



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

---

---

**FACULTAD DE CIENCIAS DE LA ELECTRÓNICA**

**“SISTEMA DE MEDICIÓN CON PUNTOS FIJOS”**

**TESIS PROFESIONAL**

Para obtener el título de

**LICENCIATURA EN ELECTRÓNICA**

**PRESENTA:**

**JOSÉ DAVID SÁNCHEZ ÁVILA**

**ASESOR DE TESIS:**

**DR. GUSTAVO MENDOZA TORRES**



**JUNIO 2017**

# CONTENIDO

<b>Introducción:</b> .....	1
<b>Capítulo 1 MARCO TEÓRICO</b> .....	4
1.1 Dimensión .....	4
1.1.1 Medición.....	4
1.1.2 Principio de indeterminación de Heisenberg.....	5
1.2 Estándar de medición .....	7
1.3 Instrumentos de medición .....	9
1.4 Tecnología de la medición.....	10
1.4.1 Distanciómetro .....	17
1.4.2 Efecto Doppler .....	18
1.4.3 Diafonía .....	21
1.4.4 LIDAR y tecnología de tiempo de vuelo (ToF).....	22
<b>Capítulo 2 DESARROLLO TEÓRICO Y PRÁCTICO</b> .....	23
2.1 Sistema de medición con puntos fijos (SMPPF) .....	23
2.2 Planteamiento del problema.....	24
2.2 Diagrama de flujo.....	25
2.3 Sensor TeraRanger One .....	26
2.3.1 Características Generales del Sensor .....	26
2.3.2 Puertos de comunicación.....	27
2.3.3 Módulo de transferencia .....	28
2.3.4 Interfaz.....	29
2.4 Módulo de cámara V2 .....	31
2.5 Acelerómetro ADXL345.....	32
2.6 Raspberry Pi .....	34
2.6.1 Características Generales.....	35
2.6.2 Puertos de comunicación de la Raspberry Pi 3 B .....	38
2.7 Ley de Cosenos.....	39
2.8 Instalación de Android.....	41
2.9 Software Development Kit Android .....	42
2.9.1 Editor IDE Eclipse.....	43

2.9.2 Android Studio .....	44
2.10 Desarrollo de la Aplicación.....	45
2.10.1 Diseño gráfico .....	55
<b>Capítulo 3 PRUEBAS Y RESULTADOS</b> .....	56
3.1 Estructura de Funcionamiento .....	56
3.2 Sensor de Distancia.....	57
3.2.2 Programación.....	58
3.3 Cálculo del ángulo con el acelerómetro ADXL345.....	60
3.4 Conversión de la imagen.....	61
3.5 Cálculo de la distancia final.....	62
3.6 Unificación de los Sistemas.....	63
3.7 Apoyo visual Láser.....	64
3.8 Pruebas del Sistema .....	65
3.9 Comparación con sistemas comerciales .....	68
3.10 Resultados finales del sistema .....	69
3.11 Rango de medición del sistema con puntos fijos.....	70
<b>CONCLUSIONES</b> .....	72
<b>APÉNDICE A</b> .....	73
<b>APÉNDICE B</b> .....	74
<b>APÉNDICE C</b> .....	75
<b>APÉNDICE D</b> .....	80
<b>BIBLIOGRAFÍA</b> .....	81

## LISTA DE FIGURAS

Figura 1.1 Representación de un sistema Radar.....	11
Figura 1.2 Representación de un sensor ultrasónico.....	12
Figura 1.3 Espectro electromagnético.....	13
Figura 1.4 Representación de un sensor infrarrojo.....	13
Figura 1.5 Partes que conforman el láser.....	14
Figura 1.6 Funcionamiento de un láser.....	15
Figura 1.7 Efecto Doppler en una onda periódica.....	18
Figura 1.8 Efecto de diafonía en cables cercanos.....	21
Figura 1.9 Tipos de sensores de sistemas LIDAR.....	22
Figura 2.0 Principio de funcionamiento del SMPF.....	23
Figura 2.1 Sensor TeraRanger One "Spider".....	26
Figura 2.2 Pines del sensor.....	27
Figura 2.3 Interfaz del software Hterm.....	29
Figura 2.4 Datos binarios del sensor.....	30
Figura 2.5 Medida obtenida por el sensor en milímetros.....	30
Figura 2.6 Modulo de cámara V2.....	31
Figura 2.7 Acelerómetro ADXL345.....	32
Figura 2.8 Ejes x, y, z que utiliza el acelerómetro.....	32
Figura 2.9 Diagrama a bloques del funcionamiento del acelerómetro.....	33
Figura 2.10 Raspberry Pi 3 B.....	34
Figura 2.11 Características internas de la Raspberry Pi 3 B.....	35
Figura 2.12 Puertos de comunicación.....	38
Figura 2.13 Ley de Cosenos.....	39
Figura 2.14 Funcionamiento del sistema.....	39
Figura 2.15 Programa Matlab.....	40
Figura 2.16 Cálculo de los puntos.....	40
Figura 2.17 Android en tarjeta de memoria Micro SD.....	41
Figura 2.18 Logo Android SDK.....	42
Figura 2.19 Logo Android Studio.....	44
Figura 2.20 Diagrama Framework.....	45
Figura 2.21 Diagrama Librerías.....	46
Figura 2.22 Diagrama Kernel Linux.....	48
Figura 2.23 Distribución.....	48
Figura 2.24 Creación de nuevo proyecto.....	49
Figura 2.25 Configuración de compatibilidad.....	49
Figura 2.26 Actividad básica.....	50
Figura 2.27 Actividad principal.....	50

