UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de aplicaciones de visión para dispositivos móviles. Medición de distancias con un dispositivo Android.

Proyecto Final de Grado

Grado en Ingeniería Informática

Autor: David Fernández Delegido

Director: Vicente Luis Atienza Vanacloig

Septiembre 2014

Resumen

Se ha desarrollado una aplicación para dispositivos Android, que mediante la captura de una imagen tomada con la cámara incorporada, permite realizar mediciones de distancia entre elementos de la escena. Se han considerado diversas aproximaciones, como la inclusión de un elemento de tamaño conocido en la escena que la aplicación reconocerá automáticamente o la inserción de medidas de referencia por el usuario que se usarán para medir los demás objetos. También se han implementado otras funcionalidades que complementan la aplicación, como son medir la distancia a un objeto así como su altura.

Palabras clave: Android, medición, distancias, opencv, cámara.

Abstract

We have developed an application for Android devices, which allows measurements of distance between elements of the scene by capturing an image taken with the built-in camera. Various approaches have been considered, such as the inclusion of an element of known size in the scene that the application will automatically recognize or inserting reference measurements by the user that will be used to measure other objects. It has also been implemented some other functionalities to complement the application, such as the measurement of the distance to an object as well as its height.

Keywords : Android, measuring distances, opencv, camera.

Tabla de contenidos

Índice de ilustraciones

1. Introducción

Android es un sistema operativo que opera principalmente en dispositivos táctiles. La mayoría de estos dispositivos tienen incorporados diversos elementos como cámaras, micrófonos o giroscopios. Usando uno de los elementos, la cámara, se ha desarrollado una aplicación a un problema que se considera interesante como estudio: la medición de distancias mediante una imagen tomada con nuestro teléfono móvil.

1.1 Presentación del problema

En un principio la idea de medición de distancias en una fotografía de una escena tomada con nuestro dispositivo móvil puede parecer simple y fácil de implementar; sin embargo, existen diversos aspectos que hacen que el proceso de desarrollo de un producto final completamente funcional revista de significativa complejidad y resulte de gran interés.

Uno de los aspectos importantes que tenemos que tener en cuenta al desarrollar nuestra aplicación es la herramienta que usaremos para desarrollar en la plataforma Android. Para ello se usa el lenguaje de programación Java, muy extendido hoy en día y conocidos por todos. Actualmente existen diversos entornos de desarrollo con los que podremos trabajar en Android; sin embargo, en la propia página web de Android [1] se nos ofrece la descarga del entorno de desarrollo Eclipse con todos los elementos que necesitamos para desarrollar en esta plataforma integrados en él, pudiendo así olvidarnos de configuraciones tediosas.

Hay que tener en cuenta que para la realización del proyecto hay que llevar a cabo un estudio previo del funcionamiento de Android, tanto de forma teórica como de forma práctica. Se debe documentar cuáles son los métodos correctos para programar en Android y estudiar las particularidades de su API. Todo este proceso de toma de contacto con la metodología de desarrollo de aplicaciones para una nueva plataforma conlleva un tiempo y un esfuerzo que se ven recompensados con los resultados de aprendizaje y la experiencia obtenidos, y que no son menos importantes que el resultado final del proyecto: la elaboración de una útil aplicación móvil.

Respecto al objetivo de la aplicación, esta se basa principalmente en la captura de imágenes y su posterior procesamiento para permitir que el usuario pueda medir mediante distintos procedimientos objetos de su interés en la escena. La aproximación presentada en este proyecto se ha centrado principalmente en objetos de pequeño tamaño ya que la información y los objetos de referencia que en la práctica se usan, corresponden a esta escala.

La aproximación más fácil y la que se implementó en un principio, ha sido la de medir diferentes objetos teniendo como referencia una medida conocida. De esta manera el usuario puede realizar una fotografía con su cámara y posteriormente la aplicación le permite dibujar distancias conocidas encima de la fotografía para poder obtener las medidas deseadas.

La siguiente aproximación está directamente relacionada con la anterior. Se pretende abordar el mismo problema, pero la medida por referencia necesaria para calcular las distancias que el usuario necesite saber se obtendrá de forma automática, utilizando como referencia algún objeto conocido como por ejemplo en nuestro caso, una tarjeta de crédito.

Finalmente también se ha considerado interesante el estudio de la medición de distancias y alturas de un objeto relativamente lejano. Para este problema se ha hecho uso de la cámara y del giroscopio incorporado en la mayoría de los dispositivos móviles actuales. El usuario puede enfocar mediante una previsualización de la cámara a la base del objeto para obtener su distancia mediante técnicas trigonométricas que hacen uso de la información ofrecida por el giroscopio y posteriormente a su parte superior si quiere medir su altura.

1.2 Objetivos del proyecto

El objetivo principal del proyecto es el desarrollo de una aplicación que permite a los usuarios medir objetos en una escena fotografiada. Aparte de esta descripción general del proyecto, se definieron, en una primera fase del proyecto, una serie de objetivos primordiales que parecía necesario cumplir para que el proyecto fuera lo suficientemente funcional e interesante. Estos primeros objetivos esenciales son:

- Ofrecer la posibilidad de marcar manualmente las medidas de referencia en la escena para después poder medir otros objetos contenidos en ella. Para que esta funcionalidad resulte factible, todos los elementos de la escena (marcados y medidos) deben estar físicamente situados en un plano ortogonal al eje óptico de la cámara.
- Permitir la búsqueda automática de un objeto conocido en la escena, del cual se dispone de una vista previa almacenada en el dispositivo y medir otros elementos a partir de las medidas detectadas.
- Posibilitar la medida de alturas y distancias de objetos lejanos situados a la altura del suelo como pueden ser una persona, un edificio, un árbol, etc.
- Medir distancias en 3D mediante estereoscopia, usando librerías de tratamiento de imágenes y visión por computador.

Los tres primeros objetivos esenciales están implementados en la aplicación, por el contrario, el tercer objetivo se descartó conforme el desarrollo avanzaba ya que no se pudo encontrar una solución práctica viable al problema de obtener dos tomas de una misma escena con un dispositivo habitual de una sola cámara delantera, con una geometría conocida en la toma. El problema se consideró demasiado avanzado e incierto en su resolución por lo que decidió abandonarse y profundizar en la consecución del resto de objetivos.

Tras el análisis del proyecto se desarrollaron los objetivos concretos que la aplicación debía tener. Estos objetivos específicos hacen referencia a la mayor parte de los aspectos del proyecto y son los siguientes:

- Para acceder a todas las características de la aplicación se debe crear un menú de inicio el cual contenga diversos botones clasificados según su objetivo y que activen su correspondiente tarea.

- La carga de imágenes para su procesamiento se realiza mediante dos métodos: directamente después de realizar una fotografía en la aplicación o bien cargando la imagen seleccionándola en la galería del dispositivo móvil.
- Mediante la medición manual de objetos, el usuario puede dibujar medidas (en forma de líneas rectas) en la escena tomada con la cámara (o cargada desde la galería) e introducir medidas que se conozcan. A partir de estas medidas el usuario puede dibujar otras líneas para obtener sus medidas en el momento.
- Para la detección automática del objeto de referencia, el usuario simplemente debe realizar la fotografía o cargarla desde la galería. Una vez realizado este paso, el programa ejecuta su código para detectar el objeto y mostrarlo en la imagen. A partir de aquí se debe poder realizar las medidas pertinentes.
- Es necesario dar la oportunidad al usuario de poder elegir su propia imagen de referencia para la detección automática (una tarjeta de crédito o similar en este proyecto). Es necesario pues, crear un sistema que permita fácilmente tomar su foto con unas características adecuadas para la aplicación.
- Se implementan las opciones de calcular distancia y altura de un objeto. Para esto, se necesita que la aplicación proporcione una interfaz personalizada al usuario donde se muestre la distancia o altura en tiempo real al mismo tiempo que se muestra una previsualización de la cámara. Con esto, estudiaremos las funcionalidades que ofrece Android para manejar la cámara a un bajo nivel.
- Es necesario que la aplicación sea rápida y actúe de forma precisa en respuesta a las acciones del usuario. Este objetivo es deseable para cualquier aplicación elaborada en un dispositivo táctil ya que dada la gran variedad de tipos existentes se tiene que intentar que los requisitos de la aplicación sean los menores posibles.

Todos estos objetivos específicos conforman las bases de la aplicación final. Se ha intentado cumplir al pie de la letra cada uno de ellos ya que se consideran necesarios para la elaboración de un proyecto de calidad.

1.3 Resumen de la solución del problema

Las soluciones propuestas para cada uno de los objetivos anteriores son diversas ya que cada objetivo no se puede resolver con una única solución común.

Para todo el diseño de la aplicación se ha utilizado la propia API de Android, ya que proporciona una base completa y robusta. Nos olvidamos de librerías o añadidos de terceros que pueden complicar los diseños de los menús.

Para la detección automática del objeto de referencia usamos las librerías de OpenCV, estas librerías son usadas para tratar imágenes o para tratar visión artificial en tiempo real [3]. Para todas las demás soluciones se utilizan los propios recursos de Android.

Para la toma de fotos se hace uso de las dos formas que ofrece Android, una es usando la propia aplicación de la cámara que tienen todos los dispositivos con esta plataforma y la otra posibilidad es manejando la API de la cámara. Con esta segunda opción tenemos un mayor control de las características de la cámara ya que trabajamos a un nivel más bajo. Es necesario utilizar la API de la cámara de Android si queremos realizar previsualizaciones de la cámara al mismo tiempo que queremos mostrar algún

menú o información adicional. En las soluciones de la obtención de altura y distancia, debido a que es necesario mostrar la distancia en metros y la previsualizacion de la cámara en el mismo momento que el usuario enfoca al objeto, tenemos que hacer uso de la API de Android.

También hacemos uso de la API de la cámara de Android para mostrar la previsualizacion de la cámara al mismo tiempo que dibujamos encima la plantilla del objeto de referencia que queremos fotografiar para guardar en la tarea de guardar la imagen de referencia. Una vez el usuario obtiene la imagen deseada se recorta y guarda en el momento.

Para todas las soluciones de la aplicación se ha hecho un uso muy variado de todas las posibilidades de Android para diseñar y crear aplicaciones. El proyecto es una gran prueba de lo que se puede conseguir con las herramientas de esta plataforma.

1.4 Descripción de la estructura de la memoria

En este apartado de la memoria se pretende explicar el contenido de cada una de los 6 capítulos principales que la componen. Todos y cada uno de los anteriores apartados componen la introducción del proyecto que ayudan a comprender el tema del trabajo.

En el capítulo 2 nos centraremos en los antecedentes del tema del proyecto y describiremos los sistemas más conocidos en el área de la aplicación. Este apartado es muy importante ya que antes de empezar a escribir código, tenemos que tener bien definido qué es lo que queremos hacer y cómo queremos hacerlo. El trabajo de investigación llevado a cabo para el desarrollo del proyecto se ilustrará en este capítulo.

En el capítulo 3 (Análisis) se describe detalladamente la solución, paso a paso. Se intenta evitar hacer referencias a la concreta implementación o a tecnologías específicas. Se detalla el flujo de trabajo interno de la aplicación y se describe cómo fluye la información por toda la aplicación. También se explican muy por encima los algoritmos que usan las librerías de OpenCV para la detección automática de los objetos de referencia. Finalmente se detallan las decisiones tomadas para el diseño de la aplicación.

En el capítulo 4 (Diseño) se exponen las clases principales del programa utilizando un lenguaje de descripción. Se describe la metodología de desarrollo así como el funcionamiento de la aplicación. Para ayudar a entender la aplicación y, sobre todo, ayudar a diseñarla se han incluido casos de uso que simulan distintas interacciones entre el usuario y los distintos módulos del programa. También se hace una puntualización en el diseño de la interfaz de usuario ya que en el desarrollo para dispositivos móviles, una interfaz de usuario mal diseñada puede suponer un fracaso de aplicación.

En el capítulo 5 (Resultados) se desarrollan varios ejemplos completos del uso de la aplicación junto a las interacciones del usuario con el programa y sus respectivos resultados. Se evalúa el sistema y se comprueba si hemos alcanzado los objetivos descritos en este primer capítulo.

El último capítulo importante son las conclusiones. Se describen las características del desarrollo del proyecto. Se complementa con las conclusiones al trabajo realizado y los futuros trabajos posibles en esta área de trabajo.

Finalmente se adjuntan tres apéndices considerados importantes para complementar el trabajo. El primero de ellos es en el que se describe la utilización de la aplicación para que el usuario pueda consultarla en caso de duda. En el segundo apéndice se describe el proceso seguido para la instalación y configuración del entorno de desarrollo utilizado en la elaboración de este trabajo. El tercero es una lista de clases con sus métodos más importantes.

2. Antecedentes

Para la realización del proyecto se han utilizado diferentes tecnologías. Cada una de ellas ha sido estudiada para su correcta utilización. Se ha intentado escoger aquellas que sean de un uso simple y comúnmente utilizadas. A continuación se expondrán dos apartados. En el primero de ellos se describirán los sistemas o tecnologías que existen actualmente para el desarrollo de aplicaciones y en el segundo se habla sobre el estado actual de las tecnologías usadas y sobre el uso que se les da en el desarrollo móvil.

2.1 Tecnologías utilizadas

A continuación se describen tres de las tecnologías más importantes utilizadas para el desarrollo del proyecto.

2.1.1 Eclipse ADT (Android Developer Tools)

Eclipse es un IDE (integrated development environment) que contiene una base de trabajo para poder desarrollar aplicaciones en diferentes lenguajes. Tiene un sistema de complementos que se utiliza para personalizar su entorno. Debido a esta gran flexibilidad Android tiene una versión propia de Eclipse, ADT [2]. Es un añadido diseñado para darnos un entorno poderoso e integrado en el cual podremos desarrollar aplicaciones Android. ADT extiende las capacidades de Eclipse para permitirnos construir rápidamente proyectos Android, añadir librerías necesarias de la API de Android, depurar nuestras aplicaciones con herramientas diseñadas específicamente o exportar los archivos necesarios para instalar la aplicación en los dispositivos.

2.1.2 Android

Android es un sistema operativo diseñado para dispositivos táctiles. Una de sus grandes características es que está basado en Linux, sistema operativo libre y multiplataforma. El sistema utiliza una variación del lenguaje de programación Java llamado Dalvik. Además con Android se nos proporciona diferentes interfaces con las cuales podemos acceder a las distintas funciones del dispositivo como la cámara, que usaremos en este proyecto.

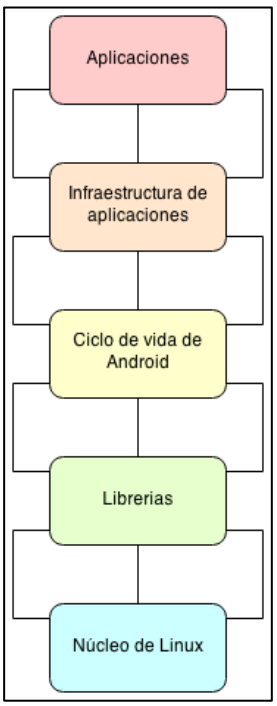


Ilustración 1 - Arquitectura Android

Un punto a recalcar es que Android es un sistema que ha ido sufriendo variaciones a lo largo de los años por lo que es posible que algunas funciones descritas en este trabajo no se mantengan conforme avance el tiempo.

Si queremos desarrollar en Android tenemos que utilizar el SDK de Android [1], que nos proporciona las librerías API y las herramientas necesarias para construir, testear y depurar aplicaciones. El SDK viene incorporado en el Eclipse ADT mencionado en el anterior apartado.

Para entender un proyecto Android es necesario entender el ciclo de vida de sus aplicaciones. Cada una de las actividades o clases que creamos en un proyecto Android tienen su ciclo de vida creado a partir de una serie de llamadas a métodos. A diferencia de otros paradigmas cuyas aplicaciones inician mediante un método main(), el sistema de Android inicia su actividad invocando una serie de llamadas que corresponden a las diferentes etapas de su ciclo de vida. Existen una serie de métodos que inician una actividad y otras que la suspenden.

En la siguiente imagen veremos representado el ciclo de vida mediante una pirámide de pasos. En la ilustración observamos que por cada llamada que avanza a partir del estado "Resumido" de la pirámide hay otra llamada por la cual volvemos atrás en el ciclo de vida. La actividad también puede volver a el estado "Resumido" desde el estado "Pausado".

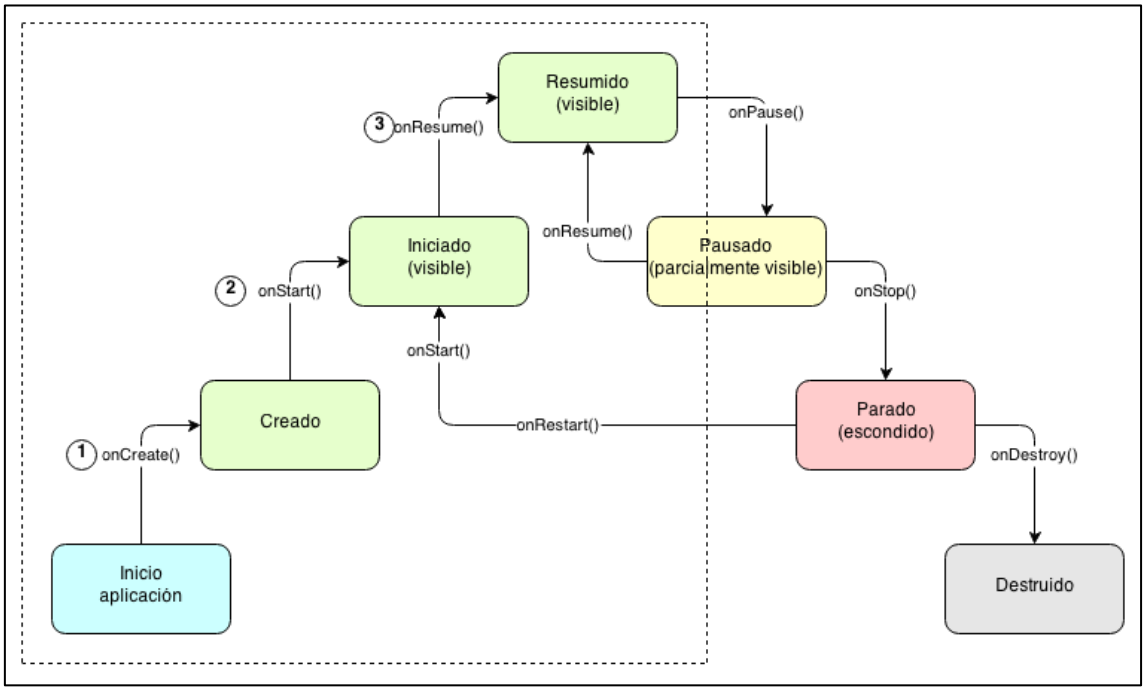


Ilustración 2 - Ciclo de vida Android

Existen tres tipos de situación en la que una actividad puede encontrarse:

- Resumido: En este estado, la actividad puede interactuar con el usuario.
- Pausado: En este estado, la actividad está parcialmente bloqueada por otra actividad, esta otra actividad está por encima de la anterior y es con la que el usuario interactúa.
- Parado: En este estado, la actividad está completamente escondida y no esta visible al usuario. La actividad y toda su información se retiene pero no ejecuta código.

