |
| 112 | 23 | 2 | 29,14311602 | -15,4541533  | 5,353300343  |
| 113 | 23 | 3 | 31,32640573 | -12,13813286 | 2,16133869   |
| 114 | 23 | 4 | 29,9809288  | -12,17906227 | -0,480463543 |
| 115 | 23 | 5 | 30,89617742 | -11,99864323 | 0,883759761  |
| 116 | 24 | 1 | 70,14297755 | -1,284049    | 2,434806905  |
| 117 | 24 | 2 | 67,90620434 | -1,292341479 | 2,45120794   |
| 118 | 24 | 3 | 68,09172226 | -0,38574482  | 2,184973639  |
| 119 | 24 | 4 | 68,07071095 | -2,38180726  | 3,23492483   |
| 120 | 24 | 5 | 67,32222412 | -2,387043248 | 3,242320463  |
| 121 | 25 | 1 | 46,9047795  | 2,597491696  | -3,473241579 |
| 122 | 25 | 2 | 45,75904899 | 3,037416994  | -4,597873917 |
| 123 | 25 | 3 | 42,84944582 | 0,680591425  | -2,956645239 |
| 124 | 25 | 4 | 45,46916486 | 3,375068217  | -3,21373029  |
| 125 | 25 | 5 | 45,55374842 | 2,071589733  | -4,320475881 |
| 126 | 26 | 1 | 23,23577019 | -1,545194107 | 1,571703445  |
| 127 | 26 | 2 | 27,53469245 | 0,001974261  | -0,003906182 |
| 128 | 26 | 3 | 25,54388397 | -1,073238703 | 0,316345726  |
| 129 | 26 | 4 | 24,72141487 | 1,095079165  | 2,479942342  |
| 130 | 26 | 5 | 25,36236416 | -1,914667263 | 0,029600695  |
| 131 | 27 | 1 | 26,12071405 | 1,502425543  | 1,226157836  |
| 132 | 27 | 2 | 27,82793215 | 1,069771969  | 2,4322714    |
| 133 | 27 | 3 | 26,20744082 | 0,001914071  | -0,003787093 |
| 134 | 27 | 4 | 26,60416558 | -2,805386111 | 3,848385701  |
| 135 | 27 | 5 | 25,76271077 | 0,001893903  | -0,00374719  |
| 136 | 28 | 1 | 26,05997602 | 1,81838689   | -3,523502503 |
| 137 | 28 | 2 | 25,83711113 | 0,758019818  | -3,213462963 |
| 138 | 28 | 3 | 25,89437137 | 3,460040635  | -6,414534412 |
| 139 | 28 | 4 | 26,17103111 | 0,445847218  | -4,708637574 |
| 140 | 28 | 5 | 26,61439558 | 0,442926645  | -4,696886121 |
| 141 | 29 | 1 | 25,13913853 | 17,81185159  | 2,797180619  |
| 142 | 29 | 2 | 26,15130264 | 16,82141023  | 2,267531612  |
| 143 | 29 | 3 | 27,7545849  | 17,71165991  | 2,019037903  |
| 144 | 29 | 4 | 25,8714201  | 15,95744444  | 1,827583458  |
| 145 | 29 | 5 | 26,58368425 | 16,78015685  | 2,243946321  |
| 146 | 30 | 1 | 42,30993477 | 3,178736353  | 21,80631644  |
| 147 | 30 | 2 | 42,45663879 | 3,636873019  | 24,36037212  |



App para móviles de detección de características hortofrutícolas mediante tratamiento de imágenes

|     |    |   |             |             |              |
|-----|----|---|-------------|-------------|--------------|
| 148 | 30 | 3 | 42,2087916  | 4,450404453 | 22,88591924  |
| 149 | 30 | 4 | 44,11149064 | 1,814181245 | 25,34889535  |
| 150 | 30 | 5 | 44,63221514 | 2,59796842  | 21,42624116  |
| 151 | 31 | 1 | 50,11906266 | 4,729107666 | -5,745855168 |
| 152 | 31 | 2 | 47,85205145 | 3,136280668 | -2,633397697 |
| 153 | 31 | 3 | 50,32756848 | 5,6771872   | -6,007223731 |
| 154 | 31 | 4 | 46,56516138 | 3,027090485 | -4,58338996  |
| 155 | 31 | 5 | 46,2263966  | 3,464457181 | -5,694490146 |

#### **Anexo 4 – Fórmulas de Conversión Cámara “Lab” a Colorímetro “Lab”**

En estadística, el análisis de la regresión es un proceso estadístico para estimar las relaciones entre variables. Un procedimiento de Regresión Polinomial está diseñado para construir un modelo estadístico que describa el impacto de un solo factor cuantitativo  $X$  y potencias de  $X$ .