Figura 2.28 Soporte para lenguaje C++ .....	51
Figura 2.29 Módulos de procesamiento de imagen .....	51
Figura 2.30 Código de compilación .....	52
Figura 2.31 Selección de emulador .....	53
Figura 2.32 Configuración de memoria RAM .....	53
Figura 2.33 Versión de Android.....	54
Figura 2.34 Inicio de la emulación.....	54
Figura 2.35 Aplicación instalada.....	55
Figura 2.36 Gráficos diseñados .....	55
Figura 3.1 Diagrama sistema de medición con puntos fijos.....	56
Figura 3.2 Funcionamiento del sensor TeraRanger One.....	57
Figura 3.3 Código de programación .....	58
Figura 3.4 Datos binarios en Hterm .....	59
Figura 3.5 Acelerómetro y Arduino Nano .....	60
Figura 3.6 Interfaz gráfica con referencia en 0 grados.....	60
Figura 3.7 Interfaz gráfica con ángulo medido.....	60
Figura 3.8 Comparación de imagen vectorial con imagen de mapa de bits .....	61
Figura 3.9 Imagen bmp con los puntos A y B .....	61
Figura 3.10 Diagrama del procesamiento de la información .....	62
Figura 3.11 Funcionamiento del sistema .....	62
Figura 3.12 Sistemas unificados en prototipo.....	63
Figura 3.13 Apoyo visual láser.....	64
Figura 3.14 Marcación con láser en el punto A.....	65
Figura 3.15 Marcación con láser en el punto B.....	65
Figura 3.16 Referencia punto A.....	66
Figura 3.17 Punto A en la aplicación .....	66
Figura 3.18 Medida obtenida por el sistema .....	66
Figura 3.19 Medición con cinta métrica.....	67
Figura 3.20 Medición Sistema con puntos fijos .....	67
Figura 3.21 Sistema de medición con puntos fijos.....	69
Figura 3.22 Imagen con la medición .....	69
Figura 3.23 Rango de medición máximo y mínimo.....	70

## LISTA DE TABLAS

Tabla 1.1 Tipos de láser de acuerdo al medio activo .....	16
Tabla 2.1 Especificaciones del sensor.....	26
Tabla 3.1 Mediciones calculadas y obtenidas experimentalmente .....	67
Tabla 4.1 Tabla de comparación con sistemas comerciales .....	68
Tabla 5.1 Mediciones del sistema - precisión $\pm 2$ dm.....	70
Tabla 6.1 Comparación de sensores electrónicos.....	73
Tabla 7.1 Comparación de la familia Raspberry.....	76
Tabla 8.1 Especificaciones del acelerómetro ADXL345.....	77

---

## Introducción:

Desde la aparición de los primeros seres humanos y su constante evolución como civilización, la medición pasó a ser parte de la vida fundamental de los primeros pobladores, el desarrollo intelectual, el dominio y la expansión de los primeros asentamientos humanos progresó significativamente por la necesidad de medir pesos, cantidades, y dimensiones, dando lugar a la numeración y a los primeros sistemas de medición, las primeras referencias para medir tuvieron inicios rústicos, desde el uso de pequeñas ramas, palos, hasta partes del cuerpo humano, así la medición antigua dio inicio a los primeros sistemas de medición con unidades.

Los registros históricos dónde se vislumbra el inicio de las primeras mediciones hechas por humanos corresponde a antiguas civilizaciones que nos llevan al Oriente cercano, principalmente Mesopotamia, Persia y Egipto, podemos encontrar esbozos de antiguos mapas, planos, uso de la balanza entre otros ejemplos de distintos tipos de medición que datan desde el año 5000 a. C., estos hechos nos muestran el alcance que se tuvo en su época, posteriormente en los siglos venideros, se buscaba la unificación de distintas medidas con el principal propósito de facilitar el comercio entre las principales sociedades de la época, así pues durante la revolución francesa en 1789, se convocaron comités científicos con los más brillantes personajes de aquel entonces con el propósito de unificar pesos y medidas, una de las primeras en definirse fue el metro, para esto Jean Baptiste, Joseph Delambre y Pierre Méchain, se encargaron de medir en 1791 un arco de meridiano entre Dunkerke a Barcelona, la cuál sería la base del patrón del metro. Posteriormente en 1792 la Academia de Ciencias de Francia definía el metro como la diezmillonésima parte de la distancia que separa el Polo Norte con el ecuador terrestre a través de la superficie terrestre.

En 1889 la Comisión Internacional de Pesos y Medidas introdujo nuevos principios para denominar el metro y el kilogramo basados en un metro de platino e iridio que fueron depositados en cofre subterráneos a las afueras de París; La actual denominación del metro surgió en 1960 dónde la 11.<sup>a</sup> Conferencia de Pesos y Medidas definió el metro como 1 650 763,73 veces la longitud de onda en el vacío de la radiación naranja del átomo del criptón 86, logrando con esto una precisión cincuenta veces mayor a la obtenida en 1889.

---

En la actualidad es necesario medir objetos, superficies o distancias de forma exacta, sin perder detalles de las características métricas de estos. Por ejemplo, para medir superficies de casas, edificios, puentes. Si necesitamos conocer la distancia entre dos puntos fijos, podemos utilizar la tradicional cinta métrica o flexómetro, esta forma tradicional de medir tiene sus limitaciones, ya que no podemos medir puntos que se encuentran en las alturas o alejados de alguna superficie que nos proporcione las condiciones ideales para hacerlo. Existen en el mercado sistemas modernos basados en dispositivos electrónicos que nos permiten hacer este tipo de mediciones de manera más fácil y práctica. Estos sistemas emplean diversos tipos de sensores electrónicos como: micro-ondas, ultrasonido, infrarrojo, y de tipo laser para determinar la distancia entre puntos fijos, sin importar que estén en lugares de difícil acceso. Haciendo una exploración al mercado comercial de este tipo de medidores, se encuentra que en su mayoría son de origen extranjero y de alto precio, por lo que tomando algunos principios de funcionamiento de estos medidores en conjunto con ideas propias se desarrolla un sistema electrónico de medición con características similares en cuanto a funcionamiento, logrando una medición precisa. A su vez la creación de este sistema va a permitir crear una alternativa viable a los sistemas de medición ya existentes.

Como idea principal para este trabajo se tomó el principio de funcionamiento de un distanciómetro laser, que es un instrumento electrónico de medición que calcula la distancia desde el dispositivo hasta el siguiente punto al que se apunte con el mismo. El distanciómetro se creó para facilitar las mediciones cuando un flexómetro no podía llegar, si la distancia era muy larga y no había soporte, o cuando este se doblaba o no era lo suficientemente largo.

Para la medición de distancias el sensor del distanciómetro láser funciona según el tiempo de emisión de luz. Se emite un haz de luz y se refleja a través de un objeto; Se mide el tiempo que el haz de luz necesita para realizar el recorrido de la unidad al objeto y del objeto a la unidad; Ya que la velocidad de la luz es constante, el tiempo de ejecución permite calcular la distancia.