En la ilustración anterior también observamos un cuadrado de línea discontinua que nos indica las tres principales llamadas que el sistema realiza en secuencia para crear una nueva actividad. Una vez esta secuencia se completa, la actividad alcanza el estado "Resumido" donde el usuario puede interactuar con la actividad hasta que se cambie a otro estado diferente.

2.1.3 OpenCV

OpenCV es una librería libre para tratar visión artificial [3]. Fue diseñada para ser computacionalmente eficiente y con un gran enfoque a aplicaciones de tiempo real. Es muy conocida y se ha utilizado en infinidad de aplicaciones, como en sistemas de seguridad o en sistemas de reconocimientos de objetos.

Existen diferentes versiones debido a que tiene interfaces en lenguajes como Python o Java. La versión se utiliza en este proyecto es la de Android, llamada OpenCV4Android. Usaremos esta librería para desarrollar la parte de detección automática, ya que OpenCV proporciona una serie de funciones y herramientas que hacen fácil el tratamiento de imágenes.

En el proyecto usamos la faceta de OpenCV para tratar con imágenes y obtener cierto tipo de información de ellas para después contrastarla y estudiar sus resultados.

2.2 Estado actual

Los sistemas que realizan tareas parecidas a las que está enfocado este proyecto son pocos. Si investigamos entre las aplicaciones que ofrece el mercado de Android, descubriremos que existen algunas aplicaciones las cuales sólo permiten dibujar medidas en imágenes estáticas tal y como lo hace nuestra aplicación.

Si nos centramos en la medición introduciendo medidas a mano, observaremos que existen algunas aplicaciones, muy similares entre ellas, que están disponibles en el mercado de Android:

- Photo Measures Lite [4]: Con esta aplicación podremos tomar fotográficas y dibujar las medidas directamente sobre ella, así como añadir texto o ángulos.
- Measure & Sketch [5]: Este programa realiza prácticamente la misma actividad al anterior, dibujar medidas sobre una imagen, cambiando alguna que otra funcionalidad.

Hemos comentado dos de las pocas aplicaciones que existen para poder medir distancias con un dispositivo móvil. Al analizarlas comprobamos que todas ellas realizan la misma tarea. En nuestro caso hemos implementado esta característica en el proyecto dándole otro enfoque, ya que las medidas se calculan solas a partir de una inicial.

A continuación, el estudio se ha realizado para las aplicaciones que miden distancias o alturas de objetos o lugares. En el proyecto también se encuentra desarrollada esta característica. Se han realizado diversos enfoques de la medición de distancias por que se consideraba interesante tener clara cada una de las posibilidades y una vez analizadas se ha conseguido implementarlas. Se puede comprobar que en el mercado de Android existen también aplicaciones que realizan este trabajo:

- Telémetro - Smart Measure [6]: Con esta aplicación se puede medir la distancia, la altura y el ancho de un objeto mediante trigonometría. El uso es muy simple y con unos sencillos pasos se puede medir las distancias deseadas.

En este tipo de programas se hace uso de trigonométrica básica para calcular las distancias. Se usan los sensores incorporados en los dispositivos para calcular el ángulo necesario en el cálculo de las distancias. Es completamente imprescindible que mientras se realice las medidas de la distancia o la altura, se muestre en tiempo real los valores en sus respectivas unidades de medida para que el usuario pueda calibrar adecuadamente las medidas. Además, la aplicación cuenta con una interfaz que viene sobre la previsualización de la cámara y que cuenta con un punto de mira para que la medición sea la más precisa posible.

Más adelante se explicarán los pasos seguidos para calcular estas distancias, ya que en el proyecto también se desarrolla una aproximación a esta característica.

2.3 Innovación

Para la realización del proyecto se han usado herramientas proporcionadas por Android o por OpenCV. En todas las tareas se han usado diferentes formas de realizarlas pero siempre usando los elementos proporcionados. No es necesario invertir tiempo en crear nuevas herramientas que realicen un determinado trabajo en tu aplicación si ya existen herramientas creadas para ese fin, y que probablemente sean mucho mejores que las que uno mismo realice.

No hace falta volver a inventar algo. Hay que buscar algoritmos, librerías, artículos, libros, etc. que implementen o describan la solución a alguno de los problemas que estamos resolviendo. Para cualquier ingeniero es importante saber apoyarse en tecnologías existentes para conseguir una mayor productividad. De esta forma, las metas del proyecto pueden ser más ambiciosas, con lo que desarrollamos algo realmente útil.

Un enfoque algo distinto es intentar desarrollar aplicaciones que den una vuelta de tuerca a las existentes. En el caso de este proyecto existen algunas aplicaciones que ya hacen las actividades que deseamos implementar. No por ello debemos dejar de lado el proyecto ya que el fin más importante de este estudio es el aprendizaje y no la creación de un programa innovador.

En el presente proyecto se intenta dar un enfoque de innovación al método de medición introduciendo un elemento que proporciona medidas preestablecidas para que el usuario pueda realizar otras medidas al instante. Este elemento automático no se encuentra extendido en las diversas aplicaciones que realizan mediciones. Se puede considerar que el trabajo tiene un punto de innovación característico que hace distinguirse entre las demás aplicaciones.

Un punto en contra de innovar en un trabajo de estas características es el tiempo acotado en el que se realiza. Normalmente la innovación se consigue tras mucho esfuerzo y tiempo por lo que en este trabajo se presenta la característica de la detección automática como algo básico fruto del trabajo de un estudiante de ingeniería.

3. Análisis

El apartado de análisis pretende dar respuesta a diferentes aspectos de la aplicación planificados previamente a su implementación. Este punto es quizás el más importante ya que un buen análisis y planificación ahorra mucho trabajo y esfuerzo futuro.

3.1 Descripción de la solución

Este apartado contesta a la pregunta ¿Qué hace la aplicación? Explicaremos qué es lo que realizan las distintas tareas del programa. Tenemos dividida la aplicación en siete módulos. La medición automática y la medición manual tienen dos cada uno.

Para la medición manual tenemos que tener una imagen sobre la cual el usuario dibuja con el dedo líneas que representan medidas. Se le pide que la primera medida introducida sea una conocida por él, y una vez insertada se pueden dibujar más líneas cuyas medidas se calcularán respecto a la primera medida introducida.

En el apartado de la medición automática, vamos un paso más allá de la medición manual. En este tipo de tarea nos saltamos el paso de pedirle al usuario que introduzca una medida conocida por él. El usuario deberá fotografiar un objeto (en nuestro proyecto es una tarjeta de crédito) junto a los objetos que quiera medir. La aplicación calculará automáticamente donde está el objeto de referencia y mostrará las medidas del objeto dibujadas encima de él. El usuario se olvida de introducir medidas automáticas y simplemente tiene que dibujar las líneas de las medidas que quiere calcular. Las medidas dibujadas por el usuario se calcularán a partir de las medidas obtenidas del objeto de referencia.

Para el cálculo de la distancia y altura a un objeto se le mostrará al usuario una pre visualización de la cámara junto a unas líneas en cruz en el centro de la pantalla. Para el cálculo de la distancia el usuario debe apuntar la cruceta a la base del objeto, de esta manera se obtiene en tiempo real la distancia al objetivo. El usuario podrá congelar la imagen y la medida para obtener el resultado. De la misma manera se trabaja para el cálculo de la altura de un objeto. Primero se le pide al usuario calcular la distancia de la misma manera descrita anteriormente. Una vez se tiene la distancia al objeto el usuario deberá enfocar la cruz de la aplicación hacia la zona más alta del objeto para obtener la altura. De igual manera que la distancia, se puede congelar la altura para obtener el resultado.

Para la obtención de la imagen de referencia, el usuario inicia una tarea en la que se le pide fotografiar el objeto ayudado de una plantilla. Como en nuestro caso el objeto de referencia utilizado es una tarjeta común, la plantilla será un rectángulo con las mismas proporciones que una tarjeta. Cuando el usuario realiza la fotografía, el programa la guarda en memoria para ser usada posteriormente.

3.2 Descripción del flujo de la aplicación

En este apartado se detalla cómo fluye la información por toda la aplicación. Para realizar este paso, es indispensable aclarar la estructura del programa, por ello se ha realizado un esquema en el que se distribuyen las distintas partes de la aplicación y que se muestra a continuación:

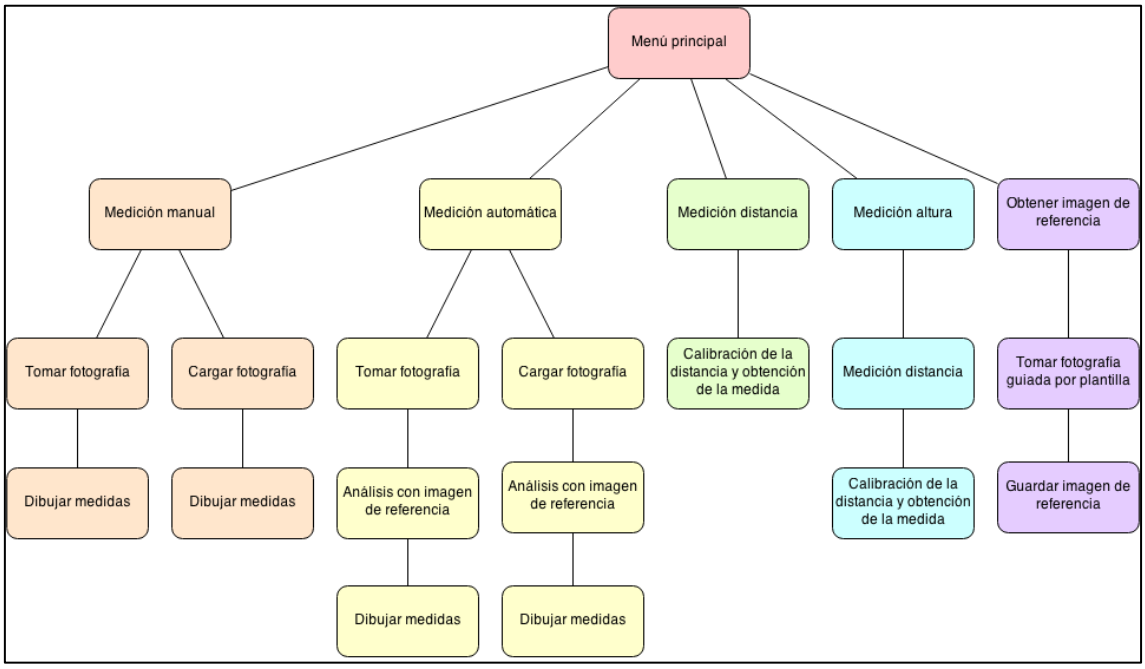


Ilustración 3 - Esquema de la aplicación

En el esquema se observa que desde el menú principal se acceden a todas las partes de la aplicación que han sido implementadas. También se aprecia que existen cinco principales tareas que el programa puede realizar marcadas con distintos colores.

La primera de ellas es la medición manual de una imagen. La segunda es la medición de una imagen ayudándonos de un objeto que se detecta de forma automática y que nos proporcionara las medidas con las que se puede calcular las que el usuario introduzca. La tercera rama con el color verde es la actividad de medición de una distancia hacia un objeto. La cuarta con el color azul permite medir la altura de un objeto. La última actividad guardará la imagen del objeto de referencia en nuestro dispositivo que es usada por la medición automática.

En los siguientes apartados se muestran las ilustraciones que son diagramas de flujo pertenecientes a las cinco actividades nombradas anteriormente y que detallan el camino de la información en nuestra aplicación cuando el usuario hace uso de ellas.

3.2.1 Medición manual

El enfoque que se ha dado para la selección de la fotografía ha sido el poder dar a elegir al usuario entre dos opciones. La primera de ellas es la toma de la fotografía mediante la cámara incorporada en el dispositivo y la segunda es la carga de la imagen a través de la galería de Android, desde una colección de fotografías guardadas en el aparato. Una vez cargada la foto se le muestra al usuario para que realice los pasos pertinentes.

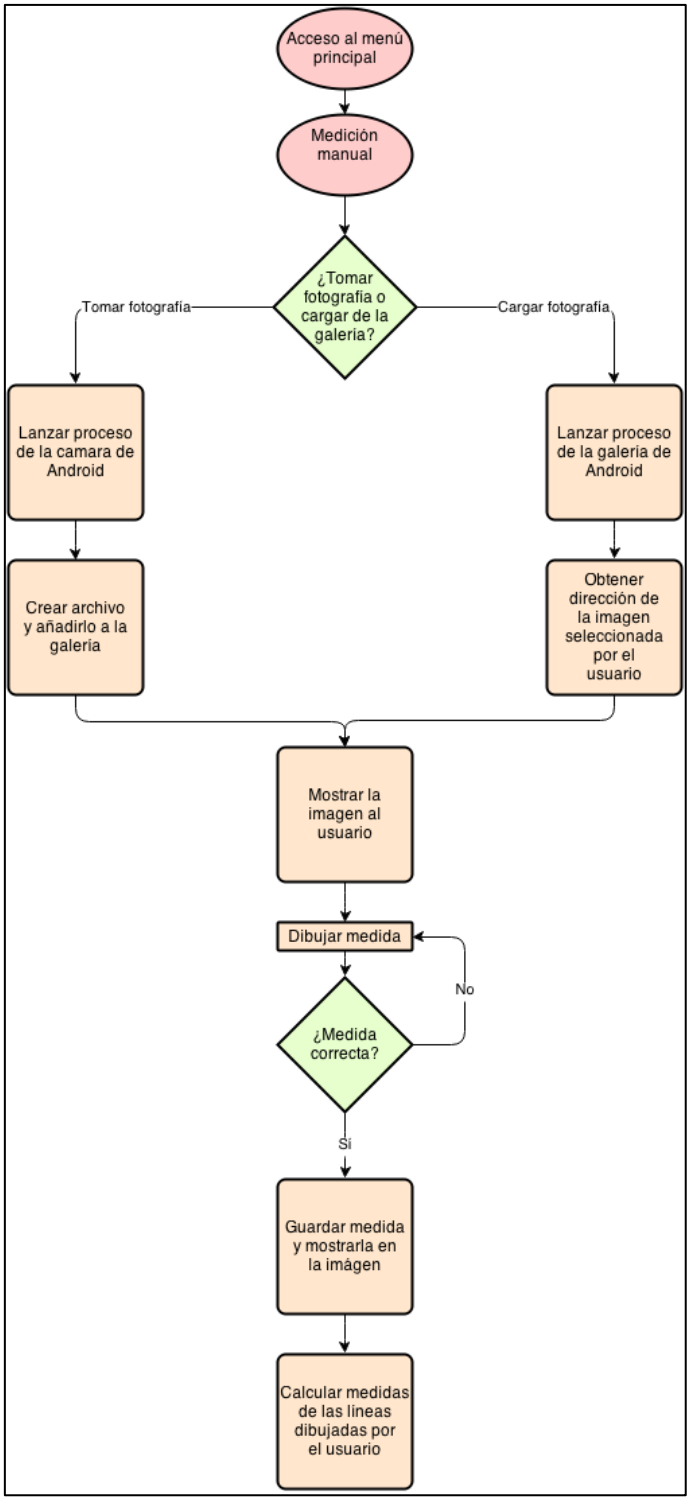


Ilustración 4 - Flujo medición manual

3.2.2 Medición automática

En la medición automática se realizan los primeros pasos de la misma manera que la medición manual, cargando la imagen desde la cámara o la galería. Seguidamente, antes de mostrar la imagen al usuario, se realiza la búsqueda automática de la imagen de referencia en la imagen seleccionada. Si la comprobación es correcta, se dibujará en la imagen las medidas del objeto de referencia. El usuario podrá, a partir de ahora, dibujar líneas en la imagen para obtener sus respectivas medidas.

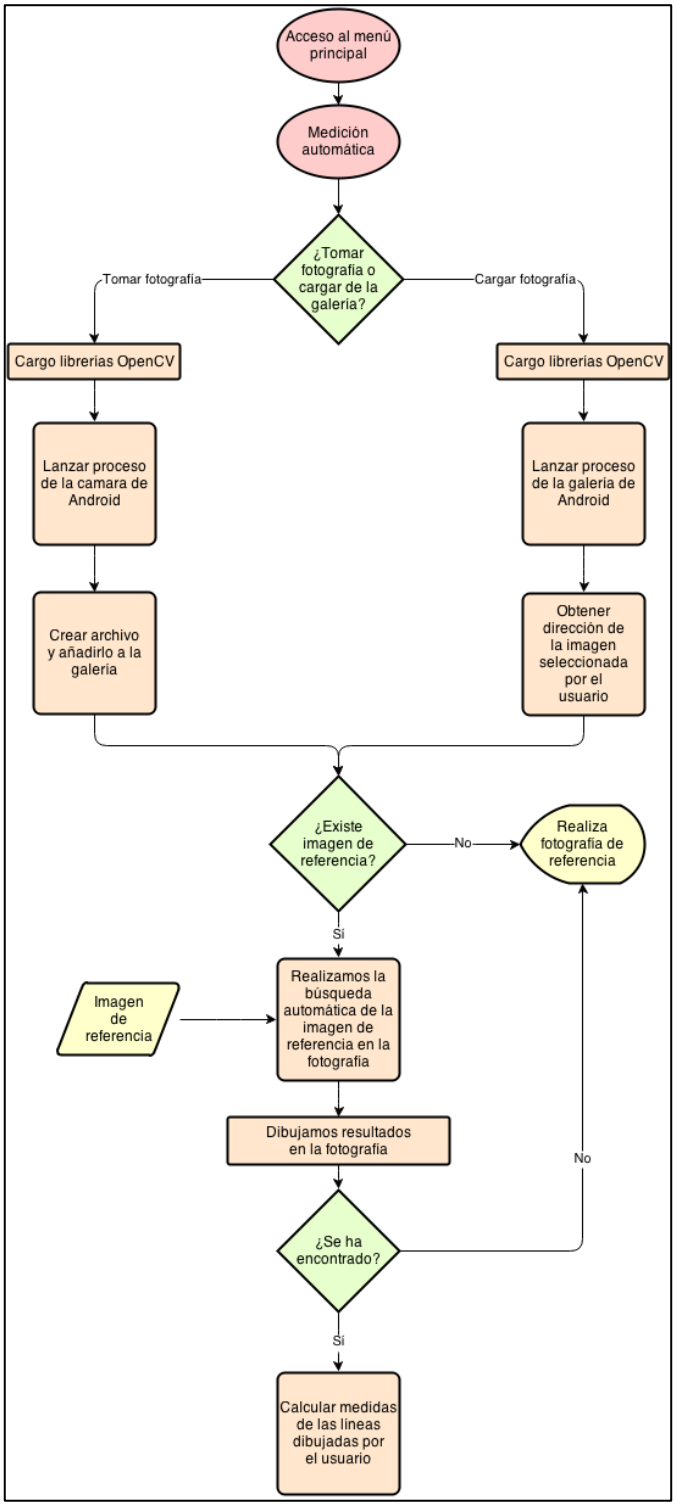


Ilustración 5 - Flujo medición automática

3.2.3 Medición distancia

En el proceso de medir la distancia de un objeto se hace hincapié en el manejo de la cámara. Para poder permitir al usuario enfocar en tiempo real se tiene que realizar una previsualización de la cámara al mismo tiempo que se calcula y muestra la distancia en una zona del dispositivo. También es necesario dibujar una pequeña cruz en mitad de la pantalla para que el usuario pueda enfocar adecuadamente a la base del objeto.