En nuestro caso, se realiza la conversión de 3 factores cuantitativos, por lo que usamos una **Regresión Polinomial Múltiple**. Los 3 factores cuantitativos y sus potencias forman cada una de las 3 fórmulas de conversión.

Disponemos de 3 fórmulas de conversión por las 3 variables que deseamos estimar. Se requiere la conversión de modo de color LAB de la cámara del Smartphone a modo de color LAB del colorímetro. Dichos modos de color disponen de 3 valores ( $L$ ,  $a$ ,  $b$ ). Por tanto, en cada fórmula respectivamente, los 3 valores LAB de la cámara del Smartphone estiman el valor de cada variable LAB de colorímetro.

(L,a,b) de cámara Smartphone -> “L” de colorímetro

c = L (Cámara)

e = a (Cámara)

f = b (Cámara)

y = L (Colorímetro)

$$\begin{aligned} \text{Result: } y = & -5.6433542 \cdot 10^{-5} \cdot c^4 - 2.729373046 \cdot 10^{-6} \cdot c^3 \cdot e - 2.542036479 \cdot 10^{-5} \cdot c^3 \cdot f - \\ & 8.692019067 \cdot 10^{-6} \cdot c^2 \cdot e^2 - 7.260781609 \cdot 10^{-5} \cdot c^2 \cdot e \cdot f + 2.125378615 \cdot 10^{-5} \cdot c^2 \cdot f^2 - \\ & 3.140729563 \cdot 10^{-6} \cdot c \cdot e^3 + 1.15675544 \cdot 10^{-5} \cdot c \cdot e^2 \cdot f + 3.764521033 \cdot 10^{-5} \cdot c \cdot e \cdot f^2 - \\ & 1.296331606 \cdot 10^{-5} \cdot c \cdot f^3 + 2.189807162 \cdot 10^{-7} \cdot e^4 + 5.424451046 \cdot 10^{-7} \cdot e^3 \cdot f + \\ & 9.213962171 \cdot 10^{-6} \cdot e^2 \cdot f^2 - 7.53054916 \cdot 10^{-6} \cdot e \cdot f^3 + 2.151807588 \cdot 10^{-6} \cdot f^4 + \\ & 8.964967317 \cdot 10^{-3} \cdot c^3 + 1.601962232 \cdot 10^{-3} \cdot c^2 \cdot e + 4.036057663 \cdot 10^{-3} \cdot c^2 \cdot f - \\ & 8.747822934 \cdot 10^{-5} \cdot c \cdot e^2 + 5.353660233 \cdot 10^{-3} \cdot c \cdot e \cdot f - 2.678962354 \cdot 10^{-3} \cdot c \cdot f^2 + \\ & 1.293994576 \cdot 10^{-4} \cdot e^3 - 8.673035123 \cdot 10^{-4} \cdot e^2 \cdot f - 1.864793144 \cdot 10^{-3} \cdot e \cdot f^2 + \\ & 7.448971447 \cdot 10^{-4} \cdot f^3 - 4.763337062 \cdot 10^{-1} \cdot c^2 - 1.029182322 \cdot 10^{-1} \cdot c \cdot e - \\ & 1.960291418 \cdot 10^{-1} \cdot c \cdot f + 2.642941511 \cdot 10^{-2} \cdot e^2 - 8.169014945 \cdot 10^{-2} \cdot e \cdot f + \\ & 7.881707427 \cdot 10^{-2} \cdot f^2 + 11.09559613 \cdot c + 1.438869318 \cdot e + 2.882452371 \cdot f - \\ & 71.04844428 \end{aligned}$$

Residual Sum of Squares: rss = 624.6807062

Coefficient of Determination: R2 = 0.9857038057

(L,a,b) de cámara Smartphone -> “a” de colorímetro

c = L (Cámara)

e = a (Cámara)

f = b (Cámara)

y = a (Colorímetro)