---

La propuesta de mi trabajo de tesis tiene un punto positivo sobre los modelos comerciales actuales, ya que el resultado es medir la distancia entre dos puntos fijos y mostrar la medición en una aplicación para el sistema operativo móvil Android; Esto permitirá además de tener una medición confiable, obtener una imagen que contiene la medida, para poder intercambiar la información o generar una base de datos para su extracción y edición. Como se describe en el desarrollo del siguiente trabajo se empleó conocimiento sobre las áreas de Electrónica, Programación, Óptica y Optoelectrónica.

El capítulo uno habla sobre el estado del arte, el marco teórico que abarca toda la terminología utilizada, marcos de referencia e información de los distintos dispositivos utilizados.

El capítulo dos nos muestra el uso del hardware y diseño del software de forma individual y la unión de ambos para el funcionamiento del sistema en su totalidad.

Finalmente en el capítulo tres se muestran las pruebas realizadas utilizando el sistema desarrollado, las etapas del mismo así como las conclusiones a las que se llegó después de trabajar con el sistema de medición con puntos fijos.

---

## *Capítulo 1*

### **1.1 Dimensión**

El término "dimensión" proviene del latín *dīmensiō* figurativo de *dēmētiri* 'medir', referido al aspecto o etapa de algún cuerpo. El concepto puede aplicarse a distintos hechos de acuerdo al contexto, puede tratarse de una característica, una circunstancia, una fase de algún cuerpo o de un hecho. En Física podemos definir dimensión como una de las propiedades métricas de un cuerpo u objeto de características matemáticas, características que en conjunto pueden definir un fenómeno físico, especialmente, las que se consideran en el espacio para determinar el tamaño de las cosas. Partiendo de este entendimiento físico y en los principios fundamentales de las matemáticas tradicionales, se infiere que al realizar una comparación de dimensiones previamente documentadas logramos una medición.

#### **1.1.1 Medición**

Llamamos medición a la acción que realizamos cuando comparamos una dimensión física de otra conocida. Durante siglos la medición ha formado parte del desarrollo humano, es uno de los factores fundamentales que ha permitido la evolución de la física y las matemáticas, en física al realizar una medición de dimensiones no geométricas se presentan problemas adicionales, generalmente relacionadas con la precisión y el efecto producido en el sistema, por lo que el error puede estar presente, es deducible que el aparato de medida debe contar con las características necesarias para minimizar el error y que no interfiera sobre el sistema físico a medir, de acuerdo a la mecánica clásica no existe un límite teórico a la precisión o el grado de perturbación que dicha medida provocará sobre el sistema esto puede ser contradictorio, pues la mecánica cuántica tiende a definir esto de otra manera, al igual que en experimentos sociales, se puede ver que a veces estos factores alteran los sujetos de medición provocando cambios al sistema.

Debemos tener en cuenta que las medidas realizadas generalmente cuentan con un grado de error, debido a imperfecciones físicas del instrumento empleado o las limitaciones del medidor, errores de prueba, por lo tanto, se debe de realizar la medición de manera que la perturbación producida por el medidor sea inferior que el producido por el error de prueba que pueda realizarse.

---

Por esta causa, una magnitud medida es considerada como una variable azarosa, por lo tanto se acepta que un proceso de medición es el adecuado si la media estadística de estas mediciones converge hacia la media general. En la mecánica clásica las limitaciones para el grado de precisión son siempre de origen tecnológico o práctico, sin embargo, en la mecánica cuántica existen límites teóricos para el grado de precisión que puede alcanzarse.

### 1.1.2 Principio de indeterminación de Heisenberg

En 1925 Werner Karl Heisenberg (Premio Nobel de Física 1932) enunció el llamado principio de incertidumbre o principio de indeterminación, el cual establece que es imposible medir magnitudes físicas observables y complementarias simultáneamente, con precisión absoluta, y el valor de la posición y la cantidad de movimiento de un objeto dado. Esto significa, que la precisión con que pueden ser medidas las cosas es hasta cierto punto limitada.

Para entender el principio de indeterminación empecemos por conocer las variables fundamentales de dicho principio, concretamente, la medida de la posición, la medida de la cantidad de movimiento del objeto. En la física clásica, la medida de la posición ( $x$ ) de un objeto puede ser definida como una función continua en el tiempo  $x = x(t)$ , si la masa de ese objeto es  $m$  y se mueve a una velocidad inferior a la de la luz, entonces el momento lineal  $p$  del objeto se define como la masa por velocidad, siendo la velocidad la primera derivada en el tiempo de la posición (ecuación 1.1).

$$p = m \frac{dx}{dt} \quad (1.1)$$

Podemos pensar que la relación de la incertidumbre establece una limitación sobre nuestra capacidad de medir que nos impide conocer la precisión arbitraria a la posición inicial  $x(0)$  y el momento lineal  $p(0)$ ; si pudiéramos conocer  $x(0)$  y  $p(0)$ , la física clásica permitiría conocer la posición y la velocidad de un objeto en cualquier instante; la solución a las ecuaciones de movimiento dependerían invariablemente de  $x(0)$  y  $p(0)$ , resolver las ecuaciones de movimiento nos llevaría a conocer un conjunto de trayectorias dependientes de  $x(0)$  y  $p(0)$ , según el valor que tomen  $x(0)$  y  $p(0)$ , se tendrá una trayectoria dentro de ese conjunto u otro, pero, la propia solución de las ecuaciones limita el número de trayectorias a un conjunto determinado de ellas. Según hemos analizado, con el principio de incertidumbre  $x(0)$  y  $p(0)$  no se pueden conocer exactamente, así que tampoco se conocen  $x(t)$  y  $p(t)$  en

---

Cualquier otro instante con una precisión arbitraria, por lo que la trayectoria que seguirá el objeto no podrá conocerse de una manera exacta.