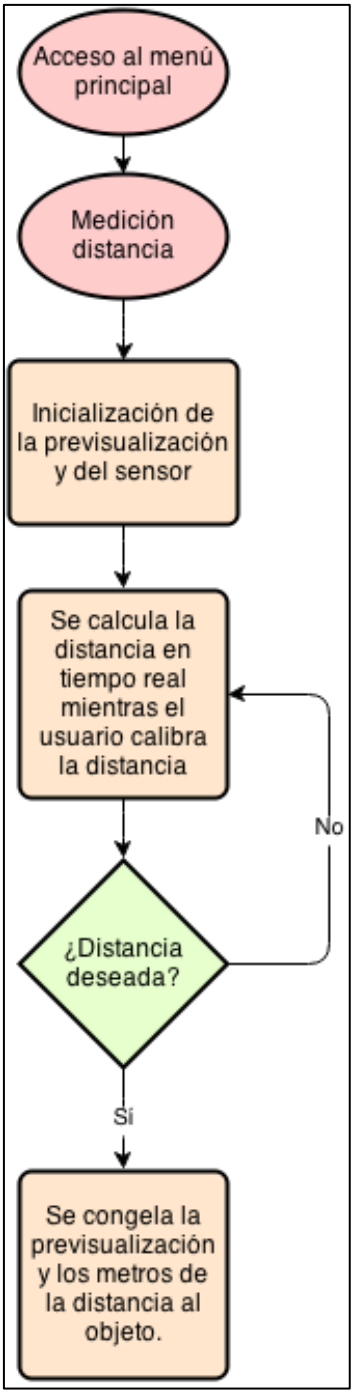


Ilustración 6 - Flujo medición distancia

El usuario puede congelar la imagen mediante un botón en el momento que desee. En ese instante la distancia quedará también parada para que se pueda consultar cómodamente.

3.2.4 Medición altura

Para medir la altura de un objeto es necesario obtener anteriormente la distancia a él. Es por esto por lo que necesitamos calcular antes la distancia para poder medir la altura. Como observamos en el flujo siguiente, la primera parte de esta tarea es idéntica al anterior apartado, por lo que una vez se obtiene el dato de la distancia se calcula también en tiempo real, la altura del objeto. En la previsualización se muestran por igual, la distancia y la altura.

Ilustración 7 - Flujo medición altura

3.2.5 Obtener imagen de referencia

Esta actividad es crucial para el correcto funcionamiento de la medición automática. En esta actividad nos encontramos con una previsualización de la cámara con una plantilla dibujada encima de ella. Para realizar los ejemplos durante todo el desarrollo del proyecto se ha utilizado una tarjeta común como puede ser una carnet de identidad o tarjeta de crédito, por lo que la plantilla en este caso será un rectángulo. El usuario debe enfocar la tarjeta que use de ejemplo ayudándose de la plantilla, para, posteriormente guardarse en el dispositivo y ser utilizada en su momento.

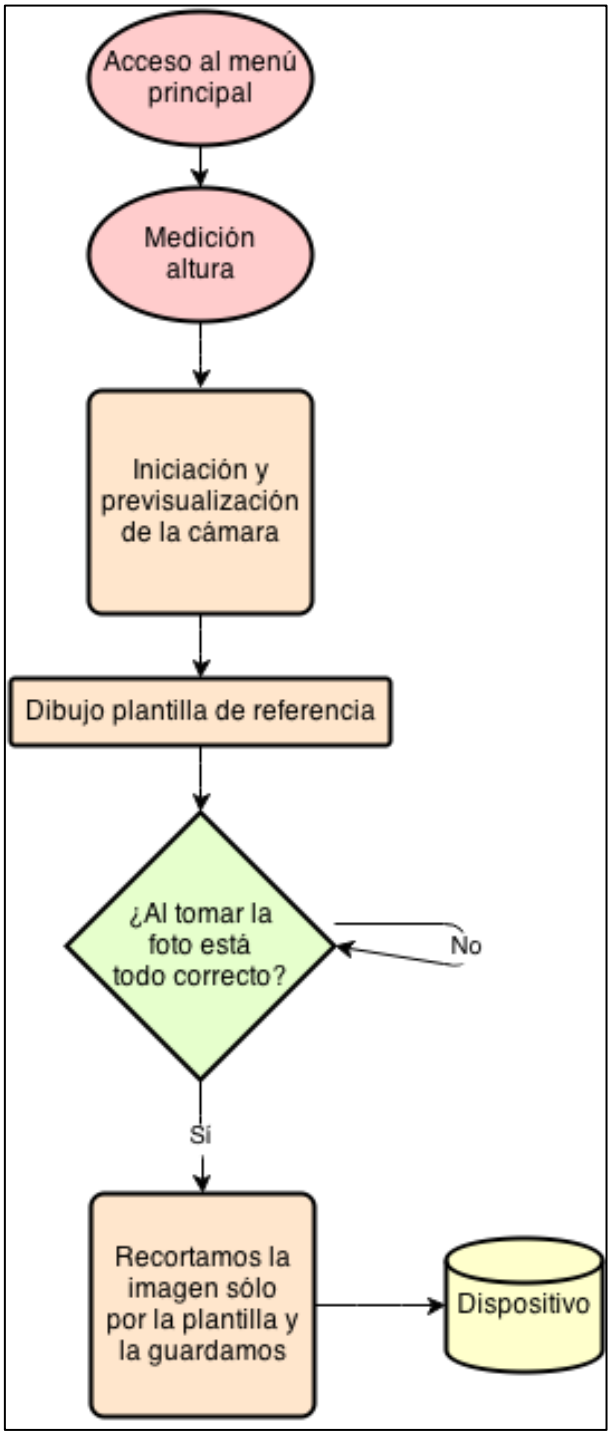


Ilustración 8 - Flujo obtener imagen de referencia

3.3 Descripción de los algoritmos utilizados

En el proyecto se ha utilizado una técnica de matching de imágenes llamada feauture matching. Las librerías de OpenCV para Android ayudan a utilizar este tipo de técnicas para el tratamiento de imágenes. Se ha escogido esta librería de tratamiento ya que es una de las más conocidas y con mayor documentación en esta área.

Al abordar el tema de la detección automática de un objeto se estudiaron diferentes maneras de realizarla, al final, la más factible es la comparación de una imagen dentro de otra que normalmente es más grande y contiene a la anterior de alguna forma. Este tipo de técnicas se las conoce por el nombre en inglés de *matching* de elementos.

Una vez se tiene claro el tipo de técnica a usar, se investiga las utilidades que nos ofrece la librería OpenCV. En un principio se propuso usar la técnica de template matching, se llegó a implementar incluso una versión totalmente funcional de esta técnica para la aplicación pero tras diversas comprobaciones se observó que es una técnica muy limitada y apenas permite variación entre la imagen de referencia y la imagen en la que buscamos.

Aun teniendo implementada la versión de template matching se decidió investigar otro tipo de técnicas. Se ha implementado pues, la técnica de feature matching a pesar de que es una técnica un poco más compleja, sin embargo; debido a su naturaleza, es posible utilizar imágenes de referencia que no sean exactamente iguales a zonas que se buscan en la imagen original. Por este motivo se decide dejar como versión final de la aplicación la búsqueda automática mediante feature matching. A continuación se explica con más detalle cada una de las técnicas investigadas para la medición automática creada para el proyecto.

3.3.1 Template Matching

El template matching es una técnica usada para la búsqueda de áreas de una imagen que concuerda (o es similar) a una imagen que sirve de plantilla (o zona). El funcionamiento de la técnica es simple y se necesitan dos componentes fundamentales:

- Imagen fuente: La imagen donde esperamos encontrar la imagen de referencia.
- Imagen plantilla (o imagen referencia): El área de la imagen que queremos que se compare a la imagen fuente.

El objetivo de esta técnica es detectar el área resaltada:

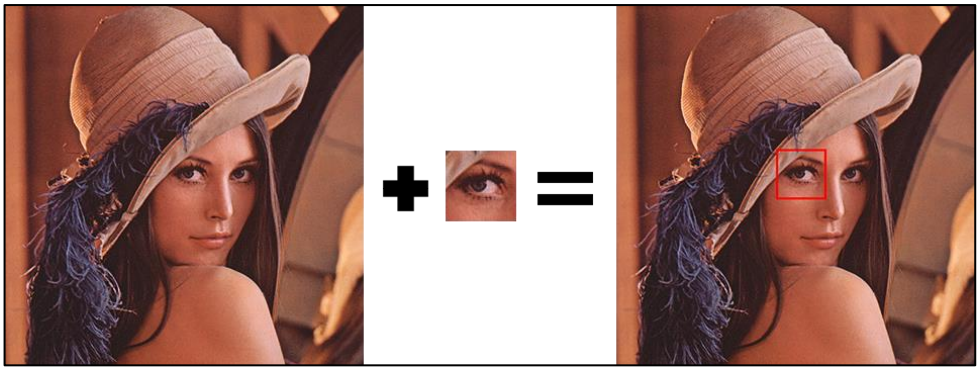


Ilustración 9 - Objetivo template matching

Para identificar el área de referencia tenemos que comparar la imagen que usaremos de plantilla contra la imagen fuente deslizándola a través de toda ella:

Ilustración 10 - Funcionamiento template matching

Cuando se dice deslizar la plantilla, nos referimos a moverla pixel a pixel (de izquierda a derecha, arriba abajo). En cada localización, una métrica se calcula para representar cuan idéntica es o diferente.

3.3.2 Feature Matching

Es un método que toma decisiones locales en cada punto de la imagen teniendo por resultado una serie de puntos distintivos para cada uno de esos puntos. El método puede decidir si la zona que está comprobando puede resultar significativa o no.

No hay una definición exacta de lo que constituye un feature (característica, punto distintivo). La definición exacta de lo que es depende del problema y del tipo de aplicación. Por lo general un punto distintivo es una parte interesante de una imagen, por lo que en diferentes imágenes que pueden representar lo mismo encontraremos puntos que se asemejan entre ellos.

Cuando se aplica la técnica a dos imágenes diferentes podemos obtener como resultado una serie de puntos distintivos. Si se quiere buscar una imagen de plantilla en otra imagen fuente, simplemente tenemos que aplicar esta técnica a las dos imágenes. Los resultados obtenidos se podrán comprobar mediante herramientas proporcionadas por OpenCV y obtener una relación, esta relación nos indicará si se han encontrado puntos comunes en las dos imágenes, si existen una gran cantidad de puntos comunes se puede decir que una imagen contiene a la otra.

A continuación se observa en una imagen como actúa la técnica de feature matching. Los círculos son puntos característicos que se han encontrado y están enlazados mediante una línea a sus respectivas copias en ambas imágenes. Hay que tener en cuenta que la siguiente imagen es un ejemplo, normalmente el método encuentra una gran cantidad de puntos en ambas imágenes y luego compara cuáles de ellos tienen semejanza. Para que el método resulte efectivo, cuantos más puntos comunes encuentre mejor será el resultado obtenido en la solución.

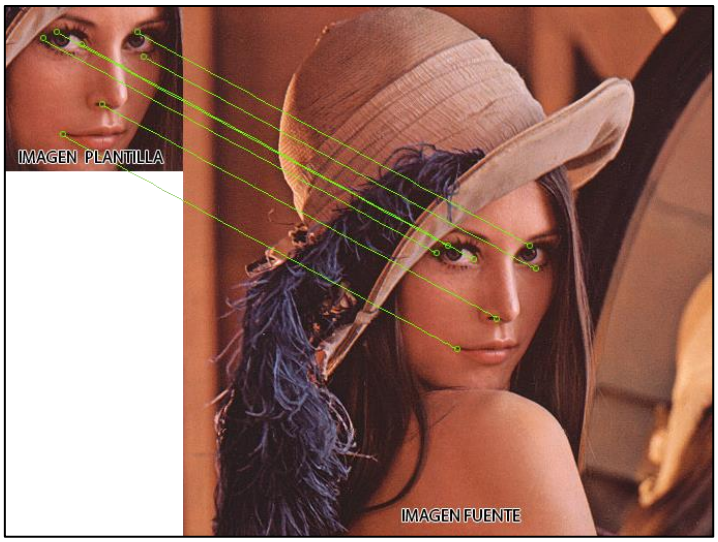


Ilustración 11 - Resultado feature matching

3.4 Decisiones tomadas

El objetivo de este apartado es explicar el porqué de algunas partes de la aplicación. Hay decisiones que son necesarias tomar para que la aplicación se complete adecuadamente. Algunas de las decisiones que se toman pueden influir adecuadamente y otras no. Son aquellas decisiones que recortan la funcionalidad de la aplicación o que reducen su eficiencia las que hay que evitar pero no siempre se puede.

Una de las decisiones referentes a la interfaz de usuario fue respecto al menú que tendría la aplicación. Se decidió que el menú fuera simple, por lo que se escogió una barra inferior que proporciona instrucciones con dos botones a los lados. Esta decisión fue tomada debido a que es necesario reducir la información de la pantalla en aplicaciones basadas en las imágenes que se muestren a usuarios o a las que usen previsualizaciones de la cámara.

Otra decisión que se ha tomado es el uso de un menú que engloba todas las tareas de la aplicación. Si estudiamos detenidamente la aplicación, es posible organizar las actividades de una forma más dinámica como por ejemplo, incluir la actividad de obtener la imagen de referencia en la barra de acción de Android. La justificación del menú común es la necesidad de mostrar al usuario todas las opciones disponibles de la aplicación. También es mucho más cómodo tener todas las opciones en un mismo sitio.

Se ha optado por guiar al usuario a través de la aplicación mediante mensajes en la barra inferior creada para cada actividad. Con estos mensajes aclaramos al usuario las acciones que debe realizar en cada momento, de esta manera queda claro que acciones puede realizar en ese mismo instante.

Finalmente la última decisión importante tomada es la utilización del método feature matching y no otro. Como se ha comentado en el apartado anterior, el método de feature matching es tiene un nivel de implementación medio y suele arrojar buenos resultados. Se descarta la utilización del método de template matching ya que no se puede implementar para tamaños de plantilla variantes.

3.5 Lista de requisitos

En este apartado se detallan los requisitos de la aplicación. Los requisitos se obtienen de los objetivos iniciales del proyecto, ordenándolos y especificándolos de forma clara. Los requisitos ayudan a completar el funcionamiento del sistema a desarrollar. Se han dividido en dos tipos: funcionales y no funcionales.

3.5.1 Requisitos funcionales

Este tipo de requisito son los que el usuario necesita que efectúe el software. Normalmente este tipo de requisitos explican el qué realizan ciertas actividades de nuestra aplicación. El usuario debe esperar una respuesta del sistema una vez se inicien este tipo de acciones. Seguidamente se listan los requisitos funcionales:

Identificador:	**F01**
Título:	Inicio aplicación
Descripción:	La aplicación se lanzará con el icono correspondiente en el dispositivo Android.

Identificador:	**F02**
Título:	Menú principal
Descripción:	La aplicación dispondrá de un menú principal desde el cual acceder a todas las características de la aplicación.

Identificador:	**F03**
Título:	Medición manual - Interfaz
Descripción:	La interfaz de esta tarea debe ser simple para mostrar el mayor área de la imagen posible y que el usuario pueda dibujar fácilmente en ella.

Identificador:	**F04**
Título:	Medición manual – Tomar fotografía
Descripción:	La aplicación debe dar la opción de poder tomar una fotografía para su posterior medición manual.

Identificador:	**F05**
Título:	Medición manual – Cargar galería
Descripción:	La aplicación debe dar la opción de poder cargar de la galería una fotografía para su posterior medición manual.

Identificador:	**F06**
Título:	Medición manual - Interfaz
Descripción:	La interfaz de esta tarea debe ser simple para mostrar el mayor área de la imagen posible y que el usuario pueda dibujar fácilmente en ella.

Identificador:	F07
Título:	Medición manual – Tomar fotografía
Descripción:	La aplicación debe dar la opción de poder tomar una fotografía para su posterior medición automática.

Identificador:	F08
Título:	Medición manual – Cargar galería
Descripción:	La aplicación debe dar la opción de poder cargar de la galería una fotografía para su posterior medición automática.

Identificador:	F09
Título:	Dibujar para medir
Descripción:	El usuario debe poder introducir líneas para su posterior medida de forma fácil y rápida.

Identificador:	F10
Título:	Sistema de instrucciones
Descripción:	La aplicación debe de guiar al usuario mientras la use. Deben mostrarse mensajes indicando al usuario las actividades a realizar.

Identificador:	F10
Título:	Mostrar medida en tiempo real
Descripción:	En las aplicaciones de medir distancia y altura, se debe mostrar las medidas en tiempo real para que el usuario pueda corregir los posibles errores.

Identificador:	F11
Título:	Fallos con mensajes
Descripción:	La aplicación debe avisar al usuario con un mensaje si el usuario intenta realizar una medida automática sin la imagen de referencia.

3.5.2 Requisitos no funcionales

Este tipo de requisitos son los recursos que necesita el sistema para realizar las actividades correspondientes, como que la base de datos debe ser una determinada. Estos tipos de requisitos especifican la manera de trabajar del sistema, por ejemplo, de manera rápida o el uso fácil de la interfaz.