$$\begin{aligned} \text{Result: } y = & 4.002696867 \cdot 10^{-5} \cdot c^4 - 7.791370788 \cdot 10^{-5} \cdot c^3 \cdot e - 1.610935814 \cdot 10^{-5} \cdot c^3 \cdot f + \\ & 4.972142022 \cdot 10^{-6} \cdot c^2 \cdot e^2 + 1.325464287 \cdot 10^{-4} \cdot c^2 \cdot e \cdot f + 1.692922586 \cdot 10^{-5} \cdot c^2 \cdot f^2 + \\ & 5.429026582 \cdot 10^{-8} \cdot c \cdot e^3 - 1.54822404 \cdot 10^{-5} \cdot c \cdot e^2 \cdot f - 3.439066986 \cdot 10^{-5} \cdot c \cdot e \cdot f^2 - \\ & 1.965814637 \cdot 10^{-5} \cdot c \cdot f^3 + 1.001576885 \cdot 10^{-6} \cdot e^4 + 5.961740568 \cdot 10^{-6} \cdot e^3 \cdot f + \\ & 4.536510191 \cdot 10^{-6} \cdot e^2 \cdot f^2 + 4.301046432 \cdot 10^{-6} \cdot e \cdot f^3 + 4.721310113 \cdot 10^{-6} \cdot f^4 - \\ & 7.622999388 \cdot 10^{-3} \cdot c^3 + 6.700679003 \cdot 10^{-3} \cdot c^2 \cdot e + 2.966467148 \cdot 10^{-3} \cdot c^2 \cdot f - \\ & 1.644213219 \cdot 10^{-4} \cdot c \cdot e^2 - 9.47190348 \cdot 10^{-3} \cdot c \cdot e \cdot f - 1.943058629 \cdot 10^{-3} \cdot c \cdot f^2 - \\ & 3.201487341 \cdot 10^{-4} \cdot e^3 + 5.586738105 \cdot 10^{-4} \cdot e^2 \cdot f + 8.176201035 \cdot 10^{-4} \cdot e \cdot f^2 + \\ & 9.973249911 \cdot 10^{-4} \cdot f^3 + 5.176919871 \cdot 10^{-1} \cdot c^2 - 1.468717118 \cdot 10^{-1} \cdot c \cdot e - \\ & 1.465238811 \cdot 10^{-1} \cdot c \cdot f + 6.478583279 \cdot 10^{-4} \cdot e^2 + 1.577483565 \cdot 10^{-1} \cdot e \cdot f + \\ & 4.815754326 \cdot 10^{-2} \cdot f^2 - 14.82144951 \cdot c + 1.599301155 \cdot e + 1.752270734 \cdot f + \\ & 150.8798357 \end{aligned}$$

Residual Sum of Squares: rss = 324.629771

Coefficient of Determination: R2 = 0.9956925867

(L,a,b) de cámara Smartphone -> “b” de colorímetro

c = L (Cámara)

e = a (Cámara)

f = b (Cámara)

y = b (Colorímetro)

$$\begin{aligned} \text{Result: } y = & -3.987282478 \cdot 10^{-5} \cdot c^4 + 3.50449784 \cdot 10^{-5} \cdot c^3 \cdot e + 1.16105166 \cdot 10^{-5} \cdot c^3 \cdot f + \\ & 2.856734291 \cdot 10^{-5} \cdot c^2 \cdot e^2 - 7.845264056 \cdot 10^{-5} \cdot c^2 \cdot e \cdot f - 3.830178777 \cdot 10^{-5} \cdot c^2 \cdot f^2 + \\ & 1.006685516 \cdot 10^{-5} \cdot c \cdot e^3 + 3.84620005 \cdot 10^{-6} \cdot c \cdot e^2 \cdot f + 2.187218112 \cdot 10^{-5} \cdot c \cdot e \cdot f^2 + \\ & 5.747316452 \cdot 10^{-5} \cdot c \cdot f^3 + 1.397349313 \cdot 10^{-6} \cdot e^4 - 5.529649842 \cdot 10^{-6} \cdot e^3 \cdot f - \\ & 4.105789396 \cdot 10^{-6} \cdot e^2 \cdot f^2 - 1.460175974 \cdot 10^{-6} \cdot e \cdot f^3 - 1.999337441 \cdot 10^{-5} \cdot f^4 + \\ & 7.923684658 \cdot 10^{-3} \cdot c^3 - 2.077820723 \cdot 10^{-3} \cdot c^2 \cdot e - 4.784477637 \cdot 10^{-3} \cdot c^2 \cdot f - \\ & 2.206411379 \cdot 10^{-3} \cdot c \cdot e^2 + 4.350142517 \cdot 10^{-3} \cdot c \cdot e \cdot f + 5.245497757 \cdot 10^{-3} \cdot c \cdot f^2 - \\ & 1.6893680 \cdot 10^{-4} \cdot e^3 - 2.939370635 \cdot 10^{-4} \cdot e^2 \cdot f + 1.410184069 \cdot 10^{-4} \cdot e \cdot f^2 - \\ & 2.854511738 \cdot 10^{-3} \cdot f^3 - 5.535855334 \cdot 10^{-1} \cdot c^2 - 1.909512648 \cdot 10^{-2} \cdot c \cdot e + \\ & 2.949787353 \cdot 10^{-1} \cdot c \cdot f + 3.189695502 \cdot 10^{-2} \cdot e^2 - 4.277717828 \cdot 10^{-2} \cdot e \cdot f - \\ & 1.298304627 \cdot 10^{-1} \cdot f^2 + 15.90785368 \cdot c + 1.433060643 \cdot e - 3.375125386 \cdot f - \\ & 153.8029171 \end{aligned}$$

Residual Sum of Squares: rss = 325.7945868

Coefficient of Determination: R2 = 0.997696286