El principio de indeterminación de Heisenberg indica que las medidas de la posición  $x$  y de la cantidad de movimiento  $p$  variarán debido a una distribución de probabilidad característica de un estado cuántico del sistema. Las medidas del objeto observable sufrirán una desviación estándar  $\Delta x$  de la posición y el momento  $\Delta p$ . (ecuación 1.2).

$$\Delta x \cdot \Delta p \geq \frac{h}{2\pi} \quad (1.2)$$

Donde:

$\Delta x$  = Indeterminación de la posición

$\Delta p$  = Indeterminación de la cantidad de movimiento

$h$  = Constante de Planck ( $h=6,626 \cdot 10^{-34}$  J · s)

Debemos mencionar que el principio de indeterminación no se origina de los instrumentos de medición, **sino hace referencia a la acción de medir**, aún con los instrumentos más modernos y precisos el principio sigue existiendo, con lo que entre más grande sea la precisión en las medidas de magnitudes, mayor será la indeterminación en la medida de la variables complementarias.

---

## 1.2 Estándar de medición

Llamamos unidad de medición a una cuantía que ha sido previamente estandarizada a partir de una magnitud física, establecida y respaldada por un convenio o por una ley, un valor de una cuantía física puede ser descrita como un múltiplo de la unidad que ha sido medida, una unidad de medición adquiere su valor a partir de un patrón establecido o de una estructuración de otras unidades previamente definidas, las primeras unidades establecidas por los primeros convenios internacionales se conocen como unidades básicas de medición, las que se establecieron posteriormente se llaman unidades derivadas. Un conjunto de unidades de medición que no posea con más de una unidad asociada es llamado sistema de unidades. Todas las unidades determinan cantidades escalares, en caso de las magnitudes vectoriales, se determina que cada una de las componentes está indicada en la unidad establecida.

### Sistema MKS

El sistema MKS conocido así por las iniciales de sus principales unidades (Metro, Kilogramo y Segundo) es un sistema de unidades de medición que sentó las bases del SI (Sistema Internacional de Unidades) que es el actual estándar internacional, el MKS consta principalmente de tres unidades de medida para distancia, masa y tiempo, este sistema no cuenta con un instituto regulador por lo que el valor de estas unidades puede diferir dependiendo el lugar y la época.

### Sistema Internacional de Unidades

Precursor del Sistema Métrico Decimal, el Sistema Internacional de Unidades conocido como (SI) es el estándar internacional de medición ocupado en todo el mundo exceptuando Estados Unidos de América, Birmania y Liberia. Se decretó en el año de 1960, en la XI Conferencia General de Pesas y Medidas, durante la cual se reconocieron inicialmente seis unidades físicas básicas: Longitud-Metro, Masa-Kilogramo, Tiempo-Segundo, Corriente Eléctrica-Amperio, Temperatura-Kelvin, Intensidad Luminosa-Candela. En 1971 se añadió la séptima unidad básica para la cantidad de sustancia, el mol.

---

## **Sistema Cegesimal de Unidades**

El Sistema Cegesimal de Unidades (CGS) es un sistema de unidades basado principalmente de tres unidades: Centímetro, Gramo y Segundo, de ahí su abreviatura, fue propuesto por Carl Friedrich Gauss en 1832 y posteriormente implementado en 1874 por la BAAS (British Association for the Advancement of Science) actualmente BA. Aunque está casi en su mayoría ha sustituido por el Sistema Internacional de Unidades, su implementación en algunos campos de la física tiende a ofrecer resultados técnicamente ventajosos de acuerdo al contexto de su aplicación tal es el caso en el área de electromagnetismo, donde su uso de unidades facilita la resolución de ecuaciones. La Oficina Internacional de Pesos y Medidas, reguladora del Sistema Internacional de Unidades, valora y reconoce estos hechos e incluye en sus referencias y equivalencias de algunas unidades electromagnéticas del Sistema Cegesimal de Unidades gaussiano, aunque desaconseja su uso.

## **Sistema Anglosajón de Unidades**

Este sistema de medición es el que utiliza principalmente los Estados Unidos de América, es la agrupación de unidades no métricas que se deriva de la evolución de las medidas locales a través de los siglos en esa región, tomando como origen principal a algunas unidades de la antigua Roma, así como de los intentos de Inglaterra por estandarizar ciertas medidas a través del Sistema Imperial, por lo que existe ciertas diferencias entre ambos sistemas. El sistema cuenta principalmente con cuatro unidades de medición para longitudes, la pulgada, el pie, la yarda y la milla. Cada una de estas unidades tiene dos definiciones ligeramente distintas, la principal y la de agrimensura, lo que ocasiona que existan dos diferentes sistemas de medición.

---

### 1.3 Instrumentos de medición

Los seres humanos siempre hemos tenido la necesidad de medir magnitudes físicas, desde tiempos antiguos se buscaban diversas maneras de hacerlo, como un ejemplo de esto fue la distancia, esto se consiguió utilizando herramientas primitivas básicas como; ramas de árboles, objetos rígidos, incluso utilizando partes del cuerpo humano. No obstante también existieron referencias para otro tipo de magnitudes como la masa y el tiempo, estas metodologías fueron evolucionando con el pasar de las eras, se fueron estandarizando las unidades y los métodos de medición, minimizando los márgenes de error, llegando al grado de exactitud que conocemos hoy en día, gracias a la implementación de la tecnología de alta precisión.

Los instrumentos de medición son dispositivos mecánicos o electrónicos que nos permiten contrastar magnitudes físicas distintas a través de un proceso de medición, estos se clasifican de acuerdo a la magnitud física a medir, por ejemplo ya sea para la masa se tiene la báscula, para el tiempo el reloj, para la presión atmosférica el barómetro, para la temperatura el termómetro, la velocidad el velocímetro, por mencionar algunas, para la distancia se han desarrollado una cantidad notable de dispositivos de medición mecánicos y electrónicos, actualmente existen en el mercado sistemas electrónicos que nos permiten realizar mediciones de distancias de manera más fácil y práctica. Estos sistemas emplean diversos tipos de sensores, de microondas, de ultrasonido, infrarrojo, y de tipo laser para determinar la distancia exacta entre superficies, sin importar las condiciones topográficas, por ejemplo las que están en lugares de acceso restringido al ser humano, o de características que requieran formas o métodos especiales para realizar la medición, estos dispositivos electrónicos han servido para el desarrollo de las ciencias exactas, así como en la tecnología industrial y comercial.

---

## 1.4 Tecnología de la medición

Existe una amplia gama de sensores electrónicos que nos permiten interpretar diferentes variables físicas para realizar una medición, dependiendo del tipo de medición a realizar cada uno de estos sensores adapta a las características y necesidades del usuario, entre los más utilizados a nivel industrial y comercial encontramos los sensores de tipo microondas, ultrasónicos, infrarrojos y láser.