Identificador:	NF01
Título:	Toma de fotografías
Descripción:	El sistema debe ofrecer un sistema de toma de fotografías robusto. Se hará uso de la propia aplicación nativa de Android.

Identificador:	**NF02**
Título:	Accesos
Descripción:	Al usar la aplicación se aceptan los permisos de escritura y lectura del dispositivo. Además también se dará permiso del uso de la cámara.

Identificador:	**NF03**
Título:	Cargas de la aplicación
Descripción:	Las cargas de la aplicación deben ser en el menor tiempo posible. Esto se puede conseguir cargando la pantalla actual, y lanzando los demás procesos de forma asíncrona.

Identificador:	**NF04**
Título:	Instalación aplicación
Descripción:	Si se quiere instalar la aplicación se hará mediante el archivo .apk generado por el framework de desarrollo.

Identificador:	**NF05**
Título:	Consistencia de formato
Descripción:	El formato utilizado para guardar las imágenes generadas en la aplicación debe ser siempre el mismo.

Identificador:	**NF06**
Título:	Interfaz consistente
Descripción:	La interfaz de toda la aplicación debe mantener una consistencia para no confundir al usuario.

La implementación de todos los requisitos listados anteriormente es un proceso ideal y dará como resultado una aplicación estable y perfectamente funcional que el usuario podrá usar sin problemas.

4. Diseño

En el apartado de diseño se entra en detalle cómo funciona nuestra aplicación. Se evita introducir código (a no ser que sea estrictamente necesario) ya que en ocasiones puede llevar a un mal entendimiento de la lectura de la memoria o en ocasiones al total desconcierto. El código de la aplicación ira aparte, junto a esta memoria. El fin de este apartado es contestar a la pregunta ¿cómo lo hace la aplicación? Referente a la pregunta contestada en el apartado anterior ¿qué hace la aplicación?

4.1 Diagrama de clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y como se relacionan unas con otras. Los diagramas de clases son diagramas "estáticos" por que muestran las clases, junto con sus métodos y atributos. Con estos tipos de diagramas podemos observar qué clases conocen a qué otras clases o qué clases son parte de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

En la siguiente ilustración se muestra el diagrama de clases de la aplicación. En Android cada una de las clases que se invocan para mostrar datos en el dispositivo se llaman "Activity", en estas clases siempre deben aparecer cierto tipos de métodos para el correcto funcionamiento. El entorno Eclipse suele introducir estas clases automáticamente.

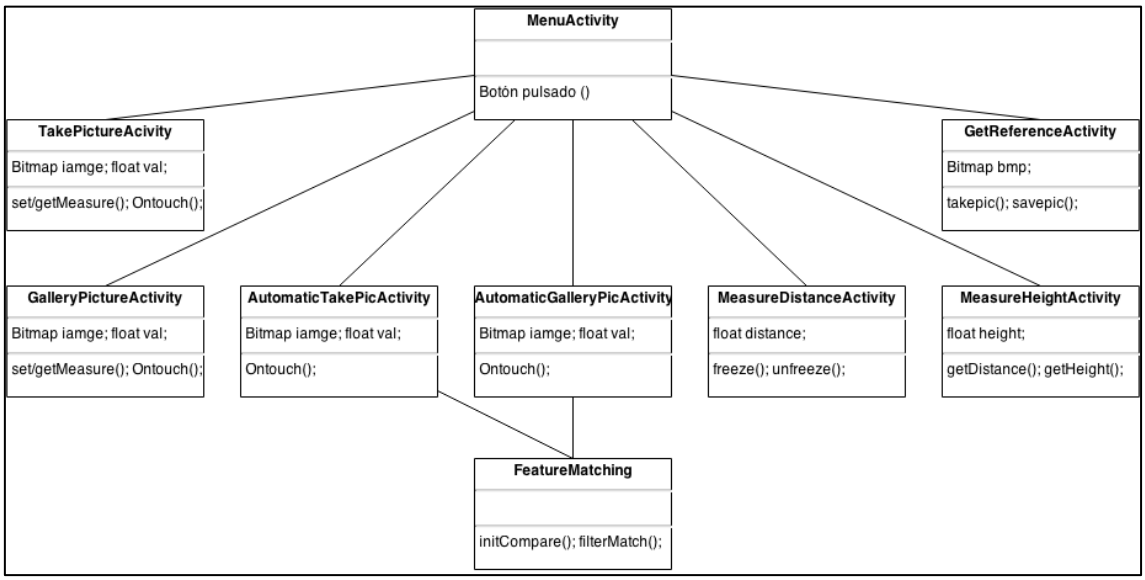


Ilustración 12 - Diagrama de clases

Para cada una de las clases se muestran el título (fila superior), los atributos más importantes de la clase (fila intermedia) y los métodos más importantes (fila inferior). Observamos que cada clase es independiente de la otra ya que en el proyecto apenas se ha introducido una subdivisión de objetos adecuada. Los métodos automáticos obtienen la funcionalidad de la detección de una clase común.

4.2 Funcionamiento de la aplicación

Con los datos obtenidos durante la fase de análisis, se realiza un diseño de los componentes de la aplicación, de forma que todos los aspectos queden claros y definidos para la fase de implementación. Con toda la información recogida no será necesario replantear ningún punto. El diseño se ha de adaptar a las necesidades del software y las clases se han de definir de manera exhaustiva para que las relaciones entre ellas queden claras.

En los siguientes apartados describiremos como realizan su trabajo algunos de los elementos esenciales que se han usado para la elaboración de la aplicación.

4.2.1 Toma de fotos y carga de imágenes

Vamos a empezar explicando el funcionamiento de la toma de imágenes y la carga de imágenes a través de la galería de Android. En el proyecto se usa el propio gestor de Android para realizar tales actividades. Para realizar esta actividad se tiene que invocar a un "Intent" que en Android se desarrolla como una actividad que se ejecuta por encima de aquella que la ha llamado. Por lo tanto cuando queremos tomar una foto o cargar una imagen de la galería tenemos que realizar este paso.

4.2.2 Dibujo en imágenes

Un elemento importante en la aplicación son los dibujos de líneas y números que se realizan cuando el usuario arrastra el dedo por la imagen para dibujar una línea. Una vez tenemos la imagen que el usuario ha elegido en la pantalla del dispositivo, procedemos a "escuchar" mediante el método onTouch de Android. Con este método obtenemos fácilmente las coordenadas de cualquier posición del dedo del usuario para su posterior dibujado.

Para dibujar las líneas y números en Android se hace uso de la clase Canvas, esta clase se tiene que crear pasándole la imagen en la que queremos dibujar.

```
Canvas canvas = new Canvas(image);
```

Luego, para dibujar cualquier elemento en el simplemente accederemos a los métodos de la clase como drawLine o drawCircle; introduciendo las coordenadas necesarias para que se pueda dibujar. A esta clase también le pasaremos un elemento de tipo Paint. Este elemento representa las características del trazo del dibujo, como el color o el grosor del trazo.

4.2.3 Previsualización de la cámara

Para la previsualización de la cámara se ha usado la clase Camera de Android. Antes de introducir el código es necesario crear un elemento del tipo SurfaceView en la interfaz de usuario que en este proyecto ocupa toda la pantalla. Una vez realizado este paso, en el código procederemos a abrir la cámara con su método open en el método onStart de Android (ejecutado al iniciarse la actividad). Es necesario también añadir un elemento surfaceHolder que manejarala conexión entre la camra y el SurfaceView.

4.2.4 Diagrama de estados

Los diagramas de estados son una técnica conocida para describir el comportamiento de un sistema. Describen todos los estados posibles en los que puede aparecer el sistema y la manera en que cambia el estado del sistema.

Las ventajas que ofrece esta técnica es que permite centrarnos en las necesidades del usuario. También es un sistema de éxito en sistemas interactivos ya que expresa la intención que tiene el usuario al hacer el uso del sistema.

En el siguiente esquema se representa el diagrama de estados de la aplicación. Los colores iguales representan estados de similitud en los que se encuentra el sistema. Cada uno de los círculos representa un estado diferente al que se accederá mediante otro estado de la aplicación. Las flechas indican la transición de un estado a otro por una determinada acción. Recordaremos que la aplicación se inicia en el menú principal por lo que el estado "Menú de inicio" será el estado de partida.

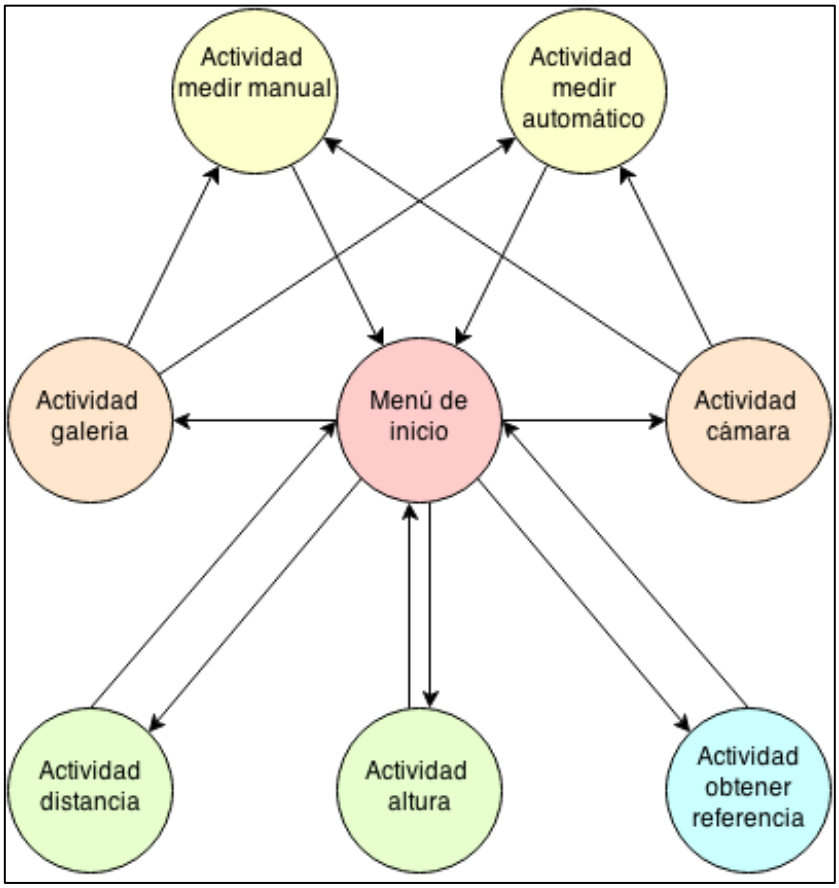


Ilustración 13 - Diagrama de estados

Un diagrama de estados sirve para mostrar la vida de una aplicación. El diagrama indica mediante flechas eventos que hace que un estado cambie a otro y cuáles son las respuestas que genera este. En nuestro caso, la aplicación es manejada por el usuario y cada uno de los estados es elegido por el usuario por lo que cada acción que cambia de estado es iniciada por el usuario ya que el diagrama de estados se utiliza normalmente para describir los estados del dominio del usuario.

4.2.5 Funcionamiento del cálculo de la distancia y altura

Para la realización de las tareas del cálculo de la distancia y altura de un objeto, se hace uso del giroscopio incorporado en muchos dispositivos Android. Este tipo de sensor nos permite saber la posición que tiene en un momento determinado el dispositivo. Utilizando la API correspondiente de Android podemos obtener mediante unos pequeños cálculos, el ángulo del dispositivo respecto a un eje de coordenadas.

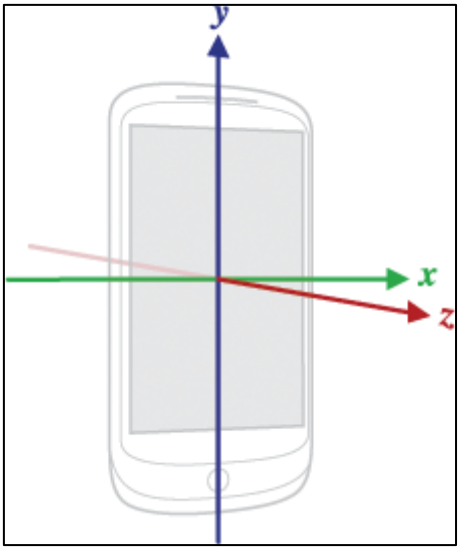


Ilustración 14 - Ejes sensor dispositivo móvil

Si observamos la imagen nos daremos cuenta que si un usuario enfoca con la cámara un determinado sitio el ángulo formado por el eje Z e Y será el que utilizaremos para realizar los cálculos.

Para explicar el cálculo de la distancia se ha realizado un pequeño esquema en donde se observa claramente la teoría detrás de los cálculos.

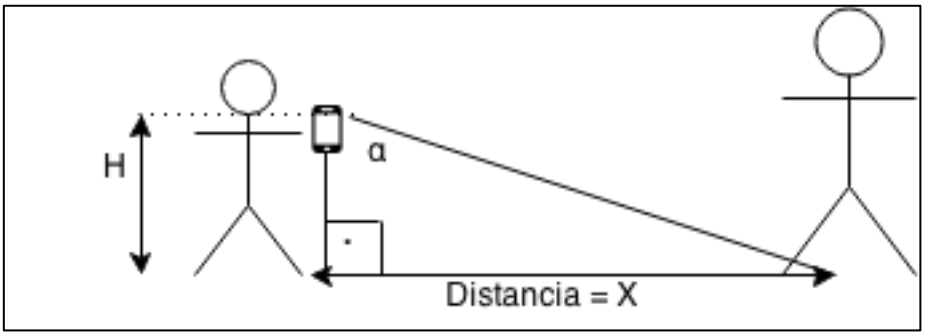


Ilustración 15 - Esquema cálculo distancia

Tenemos una constante H, que será la altura de la persona que utilice la aplicación, por defecto la aplicación inicia esta variable a 1,6 metros. Sabiendo el ángulo alfa (α) obtenido por el giroscopio del dispositivo podemos calcular X mediante el teorema del seno. No hay que olvidar que el ángulo comprendido entre H y X es de noventa grados por lo que el ángulo restante se puede calcular fácilmente:

$180 - \alpha - 90$

Si usamos el teorema del seno con todos estos datos, obtendremos la distancia X que es el resultado final.

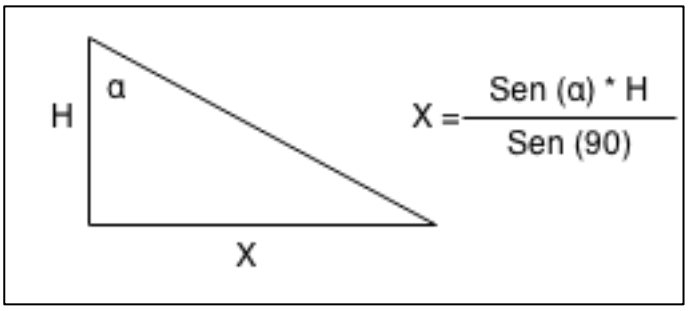


Ilustración 16 - Teorema del seno distancia

Así pues si el usuario estuviese apuntando a un objeto lejano cuyo ángulo con su dispositivo es de 75 grados (α) con una altura de 1,6 metros (H); la distancia a este objeto sería de unos 6 metros aproximadamente.

Para el cálculo de la altura de un objeto tenemos que calcular antes la distancia a él. Los cálculos se realizan igual que los descritos anteriormente pero añadimos un paso más. El esquema explicativo es el siguiente.

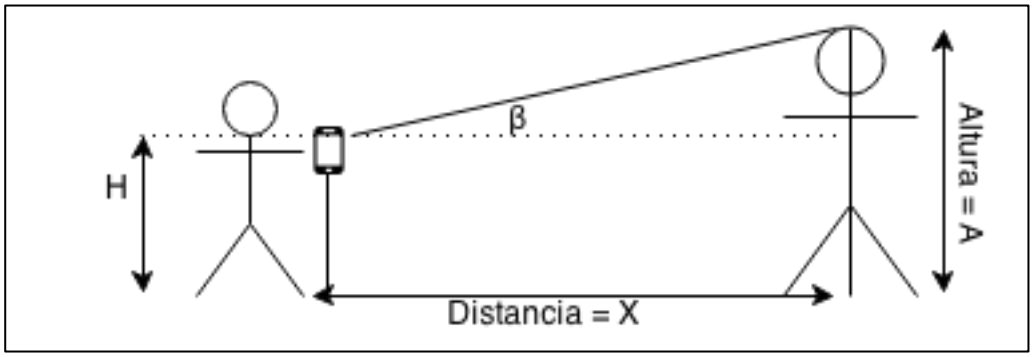


Ilustración 17 - Esquema cálculo altura

En este caso el dispositivo obtendrá el ángulo beta (β) con el teorema descrito antes. Se forma un triángulo en el que un cateto sigue siendo X y el otro A – H. Una vez calculado el cateto A-H del triángulo obtendremos la altura completa del objeto.

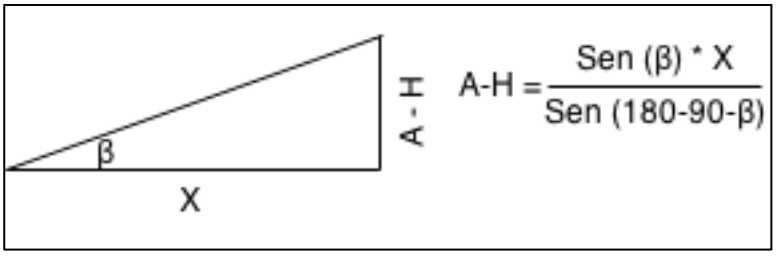


Ilustración 18 - Teorema seno altura

Todas estas operaciones trigonométricas son necesarias realizarlas cada vez que el usuario mueve el dispositivo para enfocar. Android ofrece herramientas que calculan los valores del giroscopio cada vez que alguno de ellos cambia. En la aplicación realizamos los cálculos cada cierto tiempo para no saturar de trabajo. Una vez realizada un cálculo completo todo lo visto anteriormente se muestra al usuario el valor final.

4.3 Metodología de desarrollo

El tipo de metodología elegido para el desarrollo del proyecto es modelo en espiral. Este tipo de modelo permite reducir riesgos e introducir mejoras y nuevos requerimientos durante el proyecto.

El modelo tiene una serie de actividades que se conforman en una espiral en la que se realizan iteraciones que representan un conjunto de actividades. El proyecto dividido en estas actividades se les aplica el siguiente proceso:

1. Análisis: En esta etapa se estudian detalladamente los requerimientos de cada objetivo, se identifican los riesgos y se determinan las posibles alternativas para solucionarlos. Durante esta etapa se establecen los detalles funcionales deseados.
2. Diseño: Con los datos de la etapa anterior, se diseña el sistema. Los diseños realizados son por ejemplo, las interfaces, o los datos necesarios para hacer funcionar la aplicación.
3. Implementación: En este tercer paso se realiza la programación del diseño. Se tiene que elegir un buen modelo de programación para que todo el código sea uniforme.
4. Pruebas: En este último paso es donde el proyecto es revisado. Las pruebas realizadas pueden ser diversas. Se puede realizar una prueba de un módulo completo y la posterior comparación frente a los requerimientos descritos anteriormente. A continuación se toma la decisión si se continúa con el siguiente ciclo del modelo en espiral.