### Microondas (ondas de radar)

El principio de los sensores de tipo microondas (ondas de radar) se basan en un sistema de antenas que emiten y reciben microondas ultracortas que se reflejan en una superficie (figura 1.1). Las señales se transmiten a la velocidad de la luz, siendo el tiempo entre la emisión y la recepción de las mismas proporcional a la distancia de la superficie. Gracias al procedimiento de propagación especialmente desarrollado para este tipo de sensores, se puede medir de forma fiable y exacta, incluso en intervalos de tiempo extremadamente cortos.

Los sensores de microondas funcionan relativamente con poca potencia, entre las bandas de frecuencia C (frecuencias por debajo a 6GHz) y K (frecuencia superior a 20GHz); la señal enviada al medio es reflejada por un objeto hacia la estructura de la antena, la cual tiene un diseño funcional de curvas y aristas; los objetos que más reflejan la señal son aquellos que presentan un ángulo de 90° hacia la superficie reflectiva de la antena. Las antenas con estructura de tres superficies que convergen hacia una esquina siempre van a reflejar las ondas que entren por su abertura hacia el emisor. Este tipo de reflectores, comúnmente llamados reflectores de esquina, se utilizan para detectar objetos en un radar, se utilizan en submarinos y barcos para su navegación y evitar impactos.

El valor que determina la medida de reflexión de un objeto a las ondas de radar se denomina RCS ("Radar Cross Section") por su traducción "sección transversal de radar" y es representado por ( $\sigma$ ), la potencia transmitida se refiere a la cantidad de potencia conseguida por la antena para la transmisión de la señal ( $P_t$ ), y está definida por la ecuación 1.3.

$$P_t = \frac{P_{rad} D_1}{4\pi R^2} \quad (1.3)$$

Siendo:

$P_t$  = potencia transmitida

$P_{rad}$  = potencia radiada por la antena de transmisión

$D_1$  = Ganancia direccional de potencia de la antena de transmisión (adimensional)

$R$  = distancia a la antena de transmisión

### Ecuación de radar

La potencia de la señal reflejada ( $P_r$ ) a la antena de recepción está descrita por la ecuación 1.4.

$$P_r = \frac{P_t G_t A_r \sigma F^4}{(4\pi)^2 R_t^2 R_r^2} \quad (1.4)$$

En la que:

$P_t$  = potencia transmitida

$G_t$  = ganancia de la antena de transmisión

$A_r$  = apertura efectiva (área) de la antena de recepción

$\sigma$  = sección transversal del radar, o coeficiente de decaimiento del objetivo

$F$  = factor de propagación del patrón

$R_t$  = distancia del transmisor al objetivo

$R_r$  = distancia del objetivo al receptor

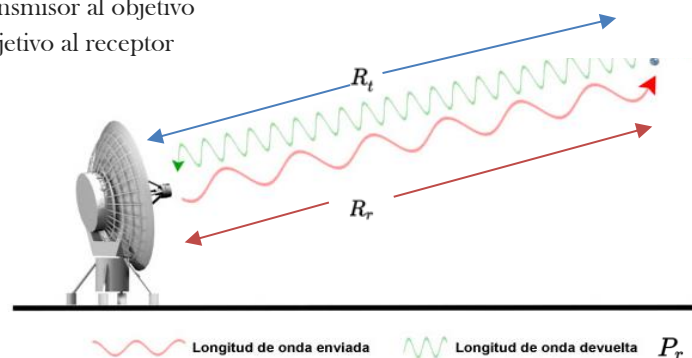


Figura 1.1 Representación de un sistema Radar

---

## Ultrasónicos

Los sensores ultrasónicos o sensores de ultrasonido son dispositivos detectores de proximidad que funcionan en base al envío y recepción de sonidos mecánicos no ionizantes cuyo rango de frecuencia oscila entre aproximadamente 20 000 Hz, su rango de detección va desde los centímetros hasta metros, el principio de funcionamiento es el de medir el tiempo en el que tarda en retornar una señal emitida, esta es reflejada en alguna superficie y el sensor recibe el eco producido el cual es transformado en señales eléctricas que son procesadas (figura 1.2). La velocidad con la cual llega la señal reflejada al receptor está íntimamente relacionada con la distancia de transmisión y rebote. Esto se puede expresar matemáticamente como:

$$d = 170 \text{ (m/s)} * t ,$$

**Siendo:**

d = distancia

m = metro

s = segundo

t = tiempo

Esto se puede demostrar partiendo de la ecuación de velocidad (ecuación 1.5).

$$v = 2d/t \text{ (1.5)}$$

**Donde:**

d es la distancia que recorre la señal para llegar al objeto,

v es la velocidad del sonido y

t el tiempo que tarda la señal en recorrer la distancia.

Asumiendo que la velocidad del sonido es aproximadamente unos 343,2 m/s, se puede obtener la distancia de acuerdo al tiempo que tarda en regresar la señal al emisor. Este tipo de sensores funcionan únicamente en el aire, pueden detectar diferente tipo de superficies u objetos con formas, figuras y colores diversos. Los materiales detectados pueden ser de estado líquido, sólido, o de superficie irregular aun así estos reflejan el sonido por lo que el sensor procesa el eco producido por estos para interpretarlo como una medición.

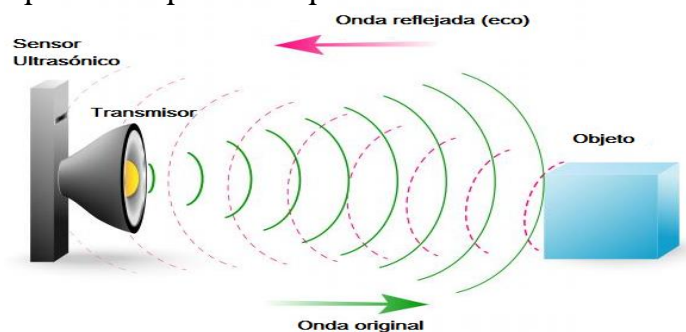


Figura 1.2 Representación de un sensor ultrasónico

## Infrarrojo

El sensor de radiación infrarroja (infrarrojo) es un sensor de medición de distancia, que se basa principalmente en la emisión y recepción de radiación lumínica en el espectro electromagnético infrarrojo cuyo rango oscila entre 0.7 y 1000 micrómetros (figura 1.3), con mayor longitud de onda que la luz visible, pero menor que las ondas de tipo microondas, una de las técnicas más habituales para la medición de la distancia de este tipo de sensores es mediante la triangulación del haz de luz colimada, como también se puede hacer una estimación de la distancia de un objeto a partir de la cantidad de energía recibida tras rebotar la luz sobre un objeto o superficie (figura 1.4). Este tipo de sensor electrónico presenta la desventaja de ser sensible a los cambios de luz ambiental como consecuencia de la radiación infrarroja presente en el ambiente. Por tal motivo, son sensores que se utilizan recomendablemente en entornos con iluminación artificial ya sea interiores o zonas de luz controlada.