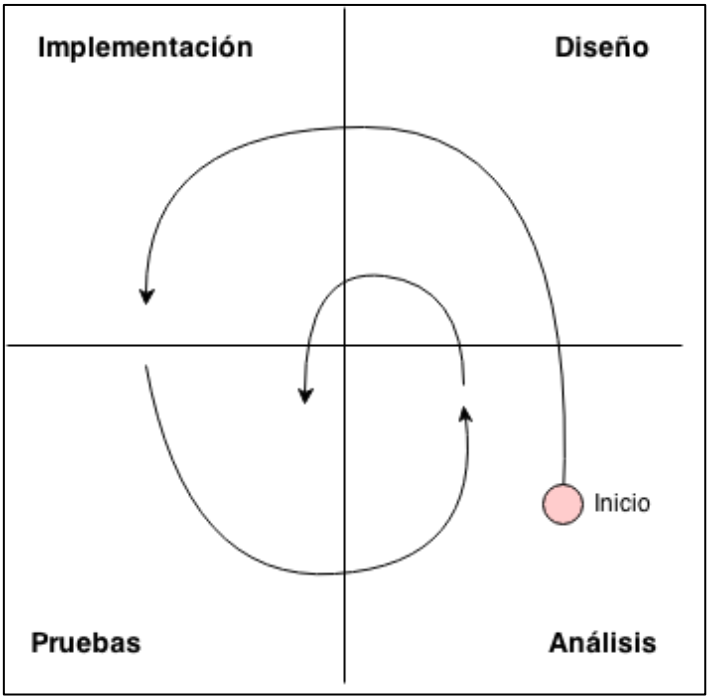


Ilustración 19 - Metodología en espiral

Con cada iteración de la espiral, se crean sucesivas versiones de la aplicación, cada vez más completas. Al final de todas las iteraciones, el sistema es uno completamente funcional

4.4 Casos de uso

Los casos de uso son una forma de ayudarnos a diseñar la aplicación. Con este método de diseño se simula las distintas interacciones entre el usuario y las distintas partes de la aplicación. Los casos de uso reúnen una serie de características:

- Es siempre iniciado por el usuario y nunca desde el interior de la aplicación.
- Desde el punto de vista del usuario, debe representar una acción completa.
- Debe completarse en un tiempo corto.
- Pueden participar varios usuarios.

Los casos de uso describen cosas que los usuarios quieren que el sistema haga. Un caso de uso debe ser una tarea completa desde el punto de vista del usuario, y debe corresponder a una tarea que se realiza en un tiempo relativamente breve. Si estas condiciones no se cumplen, es mejor definir varios casos de uso independientes.

Un caso de uso describe no solo una funcionalidad, sino también una interacción entre un usuario y la aplicación. La descripción que proporciona el caso de uso se refiere a lo que se espera que la interacción realice y no a como lo realiza. Es importante no solo considerar los casos de uso que describen la funcionalidad básica sino también considerar aquellos que describen las tareas de mantenimiento.

A continuación se han realizado los casos de uso del proyecto divididos en las cinco principales actividades que realiza la aplicación:

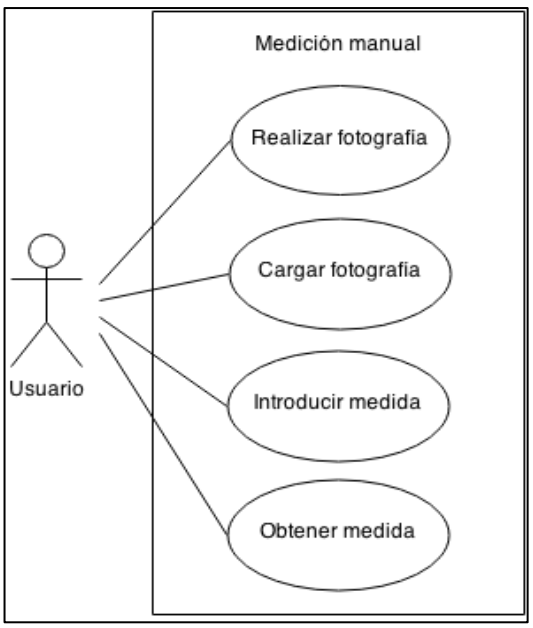


Ilustración 20 - Caso de uso medición manual

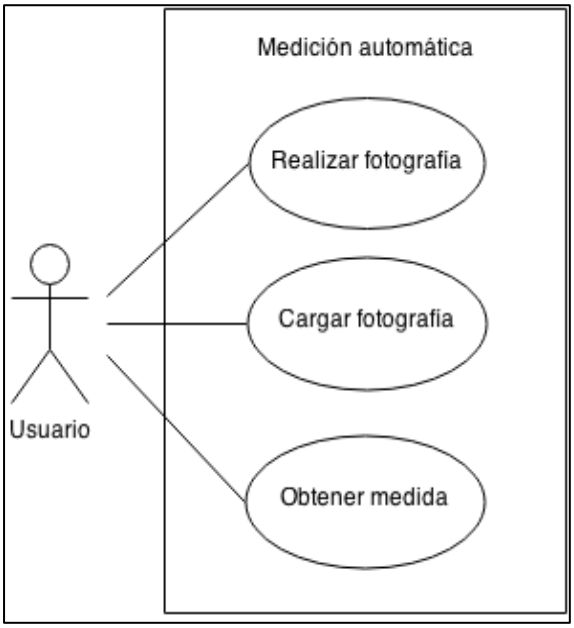


Ilustración 21 - Caso de uso medición automática

En los siguientes casos de uso hacemos referencia a acciones como “calibrar distancia” o “calibrar altura”, con estas acciones nos referimos al proceso que debe realizar el usuario para enfocar correctamente la cámara al objeto que se quiera medir.

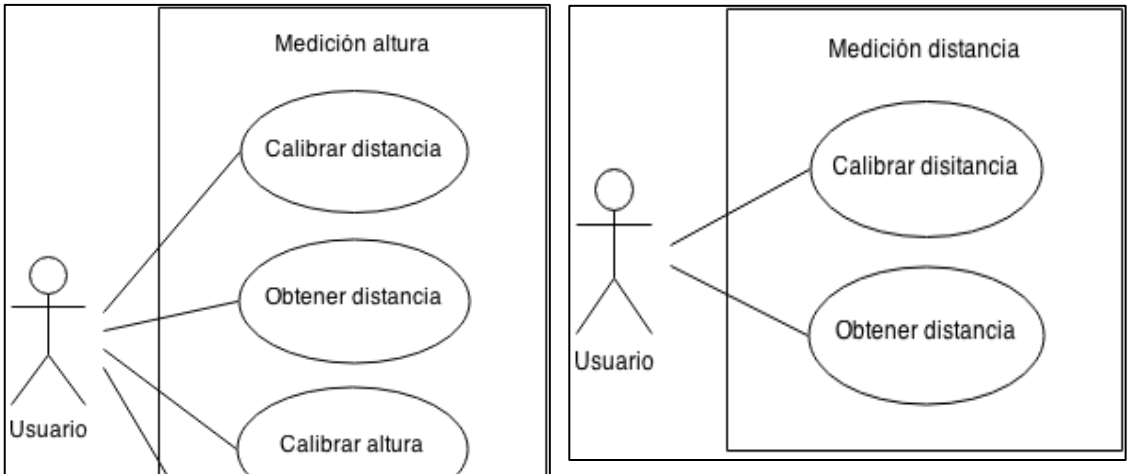


Ilustración 22 - Caso de uso medición distancia

Ilustración 23 - Caso de uso medición altura

En este último caso de uso “Enfocar objeto” se refiere a la acción que debe realizar el usuario para realizar la fotográfica con el objeto de referencia dentro de la plantilla dibujada por la tarea.

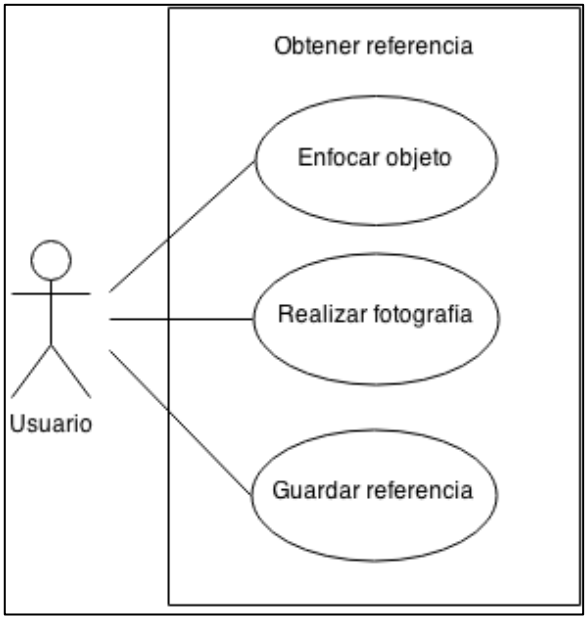


Ilustración 24 - Caso de uso obtener referencia

Como observamos, en ninguno de los cinco casos de uso expuestos se superan más de cuatro acciones que el usuario puede realizar. Esto es debido a que se ha querido que la aplicación sea simple de manejar y entender de acuerdo a uno de los objetivos preestablecidos en un principio.

4.5 Interfaz de usuario

En este apartado se aborda el diseño de la interfaz de la aplicación. Para definir las interfaces del software es necesario que se ajusten a las necesidades de los distintos usuarios teniendo en cuenta la usabilidad como pilar fundamental. La interfaz de usuario es la forma con la que el usuario puede comunicarse con la aplicación. Todas ellas deberían ser fáciles de entender y fáciles de utilizar.

Toda buena interfaz de usuario debe tener como mínimo estas cualidades:

- Claridad: sin manuales, interfaz limpia.
- Concisión: especificar todo brevemente.
- Familiaridad: reutilizar elementos preestablecidos en la sociedad.
- Sensibilidad: rapidez y buen feedback.
- Consistencia: extrapolar el manejo a diferentes áreas de la interfaz.
- Estética: las interfaces atractivas son más agradables.
- Eficiencia: debe ahorrarnos tiempo y esfuerzo.
- Errores: no castigar al usuario por sus errores.

Uno de los primeros pasos recomendables a realizar es crear un boceto en papel y lapiza que intente cubrir los distintos escenarios propuestos. Una vez tengamos claro cómo será el diseño, se procederá a realizar un prototipo para que un futuros posibles usuarios puedan decidir sobre la aplicación.

Como hemos comentado, en un primer paso se ha realizado el boceto a papel y lápiz de las principales interfaces de la futura aplicación. Una vez se está satisfecho con el resultado, se procede a la realización de los bocetos digitalmente o también llamadas maquetas digitales.

Existen múltiples herramientas que permiten realizar maquetas digitales de interfaces. Muchas de estas herramientas son específicas para este fin. Algunas de las más famosas son las herramientas MOSKitt Sketcher y Balsamiq Mockups. A pesar de poder acceder a este tipo de herramientas se ha escogido una herramienta online para la realización de todo tipo de esquemas e interfaces. La herramienta en cuestión se puede encontrar en la bibliografía de la memoria [7]. Con esta herramienta también se ha realizado todos los demás esquemas del proyecto.

Un punto muy importante a destacar en la realización de interfaces en Android es su diseño para la gran cantidad de dispositivos. Esta interfaz no cuenta con mucho diseño de interfaz pero se han seguido los pasos pertinentes para que se pueda ejecutar y visualizar adecuadamente en cualquier dispositivo Android.

Las maquetas nos permiten visualizar el diseño de la interfaz de una manera muy aproximada a la versión final. En las siguientes ilustraciones se muestran las que se han realizado en una primera instancia.

En la primera ilustración se muestra el menú de la aplicación. Se ha querido que todas las actividades realizadas por la aplicación se muestren en un único menú principal. También se puede observar que se ha dividido las actividades en diferentes apartados según su funcionalidad.

Ilustración 25 - Interfaz menú principal

Las siguientes interfaces realizan las tareas que componen nuestra aplicación. Un elemento común entre ellas es una barra en la parte inferior de la pantalla con dos botones a sus dos lados y un texto explicativo respectivo. Con esta barra se controlan las acciones básicas de cada una de las tareas.

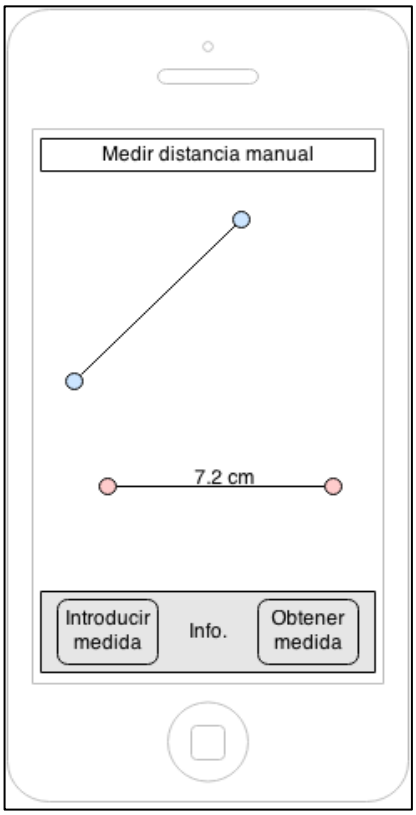


Ilustración 27 - Interfaz medir distancia manual

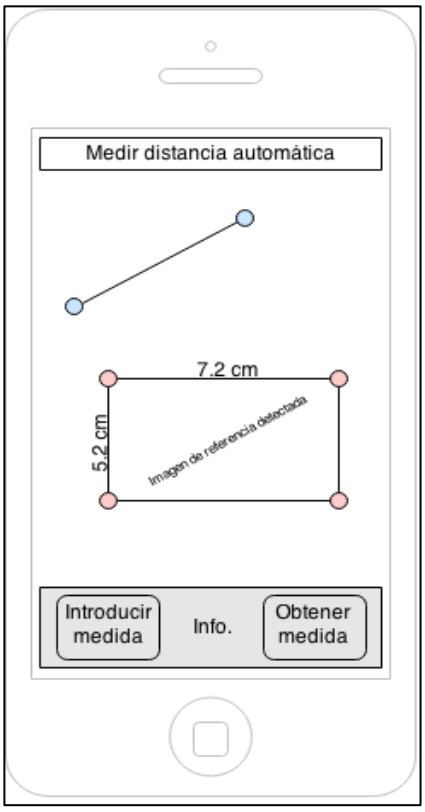


Ilustración 26 - Interfaz medir distancia automática

Como se observa en las dos imágenes anteriores tenemos dos tipos de medidas indicadas por los colores azul y rojo. La medida roja representa aquella que necesita un valor preestablecido para que el usuario pueda calcular las medidas que desee. En la medición manual el usuario introducirá la medida manualmente. En la imagen de la medición automática observamos que existe un rectángulo, este rectángulo se dibujará automáticamente a partir de la imagen de referencia (en nuestro caso una tarjeta, por eso la forma de rectángulo).

Las siguientes imágenes son parecidas ya que describen la interfaz de la medición de la distancia y la altura, sin embargo; existen algunas diferencias que explicarían a continuación.

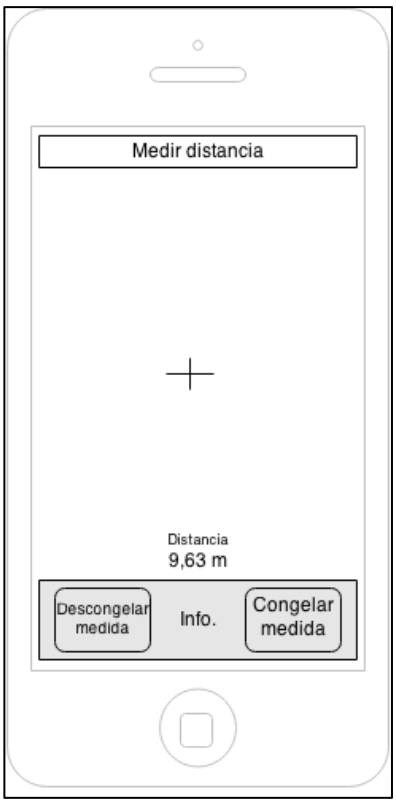


Ilustración 28 - Interfaz medir distancia

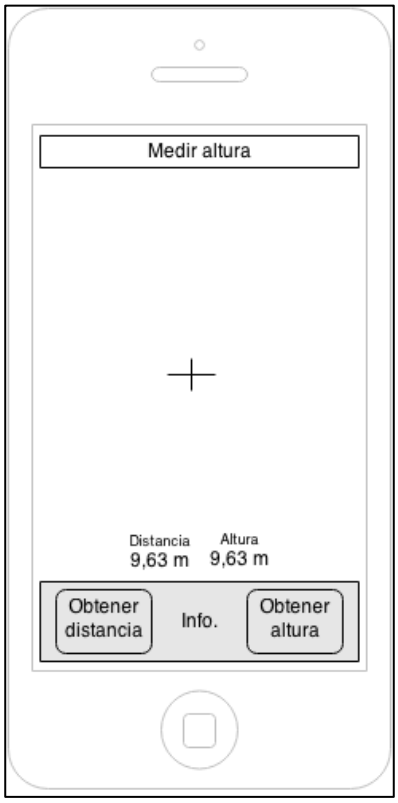


Ilustración 29 - Interfaz medir altura

Observamos en los bocetos de la interfaz como son prácticamente iguales. La diferencia entre ellos es que la interfaz de la medición de la altura incorpora también el valor de la distancia, ya que es necesario para el cálculo de la altura. Los botones de la barra inferior también cambian para adecuarse a las acciones necesarias para obtener las medidas.

La siguiente interfaz es la última y pertenece a la obtención de la imagen de referencia. En esta interfaz hacemos uso también de la previsualizacion de la cámara. Encima de e ella se dibujara una plantilla para ayudar al usuario. La plantilla está calculada a partir de las medidas comunes de las tarjetas que son 8,5 centímetros de ancho y 5,4 centímetros de alto. Una vez realizada la fotografía correctamente, el usuario puede guardar la imagen, que se recortara siguiendo la plantilla, pulsando el botón de guardar imagen.

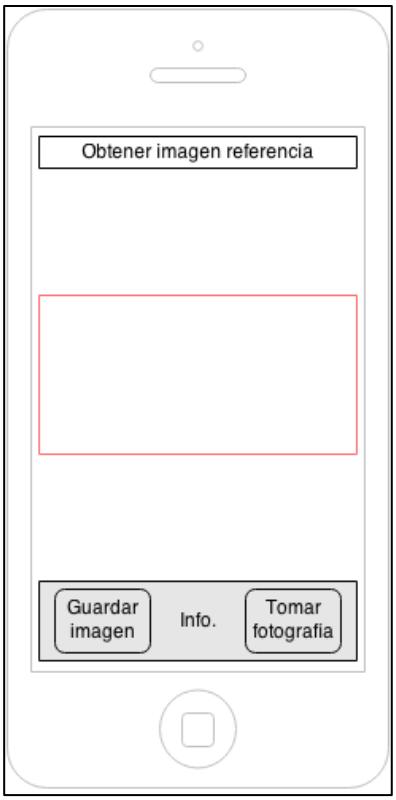


Ilustración 30 - Interfaz obtener imagen referencia

Tras el diseño de las interfaces, hay que realizar siempre un análisis de ellas. Existen multitud de métodos que ayudan a su evaluación, uno de ellos es la realización de tareas concretas por parte de personas externas al desarrollo de la aplicación. Hay que observar los pasos que realizan, fijarnos que hace y donde pierde más tiempo para luego reorganizar los controles o rediseñar la interfaz teniendo en cuenta los problemas de usabilidad que se encuentren.

En el proyecto se ha enseñado la aplicación a diferentes personas y hemos obtenidos opiniones satisfactorias. Las deficiencias encontradas son fácilmente corregibles modificando algún aspecto. Las interfaces mostradas pertenecen a la versión final del proyecto.

5. Resultados

En el punto de resultados se ha querido presentar el funcionamiento de la aplicación funcionando en un dispositivo móvil. En este apartado veremos las consecuencias finales del desarrollo de la aplicación y explicaremos si son buenos o no.

5.1 Ejemplos

La finalidad de este apartado es mostrar la aplicación funcionando en un dispositivo. Se mostraran algunas capturas del dispositivo de las distintas partes de la aplicación, así observaremos el resultado de la ejecución. Debido a que la entrada de la aplicación la proporciona el usuario, las imágenes mostraran datos ya introducidos para estudiar las distintas posibilidades que ofrece el programa.

En la primera captura observaremos el menú principal. Se aprecia la distribución por tareas para que el usuario entienda que se realiza en cada actividad.

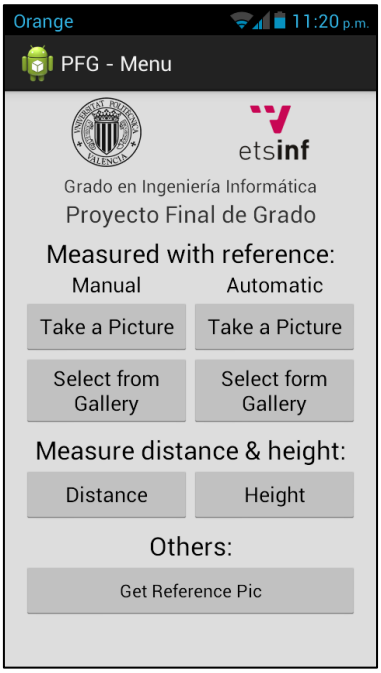


Ilustración 31 - Ejemplo menú

La aplicación está pensada para que se ejecute en modo retrato, es decir de modo vertical. No se puede girar el menú de modo que se pueda usar de modo vertical. A pesar de que el menú está diseñado para que se vea también de forma horizontal y que se adapte a esta forma, se ha tomado esta decisión para no confundir al usuario y evitar problemas en la captura de imágenes.

Las siguientes capturas muestran el funcionamiento de la medición manual. Se muestra una método a pesar de haber dos de llegar a la medición manual ya que la única diferencia de los dos métodos de medida manual son la forma que tiene en cargar las imágenes, un método lo hará a través de la galería y otro método a través de la toma de imágenes.

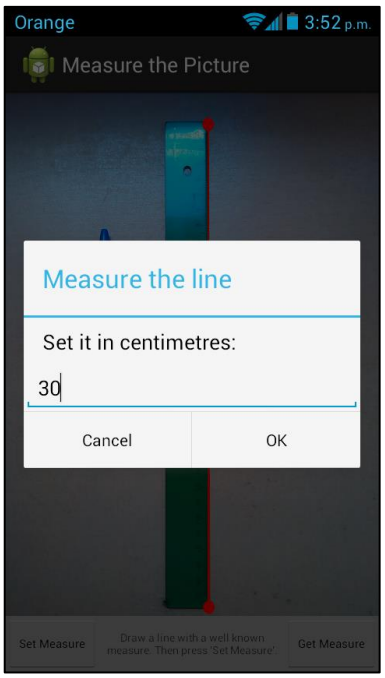


Ilustración 33 - Ejemplo medir manual introducir medida

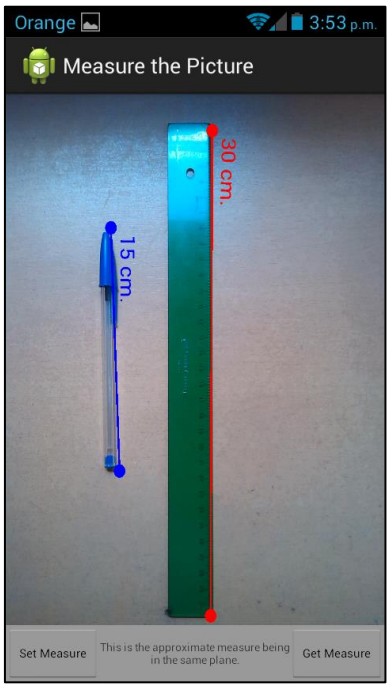


Ilustración 32 - Ejemplo medir manual resultado

Como observamos en la imagen se realiza un ejemplo simple del funcionamiento. La medida en rojo ha sido introducida por el usuario y la medida azul ha sido obtenida automáticamente contando con la anterior, utilizando el botón “Get Measure”.

En las dos imágenes siguientes se muestra el funcionamiento de la medición automática, la imagen de la izquierda muestra el estado de la actividad nada más ejecutarla. La aplicación automáticamente carga la imagen de referencia y realiza la búsqueda para después dibujar el contorno de la tarjeta encontrada y sus medidas. La imagen de la derecha es el resultado de dibujar una medida (línea azul) y obtener su medida automáticamente pulsando sobre “Get Measure”.

Ilustración 35 - Ejemplo medir automático detección

Ilustración 34 - Ejemplo medir automático resultado

Ahora se observa las actividades de medición de distancia y altura de un objeto. Dividiremos las capturas en dos tandas. La primera de ellas corresponde a la medición de distancia.

Ilustración 36 - Ejemplo medir distancia

En esta primera captura se representa la actividad de medida de distancia. Se ha realizado la prueba con un metro de dos metros de longitud y se ha enfocado al final de él para obtener una referencia aproximada.

En las siguientes dos actividades observamos la actividad de medida de altura que es una más elaborada ya que tiene incorporada la funcionalidad de medir distancias. En las dos actividades observaremos una cruz blanca en mitad de la pantalla que nos ayuda a enfocar correctamente el objeto a medir.

Ilustración 38 - Ejemplo medir altura base

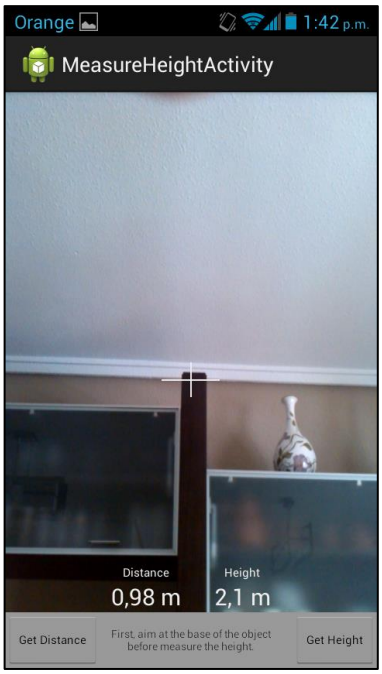


Ilustración 37 - Ejemplo medir altura zona alta

Finalmente, se muestra una captura de la actividad para obtener la imagen de referencia de la aplicación. Como ya hemos comentado durante la memoria, la imagen utilizada para la elaboración del proyecto ha sido una tarjeta común, ya que tienen unas medidas determinadas.

Ilustración 39 - Ejemplo obtener referencia

En la imagen observamos la plantilla cuadrada de color blanco con las mismas proporciones que las tarjetas comunes. El usuario debe realizar la fotografía con la tarjeta de referencia dentro de la plantilla dibujada.

Todas estas capturas de la aplicación funcionando en un dispositivo móvil nos ayudan a comprender mejor el funcionamiento de la aplicación. Además se comprueba que la aplicación funciona correctamente en un sistema real.

5.2 Evaluación del sistema

Una vez completado el proyecto queda por realizar una parte fundamental. Como se ha comentado en esta memoria, la metodología de trabajo llevada a cabo ha sido un modelo en espiral, por lo que muchos fallos fundamentales se han ido corrigiendo durante la fase correspondiente. En esta metodología se realizan pruebas cada vez que se implementa un bloque de tareas porque de alguna forma se evalúa el sistema repetidamente hasta conseguir una versión funcional estable.

A pesar de realizar esta metodología siempre existe un límite, una meta en la que se tiene que finalizar el proyecto. En este apartado se exponen algunas opiniones sobre la evaluación final del proyecto. También se habla sobre los objetivos que se han llegado a cumplir y aquellos que no se han cumplido totalmente o simplemente no se han conseguido.

De la aplicación final podemos decir que tiene una estructura muy bien definida. Una buena estructuración de un sistema ahorra muchas incomodidades al usuario que la usa. La elección de un menú principal que engloba toda la funcionalidad de la aplicación puede no ser la elección más acertada de todas, pero tampoco resulta desconcertante como pueden haber sido otros métodos de mostrar la información.

Siguiendo con una evaluación de la interfaz, se observa que en todas las actividades la interfaz es muy sencilla, el usuario puede ver perfectamente cuales son las acciones a realizar en cada una de ellas. Además de que la interfaz no es muy complicada, se ha conseguido introducir textos de ayuda en cada una de las actividades por lo que resulta ser un extra de comunicación con el usuario. También hay que decir que en una aplicación donde la funcionalidad principal es el uso de la cámara, no se puede abusar de interfaz ya que estropearía la interacción con el usuario.

La elección de introducir dos métodos de carga de imágenes en la actividad para que el usuario pueda medirlas es acertada. Si se hubiera optado por cargar las imágenes solo desde la galería, el usuario perdería mucho tiempo. Así pues, la elección de poder tomar imágenes que se van a poder medir posteriormente es acertada.

5.2.1 Cumplir objetivos

Si observamos los objetivos principales del proyecto redactados en el apartado de objetivos observamos que se han cumplido tres de los cuatro objetivos propuestos. La causa de que el último objetivo, medición por estereoscopia, no se haya resultado es gran parte por el tiempo. A pesar de ello, lo poco investigado al inicio del proyecto sobre esta posibilidad nos indica que puede resultar muy difícil realizar esta técnica, ya que en la mayoría de ocasiones se necesitan dos cámaras para poder utilizar esta técnica.

A continuación observaremos los objetivos secundarios, redactados como extensión a los principales. Empezaremos por el objetivo esencial de la aplicación, la forma en que se dibujan las líneas de las que el usuario desea obtener las medidas. El estilo de las líneas y números es simple, pero no necesita más. Si atendemos a respuesta del dedo con la pantalla, notaremos que no es del todo precisa debido a que se realiza un cálculo en base a las dimensiones del dispositivo para que la respuesta sea la misma en la mayor parte de aparatos. A pesar de todo se ha cumplido con creces la manera de medir las distancias.

La medición de distancias y alturas son algo imprecisas. Debido a la sensibilidad del sensor, es muy difícil obtener medidas precias, más aun cuando las medidas que queremos obtener varían en función de centímetros. Por otro lado, la interfaz para la medición de estas dos actividades es perfecta ya que muestra claramente y en tiempo real la medida que el usuario quiere obtener. Por lo tanto a pesar del inconveniente se cumple el objetivo propuesto.

En general los objetivos secundarios se han cumplido con más o menos acierto. Está claro que es posible realizar varias mejoras para adaptarse a los objetivos pero no hay que dejar de tener en mente que la elaboración de la aplicación final es solo una meta para todo el aprendizaje que constituye. Si nos fijamos en el objetivo de obtener conocimientos durante la elaboración del proyecto se puede decir que se ha cumplido con creces.

6. Conclusiones

6.1 Desarrollo del proyecto

La elaboración del proyecto ha constituido un objetivo principal en los últimos meses. Se ha aprendido mucho sobre Android y también se ha realizado un primer acercamiento sobre la inmensa cantidad de herramientas que ofrece la librería de OpenCV para el tratamiento de imágenes.

El estudio previo a la implementación de las soluciones ha sido arduo, ya que una gran parte del tiempo es dedicado al estudio de la API. También han sido un gran apoyo los problemas solucionados por otras personas que son expuestos en diversas páginas de internet.

6.1.1 Duración

La duración del proyecto ha sido de tres meses y medio aproximadamente. En este intervalo están también incluidos los primeros instantes en los que se estaban realizando otras tareas educativas. También se incluye el tiempo dedicado a la elaboración de esta memoria en la que se estuvo finalizando el proyecto al mismo tiempo que se empezaba a redactar.

6.1.2 Problemas encontrados

Los problemas encontrados durante la elaboración del proyecto son diversos. Uno de los problemas que tiene desarrollar con Android es la utilización de elementos que en el mismo momento del desarrollo están desactualizados, por lo que no es recomendable su utilización. En ese momento se tiene que encontrar una solución que realice la misma actividad con elementos actualizados.

Otro problema que además es muy recurrente, es la infinidad de maneras que existe en Android de realizar una misma actividad. Puede no parecer un problema pero si desarrollas una forma de realizar una tarea que no funciona con el resto de tu aplicación pierdes el tiempo en buscar información para corregir esos problemas. En el caso del proyecto existen varias formas de previsualizar una cámara, una de ellas es la previsualización mediante librerías de OpenCV que obliga al desarrollador a crear una interfaz personalizada. En este caso se usó la previsualización de la cámara que ofrece Android ya que es mucho más amigable y puede usarse con elementos propios de Android.

Otro gran problema encontrado tiene que ver con uno de los elementos más importantes del proyecto, la detección automática del objeto de referencia en una imagen. La investigación e implementación de una técnica que realizara esta acción ha llevado varias semanas y aun así no es para nada perfecta. Debido a la calidad de algunas cámaras de los dispositivos Android, la diferencia entre una imagen y el objeto de referencia es tanta que en ocasiones la técnica no reconoce absolutamente nada de parecido entre las dos imágenes, por lo que en estos casos la detección fracasa. Se han implementado diferentes mejoras propias para la mejora de la técnica como igualar el tamaño de las imágenes.

6.2 Conclusiones finales

Con la elaboración del proyecto se han realizado una serie de tareas y métodos que ayudan a entender mejor el proceso adecuado que hay que realizar a la hora de la elaboración de un proyecto.

En este caso se ha observado que debe existir un proceso largo de documentación antes del inicio del desarrollo de software. Hay que convertirse en un experto en el área de la aplicación del proyecto, y esto sólo se consigue estudiando el trabajo de otros expertos. Aplicando todas las técnicas estudiadas se puede llegar a elaborar una idea clara de la aplicación futura.

El área de desarrollo de Android es un terreno difícil ya que se encuentra en constante cambio. Por este motivo es necesario que el desarrollador invierta una cantidad de tiempo considerable en el estudio y seguimiento de esta tecnología. Aunque el proyecto tratase de medición de distancias con un dispositivo Android, no hay que olvidarse de que para la elaboración de esta meta se ha invertido una gran cantidad de tiempo en el estudio de la plataforma Android. Todo este tiempo invertido no es tiempo desperdiciado ya que se aprende una tecnología que actualmente está muy solicitada.

Durante el proyecto también se ha llevado a cabo una iniciación a las librerías de tratamiento de imágenes de OpenCV. El área de visión computacional puede ser muy compleja; el tratar estas tecnologías junto a la tecnología Android suponía un gran reto. Aun así se ha conseguido hacer un uso efectivo de las librerías de OpenCV para conseguir hacer algo que, a pesar de que no es muy complicado, resulta funcional y atractivo.

6.3 Trabajos futuros

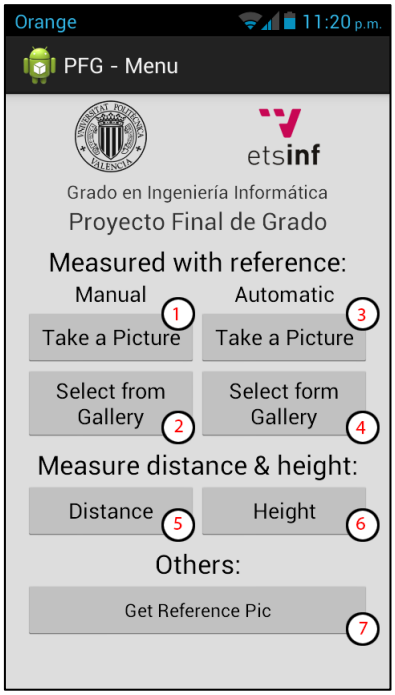
En este apartado se pretende proponer algunas mejoras que puedan complementar a este proyecto. Redactaremos algunos aspectos a perfeccionar del proyecto que por tiempo o conocimiento no se ha podido implementar.

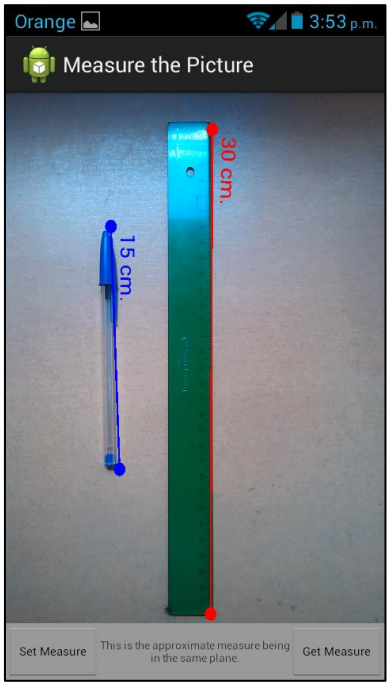
- Mejora de la detección automática. Al realizar desde cero esta aproximación y con conocimiento nulo sobre el tema, el resultado es mejorable en distintos aspectos.
- Añadir más funcionalidad a toda la interfaz de dibujo, como la posibilidad de cambiar el color de las líneas, modificar el tipo de flecha, etc.
- Introducir la medida por referencia en la barra de acción de las actividades automáticas.
- Introducir otro tipo de objetos de referencia comunes como una moneda o un billete. De la misma manera habría que cambiar las plantillas.
- Permitir al usuario guardar las imágenes con las medidas en el dispositivo.
- Mejora en general de código fuente. Elaborar un código más profesional.
- Menú más acorde a una aplicación moderna.
- Corregir la perspectiva para poder medir correctamente los objetos en caso de que el usuario realice una fotografía con perspectiva.