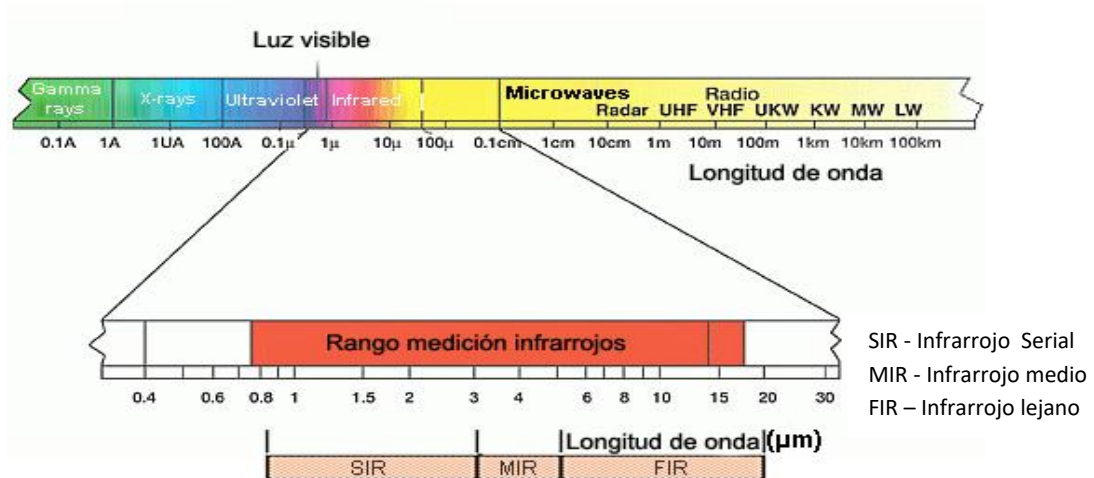


Figura 1.3 Espectro electromagnético

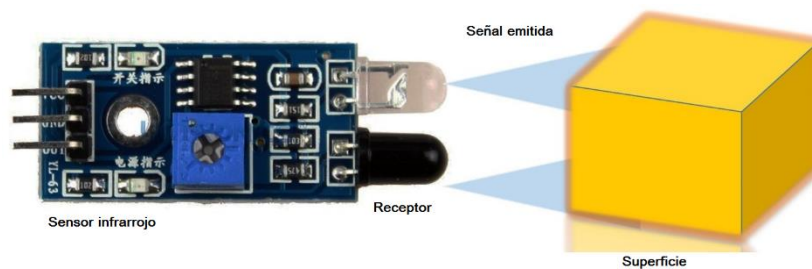


Figura 1.4 Representación de un sensor infrarrojo

---

## Láser (light amplification by stimulated emission of radiation)

Un sensor basado en la amplificación de luz de la radiación por emisión estimulada (Láser) funciona a diferencia de la luz LED (light-emitting diode) por medio de la proyección de un haz de luz concentrado que viaja en línea recta a la velocidad de la luz, aproximadamente 300 000 kilómetros por segundo, por lo que la posición del punto del haz se puede identificar rápidamente. Esto reduce considerablemente el tiempo del cálculo de una distancia en comparación con los sensores fotoeléctricos. El punto del haz se mantiene pequeño a lo largo de un gran rango, lo que elimina cualquier preocupación acerca de la distancia de detección y para determinar la distancia que se está midiendo se aprovecha el principio que la velocidad de la luz es constante, por lo que el tiempo de ejecución permite calcular el parámetro de la distancia.

Un láser necesita principalmente de estas partes fundamentales para su funcionamiento, una cavidad óptica resonante (generalmente compuesta de dos espejos dieléctricos que permiten una reflectancia cercana al 100%), un acoplador que tiene una reflectancia de menor valor y permite que la radiación láser salga de la cavidad; Un medio activo con ganancia óptica (que es el encargado de amplificar la luz), que puede ser de estado gaseoso, líquido, o sólido, para poder amplificar la luz, este medio necesita un aporte conciso de energía denominada generalmente, bombeo; El bombeo es generalmente un haz de luz "bombeo óptico" o puede ser una corriente eléctrica llamada "bombeo eléctrico".

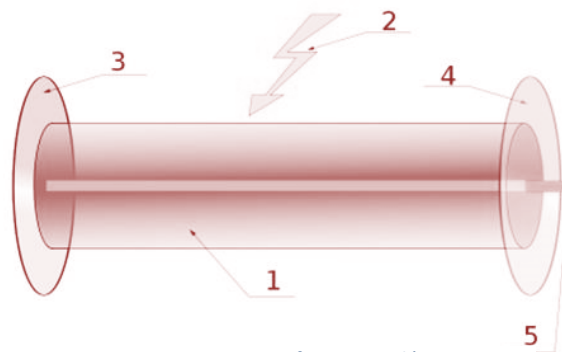


Figura 1.5 Partes que conforman un láser

- 1.- Medio Activo
- 2.- Energía bombeada
- 3.- Espejo reflejante al 100%
- 4.- Espejo reflejante al 99%
- 5.- Emisión del rayo láser

---

## Principio de funcionamiento

Se denomina sistema láser al conjunto de factores técnicos y conjunto de elementos que funcionan de manera precisa para obtener un haz de luz láser (figura 1.6). Para poder producir un haz de luz con características láser, deben conjugarse las partes fundamentales mencionadas anteriormente: medio activo, energía de bombeo y una cavidad resonante.