A. Manual de usuario

En este apéndice se explica más detalladamente el uso de la aplicación. Tras la instalación de la aplicación mediante el archivo .apk con el administrador de archivos del dispositivo, aparecerá un icono en nuestro menú.

Tras pulsar sobre el icono aparecerá el menú de inicio de la aplicación, que se divide en siete botones con los que se acceden a las distintas partes del programa.

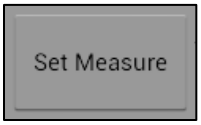




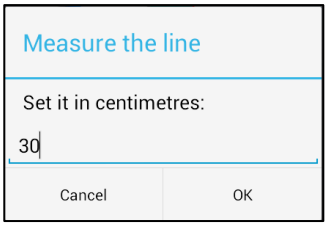
Si pulsamos sobre el número **1** o el número **2**, se nos direccionara a la medida manual, tras tomar una imagen o cargarla desde la galería respectivamente.

Tras este paso, el usuario debe dibujar una medida pulsando sobre la pantalla y arrastrando. La línea resultante será de color rojo.

Una vez dibujada la línea se debe introducir la medida pulsando sobre el botón "Set Measure".



Al pulsar el botón se abrirá un dialogo para introducir la medida en centímetros. Tras introducir el número aceptaremos pulsando en "OK".



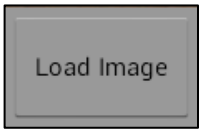
Aparecerá ahora la medida dibujada anteriormente con el número introducido pintado a lo largo de la línea. Ahora el usuario puede dibujar otra línea para obtener la medida que desee. Pulsando y arrastrando se dibujará ahora una línea azul.

Tras dibujar la línea azul, el usuario pulsara el botón “Get Measure” para obtener la medida de la línea. Al instante aparecerá la medida a lo largo de la línea.

Volviendo al menú, si pulsamos sobre los números **3** o **4**, nos pedirá que realicemos una fotografía o que carguemos una de la galería respectivamente. Es importante que sean imágenes con el objeto de referencia en ellas. Estas dos actividades pertenecen a la medida automática.

Si tenemos una imagen de referencia guardada, la actividad automáticamente detectara el objeto y lo dibujara con sus respectivas medidas.

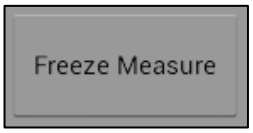
En este momento podremos realizar dos acciones. Si pulsamos sobre el botón “Load image”. Podemos cargar otra imagen desde la galería o realizar una fotografía (según en la actividad que estemos).



A partir de ahora el usuario podrá dibujar una medida con el dedo. La línea dibujada será de color azul. Para poder obtener su medida el usuario debe pulsar sobre el botón “Get Measure”, automáticamente se dibujará la medida a lo largo de la línea.

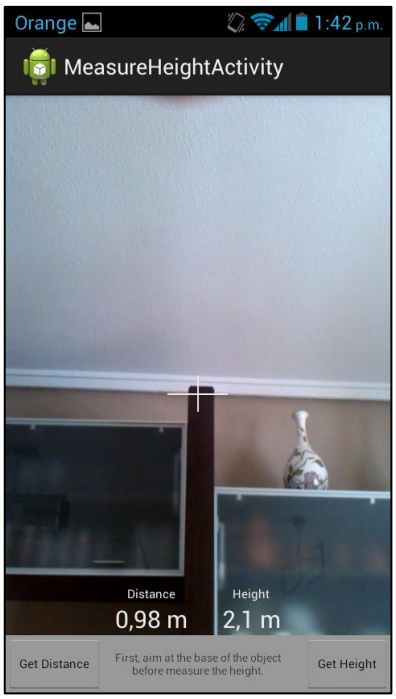
Si en el menú seleccionamos la opción número **5**, se iniciará la medición de la distancia. En la pantalla aparecerá una previsualización de la cámara con una cruceta en medio y una medida en metros que va cambiando conforme movemos el dispositivo.

Sólo tendremos que enfocar con la cruceta a la base de un objeto para obtener al momento la distancia a él.

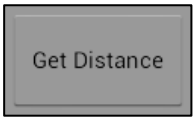


Para observar la medida adecuadamente pulsaremos sobre “Freeze Measure”. La imagen y la medida congelan al instante.

Si queremos volver a realizar una medida pulsaremos sobre el botón “Unfreeze Measure”.



Con la opción número **6** del menú de inicio iniciaremos la medición de la altura de un objeto. De la misma forma que la anterior actividad, tenemos que calcular la distancia al objeto del que queremos obtener la altura. En la pantalla también tenemos una previsualización de la cámara con las medidas de distancia y altura que se actualizan automáticamente.

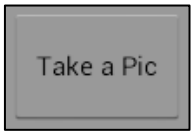


Apuntaremos a la base y pulsaremos sobre "Get Distance". La medida de distancia se pausará con la medida que teníamos al pulsar el botón.

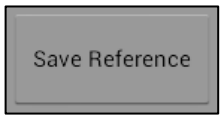
Ahora enfocaremos con la cruceta a la zona más alta del objeto y pulsaremos sobre "Get Height". La medida se pausará y obtendremos las dos medidas del objeto medido en la pantalla del dispositivo.

La última actividad tiene el número 7. Con esta tarea obtendremos la imagen de referencia necesaria para las actividades 3 y 4.

Al iniciar la actividad observaremos la previsualización de la cámara y un cuadrado dibujado por encima. El usuario debe usar este cuadrado como plantilla para ayudarse a realizar una fotografía del objeto de referencia. Recordemos que en el proyecto se ha usado una tarjeta común.



Una vez la tarjeta ocupe la totalidad de la plantilla se debe pulsar el botón "Take a Pic" para realizar la fotografía. En ese momento la previsualización se congelará y observaremos si la tarjeta ocupa la plantilla. También podemos tocar la pantalla para enfocar la cámara y así obtener una imagen de una mayor calidad.



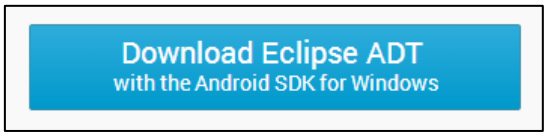
Cuando tengamos la imagen congelada y estemos contentos con el resultado, podemos pulsar el botón "Save Reference" para guardar la imagen en el dispositivo y que posteriormente, las actividades 3 y 4 puedan cargarla.

Con este pequeño tutorial de usuario se ha pasado por todas las actividades de la aplicación y como usarlas correctamente.

B. Instalación y configuración

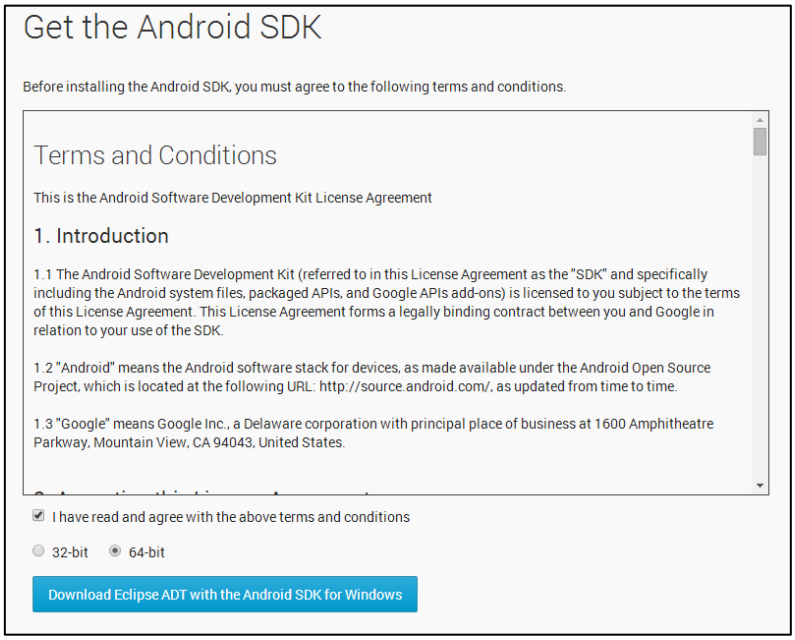
En la memoria se ha nombrado como tecnología usada el framework Eclipse ADT. En este apartado se mostrará cómo ha sido la instalación y configuración con OpenCV.

En primer lugar nos dirigiremos a la página de desarrollo de Android para descargarnos el eclipse ADT [1]. Pulsamos sobre el botón que dice “Download Eclipse ADT”.

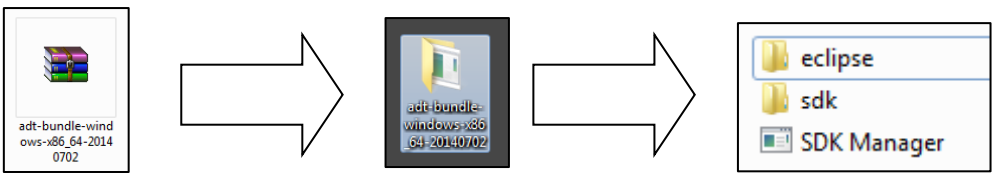


Observamos que junto a esta descarga también se incluye el SDK de Android para Windows, por lo que no tendremos que instalarlo posteriormente.

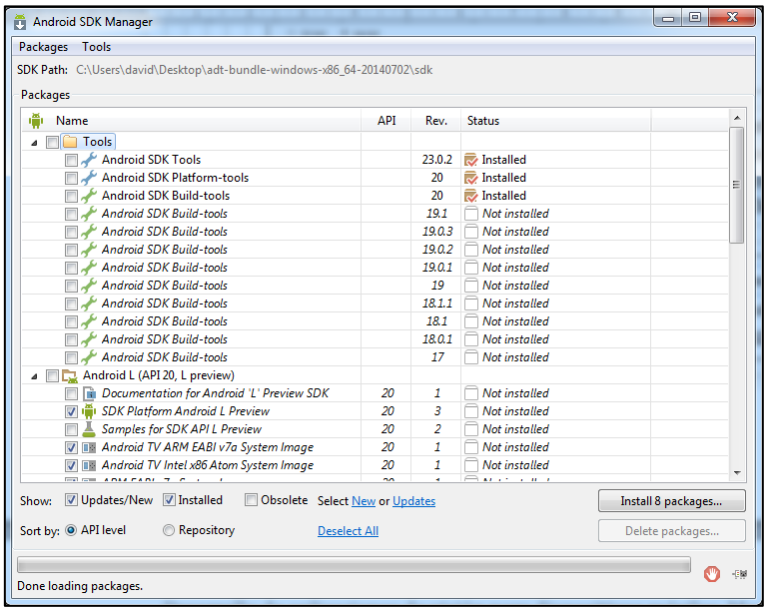
Se nos redigirá a una nueva pantalla en la que se nos pide aceptar los términos y elegir el tipo de sistema que tenemos, en nuestro caso es de 64-bit. Una vez marcadas estas dos opciones pulsaremos sobre el botón de descarga.



A continuación se descargará al ordenador el archivo que contiene todo lo que necesitamos. Dentro del archivo habrá una carpeta que tenemos que descomprimir en cualquier lugar de nuestro ordenador. Dentro de ella habrá tres elementos: una carpeta llamada “eclipse”, otra llamada “sdk” y un ejecutable llamado “SDK Manager”.

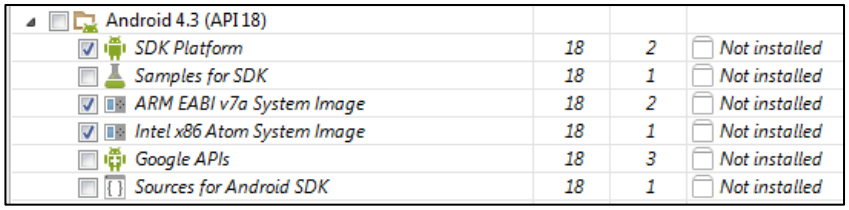


El siguiente paso será ejecutar la aplicación "SDK Manager". Con esta aplicación podemos manejar todas las versiones de Android que queremos que se instalen en nuestro ordenador para después poder desarrollar con ellas. Si queremos tener una versión de Android específica y no la hemos instalado, no podremos desarrollar para esa versión. Al ejecutarlo se abrirá una ventana con una lista de versiones de Android:

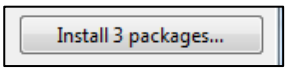


Es recomendable instalar alguna versión adicional además de las que están instaladas por defecto, ya que estas suelen ser la última versión de Android y no son del todo funcionales.

Para instalar una versión determinada desplegaremos el menú de una de las versiones y marcaremos para la instalación aquello que queramos instalar de esa versión. Es obligatorio marcar mínimo el SDK de la versión correspondiente así como uno de los dos controladores (o los dos) del procesador del dispositivo (del tipo ARM o Intel). En este ejemplo se procede a instalar solo los archivos necesarios de la versión 4.3 de Android.

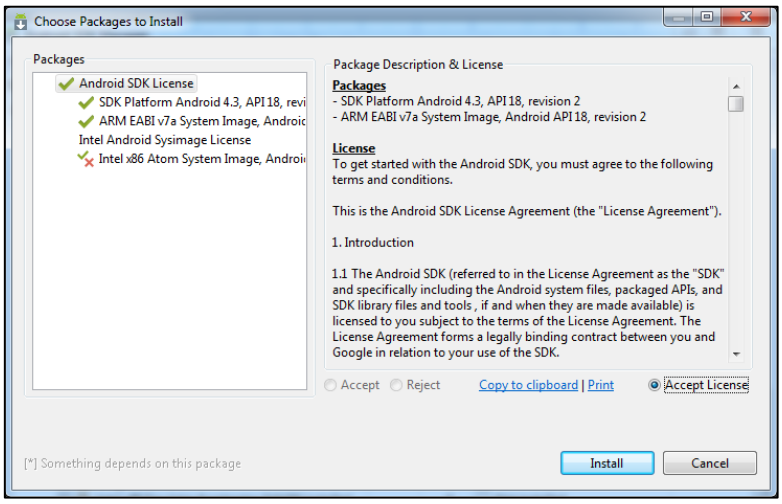


Una vez marcadas las opciones deseadas las instalaremos pulsando el botón de instalación situado en el menú inferior.

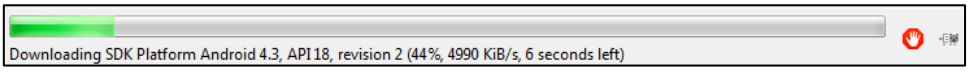


También es recomendable instalar los drivers USB de Google si queremos depurar en algún dispositivo móvil de Google. Estos drivers están situados en el menú "Extras".

Al pulsar sobre el botón de instalar se abrirá una ventana para aceptar el acuerdo de licencia del SDK de Android. Marcamos la opción “Accept License” y pulsamos “Install”.



El sistema procederá a la instalación de los paquetes seleccionados.



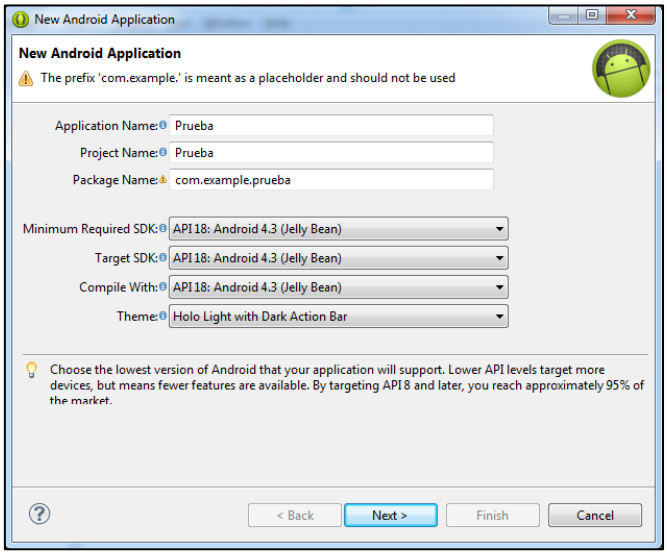
Una vez instalados, se nos mostrará el mensaje “Done loading packages.” Por lo que ya podremos cerrar la aplicación.

A continuación accederemos a la carpeta eclipse del archivo descomprimido anteriormente y ejecutaremos el programa eclipse.