**Medio activo:** Consiste básicamente en un material que mediante el suministro de energía externa "bombeo" pueden generar ganancia a la luz y amplificarla. En el medio activo los átomos se excitan y se produce una emisión de fotones o rayos de luz al exterior de este. El medio puede ser de diversos materiales y es el que determina en mayor medida las propiedades del haz de luz láser como: la longitud de onda, la emisión (continua, o pulsada), potencia, color, etc. (Tabla 1.1).

**Fuente de energía externa o bombeo:** El bombeo excita a las moléculas del medio activo; que es equivalente a poner sus átomos en un nivel cuántico mayor. Una vez excitados los átomos, pueden emitir fotones que se propagan en el medio activo y genera más fotones; una producción en cadena. Todos los fotones generan fotones iguales. Estos, al ser iguales, se mueven igual y en conjunto generan rayos iguales, "rayos coherentes". Los medios de excitación pueden ser de tipo: electrónicos, de radio frecuencia, de sonido, eléctricos, e incluso de luz.

**Cavidad resonante:** Ésta permite por medio de espejos paralelos que los fotones producidos produzcan más fotones, y formen en conjunto un rayo de luz que puede pasar por el medio activo, salir como haz de luz láser o regresar al medio y continuar con el proceso de producción de fotones. Esta ida y vuelta de un rayo de luz dentro del medio, es fundamental en el sistema y es difícil de calibrar. Aquí surge el desafío de lograr que un rayo de luz pase y regrese al material activo.

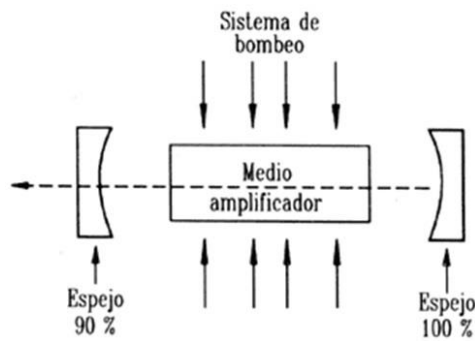


Figura 1.6 Funcionamiento de un láser

---

## Clasificación de láseres

[1]Según la UNE EN 60825-1/A2-2002 la peligrosidad de los láseres y en función del Límite de Emisión Accesible (LEA) se pueden clasificar los láseres en las siguientes categorías de riesgo:

**Clase 1:** Seguros en condiciones razonables de utilización.

**Clase 1M:** Como la Clase 1, pero no seguros cuando se miran a través de instrumentos ópticos como lupas o binoculares.

**Clase 2:** Láseres visibles (400 a 700 nm). Los reflejos de aversión protegen el ojo aunque se utilicen con instrumentos ópticos.

**Clase 2M:** Como la Clase 2, pero no seguros cuando se utilizan instrumentos ópticos.

**Clase 3R:** Láseres cuya visión directa es potencialmente peligrosa pero el riesgo es menor y necesitan menos requisitos de fabricación y medidas de control que la Clase 3B.

**Clase 3B:** La visión directa del haz es siempre peligrosa, mientras que la reflexión difusa es normalmente segura.

**Clase 4:** La exposición directa de ojos y piel siempre es peligrosa y la reflexión difusa normalmente también. Pueden originar incendios y explosiones.

Tipo de Laser	Longitud de Onda	Tipo de Luz
- Anhídrido Carbónico, Gas Carbónico o CO <sub>2</sub>	10600 nm	Invisible
- Neodimio: Ytrio- Aluminio-Granate (Nd:YAG)	1064 nm	Invisible
- Argón	488-514.5 nm	Visible
- Helio-Neón (He-Ne)	632.8 nm	Visible
- Erbium: YAG (Er: YAG)	2960 nm	Invisible
- Holmium: YAG (Ho:YAG)	2100 nm	Invisible
- XeCl-Excimer	308 nm	Invisible
- Arf-Excimer	193 nm	Invisible
- Diódico (Semiconductor, Arseniuro de Galio y Aluminio, As, Ga, Al.)	904 nm	Invisible

Tabla 1.1: Tipos de láser de acuerdo al medio activo

---

### 1.4.1 Distanciómetro

Un ejemplo de las tecnologías usadas actualmente para realizar mediciones precisas es el distanciómetro con sensor láser, también conocido como medidor láser basado en la electroerosión o por sus siglas en inglés EDM (electrical discharge machining), es un instrumento electrónico de medición que calcula la distancia desde el sensor integrado laser hasta el siguiente punto de una superficie al que se apunte con el mismo. Actualmente existen dos tipos de distanciómetros de acuerdo a su método de medición sus sensores varían en principios físicos de funcionamiento, los hay de tipo ultrasónicos y de tipo láser. Los primeros utilizan ultrasonido para calcular la distancia y los segundos un rayo láser visible.

El distanciómetro se creó para facilitar las mediciones en diversas escalas, desde nivel industrial hasta nivel comercial, cuando un sistema mecánico no es suficiente para determinar una medición los sensores laser del distanciómetro se encargan de manejar un rango de precisión muy alto, funcionan según el procedimiento del tiempo de ejecución de luz, se emite un haz de luz y se refleja a través de un objeto o superficie, se mide el tiempo que el haz de luz necesita para realizar el recorrido de la unidad al objeto y del objeto a la unidad. Ya que la velocidad de la luz es constante, el tiempo de ejecución permite calcular la distancia. De acuerdo a este principio de funcionamiento, el distanciómetro laser es mi principal referencia para el sistema desarrollado, porque la forma en que es sensada y procesada la información permite una precisión muy alta, sin embargo hay que mencionar que se debe de tener en cuenta algunas variables al momento de realizar la medición con el sistema de medición con puntos fijos, ya que los principios de funcionamiento físicos variarían al sustituir el sensor laser por un sensor infrarrojo, las condiciones de luz cambian para el sensor infrarrojo, por lo que se usan técnicas para disminuir el efecto que tienen estas circunstancias y determinar un cálculo con mayor precisión.

---

## 1.4.2 Efecto Doppler

El efecto Doppler es un fenómeno en el cual ocurre un aparente cambio en la frecuencia del sonido, una longitud de onda, u otro fenómeno periódico, que es debido al movimiento relativo del ente que la emite con relación a un observador que se encuentra fijo (figura 1.7). Su nombre se debe al físico austríaco Christian Andreas Doppler, quien lo propuso en el año de 1842 en la ciudad de Praga.