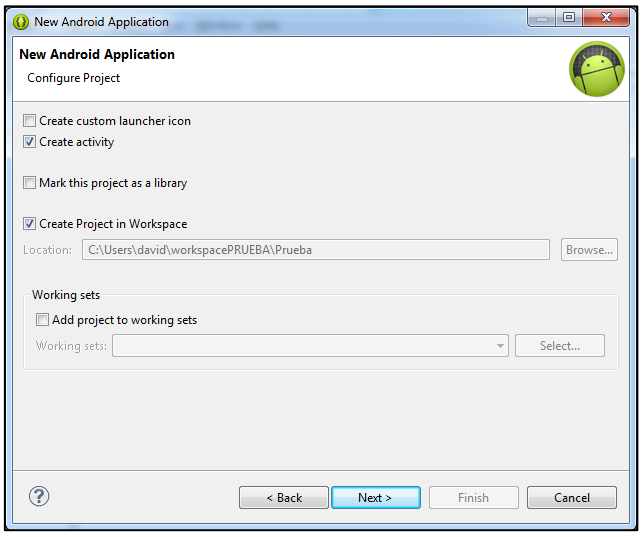
Se nos preguntará el nombre y la ubicación del workspace donde queremos trabajar. Le indicamos donde y que nombre queremos ponerle y pulsamos “OK”. Una vez iniciado procederemos a crear un proyecto Android. Para ello nos dirigimos a:

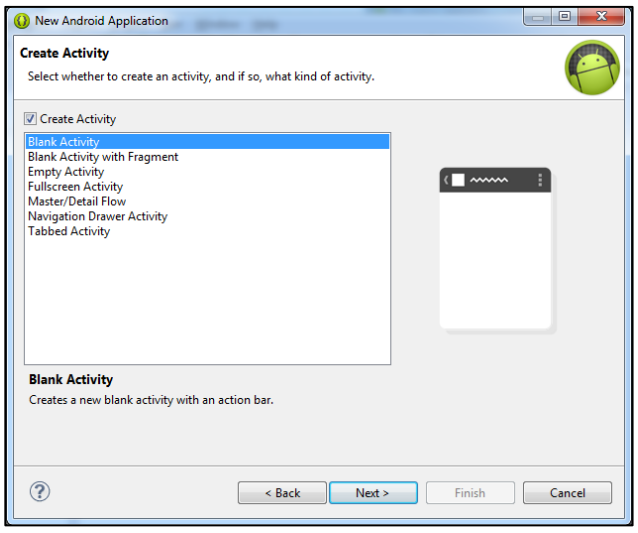
File > New > Android Application Project

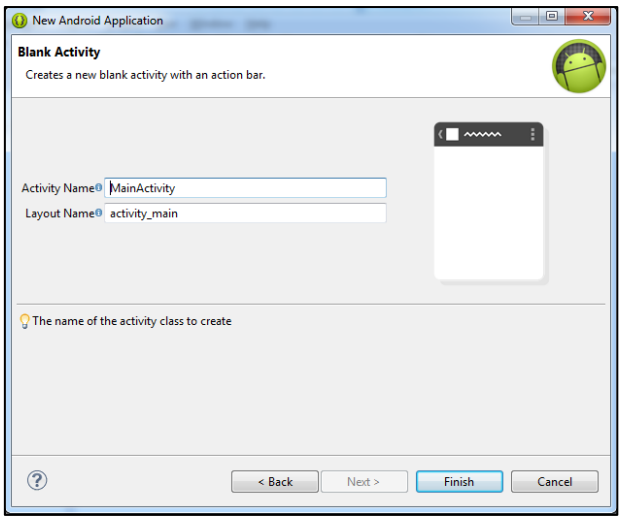
En la ventana actual seleccionaremos un nombre para nuestra aplicación.



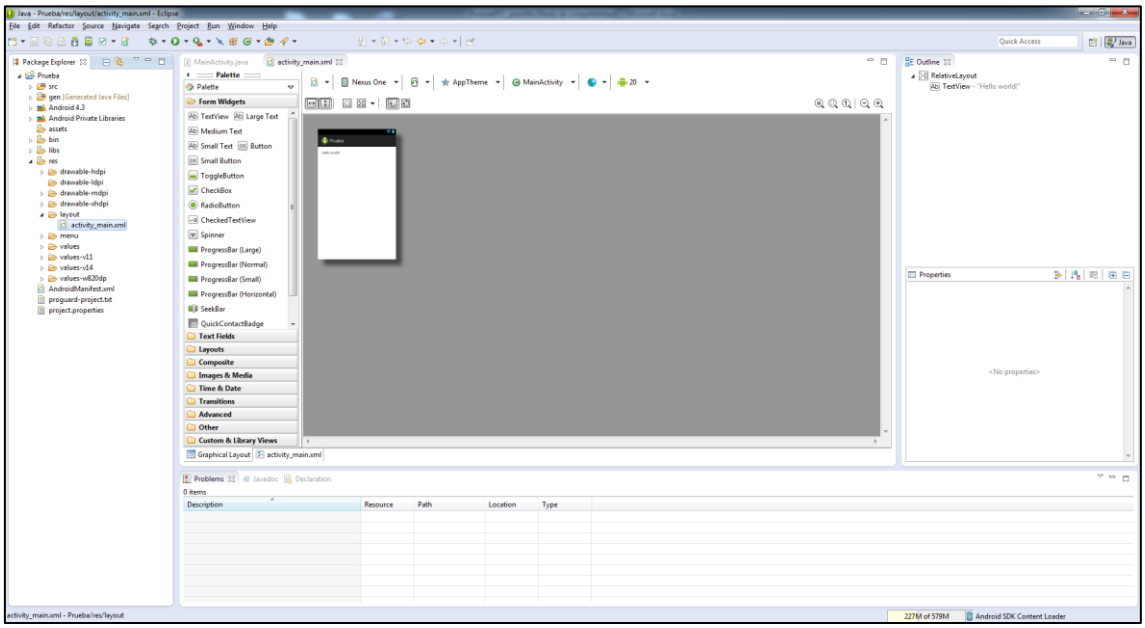
En “Minimun Required SDK” seleccionaremos la versión de Android más baja que tenemos instalada. Realizando esto nos aseguramos que nuestra aplicación se pueda utilizar en los dispositivos con versión de Android más viejos. Hay que recordar que para seleccionar una versión tenemos que tenerla instalada a través del SDK Manager. Pulsamos sobre “Next” para acceder a los siguientes pasos y finalmente sobre “Finish”.







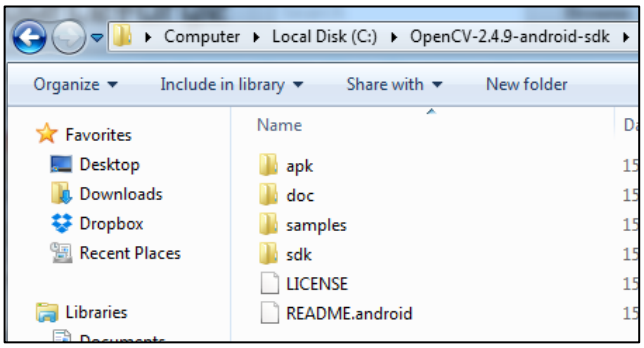
Tendremos ahora la pantalla de eclipse completamente funcional para empezar a trabajar.



Si todo ha ido bien deberemos tener enlazado el SDK de Android. Si no es así nos dirigiremos a Window > Preferences y en el apartado de Android demos verificar que el atributo SDK Location tiene la dirección del SDK de nuestra carpeta descomprimida.

A continuación añadiremos OpenCV a este proyecto Android. Iremos a la página web correspondiente para descargarlo [8]. Seleccionamos “OpenCV for Android”.

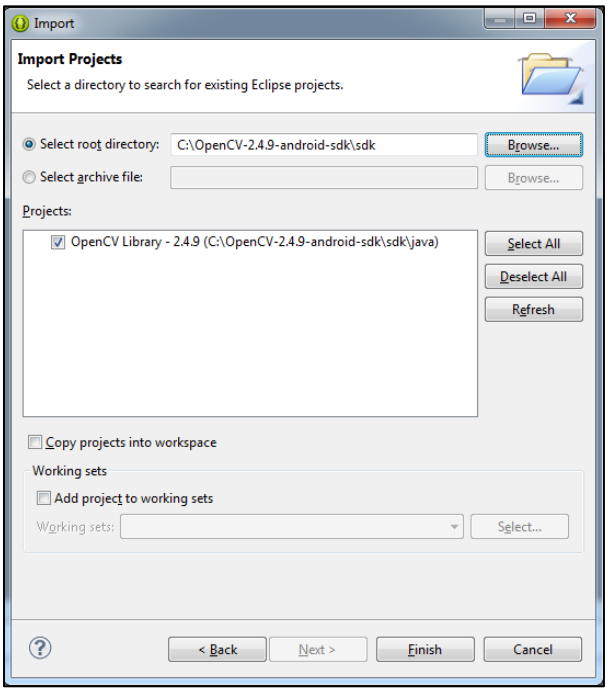
El archivo correspondiente se descargará y lo descomprimiremos en la carpeta que queramos. Para este ejemplo se ha situado directamente en la ubicación C:



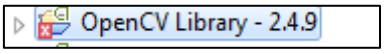
En eclipse tenemos que importar el sdk de OpenCV. Para ello usaremos el menú:

File > Import > General >Existing project in your workspace.

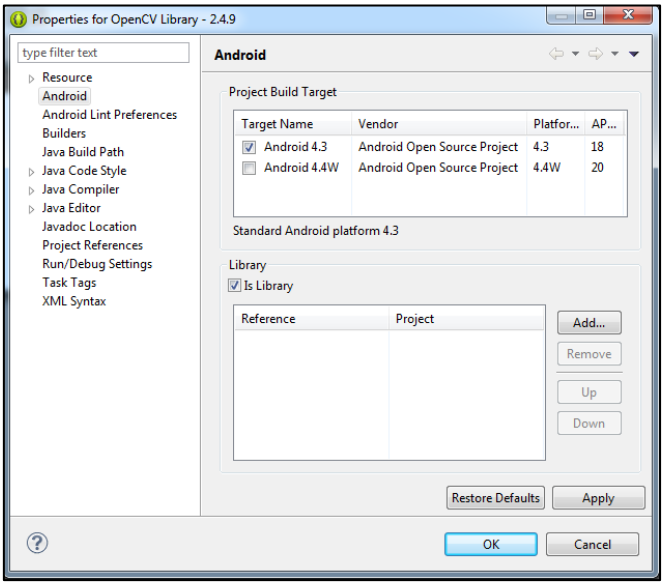
En la ventana que se iniciará, seleccionaremos la dirección del sdk de OpenCV. En la imagen siguiente se observa la ruta completa del archivo que debemos seleccionar.



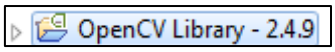
Una vez añadido pulsamos sobre “Finish”. Se añadirá al explorador de Eclipse la librería OpenCV. Es posible que muestre un error con un símbolo de una “x” roja esto quiere decir que le falta el enlace a las librerías de Android.



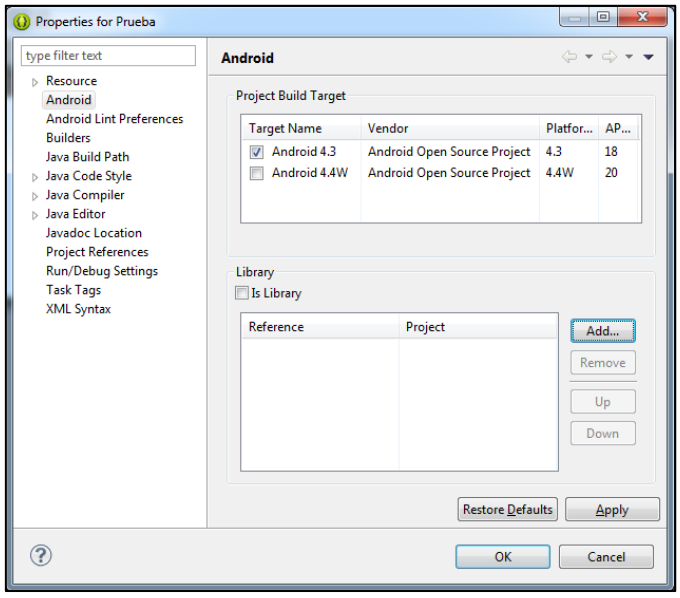
Si esto ocurre pulsamos sobre el proyecto con el botón derecho y seleccionamos “properties”. En el menú de la izquierda seleccionamos Android y marcaremos en la derecha la versión de Android con la que estamos desarrollando, en este caso la versión 4.3 de Android. Pulsamos “OK”.



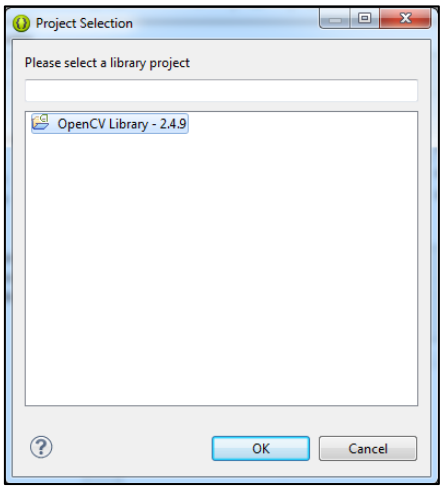
Si todo ha ido correctamente el error anterior desaparecerá.

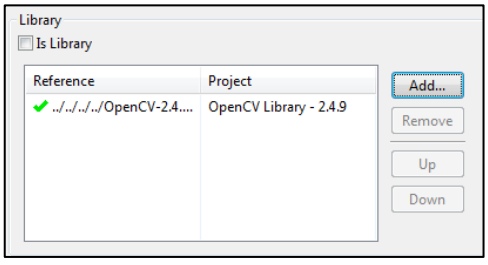


El último paso a realizar será añadir la librería de OpenCV a nuestro proyecto Android. Para ello pulsamos con el botón derecho en él y seleccionamos properties. Teniendo el apartado de Android seleccionado procederemos a añadir una librería externa (OpenCV), para ello pulsaremos el botón "Add…"



Como hemos añadido el proyecto de OpenCV anteriormente, la ventana que se mostrará al pulsar sobre añadir, tendrá un único proyecto disponible. Lo seleccionaremos y pulsaremos "OK".





Observaremos que se ha añadido el proyecto de OpenCV a nuestro proyecto Android. Si pulsamos "OK" en esta ventana, habremos terminado de configurar nuestra aplicación para el uso de la librería OpenCV y poder empezar a desarrollar para Android.

C. Lista de clases

En este apéndice se recogen las clases empleadas en el proyecto con sus métodos más importantes.

- **MainActivity.java: Menú de inicio.**
  - onCreate(Bundle)
  - buttonTakePicture(View)
  - buttonSelectGallery(View)
  - buttonSelectGalleryAuto(View)
  - buttonTakePicAuto(View)
  - buttonStreamingVideo(View)
  - test(View)
  - buttonMeasureDistance(View)
  - buttonMeasureHeight(View)
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
- **TakeAPictureActivity.java: Toma de fotografía medida manual.**
  - onCreate(Bundle)
  - onActivityResult(int, int, Intent)
  - onWindowFocusChanged(boolean)
  - onTouch(View, MotionEvent)
  - setMeasure(View)
  - getMeasure(View)
  - setDialogMeasure()
  - setPic()
  - galleryAddPic()
  - createImageFile()
  - dispatchTakePictureIntent()
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
- **GalleryPictureActivity.java: Carga de fotografía medida manual.**
  - onCreate(Bundle)
  - onActivityResult(int, int, Intent)
  - onWindowFocusChanged(boolean)
  - setPic()
  - setInit()
  - onTouch(View, MotionEvent)
  - setMeasure(View)
  - getMeasure(View)
  - setDialogMeasure()
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
- **AutomaticTakePictureActivity.java: Toma fotografía medida automática.**
  - onCreate(Bundle)
  - onActivityResult(int, int, Intent)
  - onWindowFocusChanged(boolean)
  - setPic()
  - setInit()
  - onTouch(View, MotionEvent)
  - setMeasure(View)
  - getMeasure(View)
  - setDialogMeasure()
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)

- **AutomaticGalleryPictureActivity.java: Carga fotografía medida automática.**
  - onCreate(Bundle)
  - onActivityResult(int, int, Intent)
  - onWindowFocusChanged(boolean)
  - onTouch(View, MotionEvent)
  - getMeasure(View)
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
- **FeatureMatching.java: Método detección objeto.**
  - initComparation()
  - filterMatchesByHomography(MatOfKeyPoint, MatOfKeyPoint, MatOfDMatch)
- **MeasureDistanceActivity.java: Medida de distancias.**
  - onCreate(Bundle)
  - freeze(View)
  - onWindowFocusChanged(boolean)
  - unfreeze(View)
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
  - onResume()
  - onPause()
  - surfaceChanged(SurfaceHolder, int, int, int)
  - surfaceCreated(SurfaceHolder)
  - surfaceDestroyed(SurfaceHolder)
  - onAccuracyChanged(Sensor, int)
  - onSensorChanged(SensorEvent)
- **MeasureHeightActivity.java: Medida de alturas.**
  - onCreate(Bundle)
  - getDistance(View)
  - getHeight(View)
  - onWindowFocusChanged(boolean)
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
  - onResume()
  - onPause()
  - surfaceChanged(SurfaceHolder, int, int, int)
  - surfaceCreated(SurfaceHolder)
  - surfaceDestroyed(SurfaceHolder)
  - onAccuracyChanged(Sensor, int)
  - onSensorChanged(SensorEvent)
- **GetReferenceActivity.java: Obtención imagen referencia.**
  - onCreate(Bundle)
  - onWindowFocusChanged(boolean)
  - cameraClick(View)
  - takepic(View)
  - savepicleft(View)
  - createImageFile2()
  - createImageFile()
  - onResume()
  - onCreateOptionsMenu(Menu)
  - onOptionsItemSelected(MenuItem)
  - surfaceChanged(SurfaceHolder, int, int, int)
  - surfaceCreated(SurfaceHolder)
  - surfaceDestroyed(SurfaceHolder)

Bibliografía

[1] Android Developers. Get the Android SDK.
http://developer.android.com/sdk/index.html
Septiembre 2014

[2] Android Developers. ADT Plugin.
http://developer.android.com/tools/sdk/eclipse-adt.html
Septiembre 2014

[3] OpenCV for Android. OpenCV4Android usage models.
http://opencv.org/platforms/android/opencv4android-usage-models.html
Septiembre 2014

[4] Photo Measure Lite. Big Blue Pixel Inc.
https://play.google.com/store/apps/details?id=com.bigbluepixel.photomeasures.lite
Septiembre 2014

[5] Measure & Sketch. Druidlab.
https://play.google.com/store/apps/details?id=com.druidlab.mymeasures
Septiembre 2014.

[6] SmartMeasure. Smart Tools co.
https://play.google.com/store/apps/details?id=kr.sira.measure
Septiembre 2014.

[7] Draw.io. JGraph Ltd.
https://www.draw.io/
2005-2014

[8] OpenCV. OpenCV Downloads
http://opencv.org/downloads.html
Septiembre 2014

[9] Android Ya.
http://www.javaya.com.ar/androidya/index.php
Septiembre 2014

[10] Android Developers. Sensors Overview.
http://developer.android.com/guide/topics/sensors/sensors_overview.html
Septiembre 2014

[11] Stackoverflow. Stack exchange inc.
http://stackoverflow.com/
Septiembre 2014

[12] Android Developers. Training. Getting Started.
http://developer.android.com/training/index.html
Septiembre 2014

[13] Android Developers. Capturing Photos.
http://developer.android.com/training/camera/index.html
Septiembre 2014

[14]OpenCV. Template matching.
http://docs.opencv.org/doc/tutorials/imgproc/histograms/template_matching/template_matching.html
Septiembre 2014

[15] OpenCV. Feature matching.
http://docs.opencv.org/trunk/doc/py_tutorials/py_feature2d/py_matcher/py_matcher.html
Septiembre 2014

[16] OpenCV. Object Detection.
http://docs.opencv.org/modules/imgproc/doc/object_detection.html
Septiembre 2014

[17] Image Matcher – OpenCV. Mustafa AKIN.
https://play.google.com/store/apps/details?id=in.mustafaak.imagematcher
Abril 25, 2013

[18]OpenCV. Android development with OpenCV
http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/dev_with_OCV_on_Android.html
Septiembre 2014