En la física clásica, donde las velocidades de la fuente (denominado emisor) y el observador (o también llamado receptor) con respecto al medio son inferiores a la velocidad de las ondas en el propio medio, la relación entre la frecuencia observada  $f$  y la frecuencia emitida  $f_0$  (ecuación 1.6).

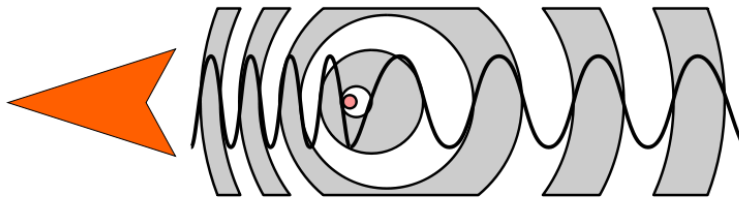
$$f = \left( \frac{c+v_r}{c+v_s} \right) f_0 \quad (1.6)$$

**Donde:**

$c$  = la velocidad de las ondas en el medio

$v_r$  = la velocidad del receptor en relación con el medio; positiva si el receptor se está moviendo hacia el emisor (y negativa en la dirección contraria).

$v_s$  = la velocidad de la fuente con respecto al medio; positiva si la fuente se aleja del receptor (y negativa en la dirección contraria).



*Figura 1.7 Efecto Doppler en una onda periódica*

---

[2] La ecuación 1.6 supone que el emisor está alejándose del observador o en caso contrario acercándose directamente. Si el emisor se acerca al observador con una velocidad constante, pero con una trayectoria "no incidente" ocurre que:

La frecuencia de la señal que el observador capta primero es más alta que la frecuencia emitida desde el objeto. Esto es porque se produce una disminución gradual de la frecuencia percibida a medida que el emisor se acerca al observador, coincidiendo la frecuencia percibida con la original cuando la onda llega desde una dirección perpendicular al movimiento relativo es decir, cuando fue emitida desde el punto más cercano al observador, aunque cuando se reciba la onda, la fuente y el observador ya no estarán en su posición más próxima.

Finalmente, el observador percibirá una continua disminución de la frecuencia a medida que se aleja del emisor. Cuando el observador se encuentra muy cerca de la trayectoria del objeto, la transición de alta a baja frecuencia es muy abrupta; en cambio, cuando el observador está lejos de la trayectoria del objeto, la transición de alta a baja frecuencia es gradual.

Si las velocidades  $v_s$  y  $v_r$  son menores en comparación con la velocidad de la onda, la relación entre la frecuencia observada  $f$  y la frecuencia emitida  $f_0$  esta dada por las ecuaciones 1.7 y 1.8.

Frecuencia observada

$$f = \left( 1 + \frac{\Delta v}{c} \right) f_0 \quad (1.7)$$

Cambio de frecuencia

$$\Delta_f = \frac{\Delta v}{c} f_0 \quad (1.8)$$

Donde:

$$\Delta_f = f - f_0$$

$\Delta v = v_r - v_s$  es la velocidad del receptor respecto al emisor

En el caso de ondas electromagnéticas la fórmula del efecto Doppler está dada por la ecuación 1.9.

$$f' = \gamma \frac{c+v}{c} f \quad (1.9)$$

---

Donde:

$f$  = la frecuencia del emisor

$f'$  = la frecuencia que ve el receptor

$v$  = la velocidad del emisor con respecto al receptor

$\gamma$  = el factor de Lorentz dado por la ecuación 1.10

$$\gamma = \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \quad (1.10)$$

Con el principio de este efecto sabemos que el pulso que se emite desde el sensor infrarrojo puede ser codificado para reducir la posibilidad de que el mismo pueda tener alteraciones en la medición por el movimiento del emisor lo que puede distorsionar la medición que se esté realizando. Al utilizar una técnica de codificación de datos en el sensor, estos no se pierden o alteran por el movimiento o las condiciones de luz, una vez decodificados dentro de la aplicación (que posteriormente se describirá) brindará una lectura necesaria que actuará en conjunto con la información de una cámara y un acelerómetro con lo que determinaremos la medida entre dos puntos de referencia en una superficie estática.

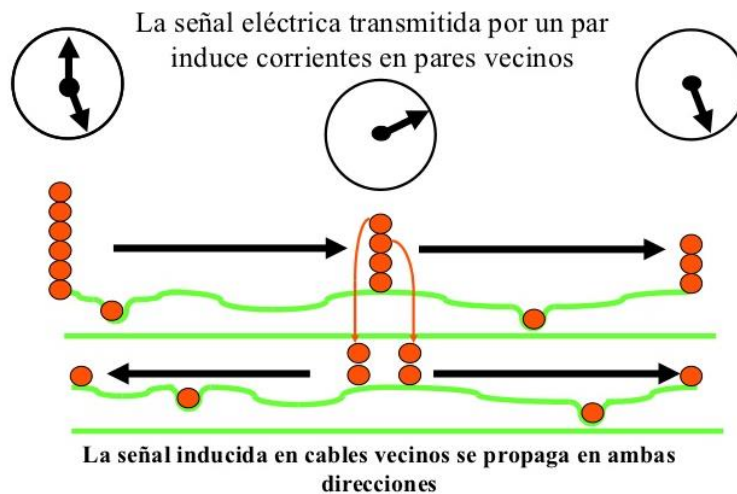
---

### 1.4.3 Diafonía

Denominada en inglés Crosstalk (XT), la diafonía ocurre cuando en dos o más circuitos una de las señales presentes en uno de ellos, considerado perturbador, aparece en el otro circuito, considerado alterado, ocasionando interferencias y funcionamientos erróneos en los circuitos. Otro caso dónde puede existir la diafonía, es en el caso de los cables de tipo entrelazados, dónde se manifiesta generalmente debido a los acoplamientos magnéticos entre los elementos que componen los circuitos sea el circuito conturbador y alterado o como consecuencia de desequilibrios de admitancia entre los hilos de ambos circuitos (figura 1.8). La diafonía se puede medir como la atenuación que existe entre el circuito perturbador y el perturbado, por lo que también se denomina atenuación de diafonía.

Para medir la diafonía se envía una señal de valor conocido por el circuito perturbador, y se mide el valor recibido en el circuito perturbado. Dependiendo si la medida que hagamos está en el mismo extremo desde el que estamos enviando la señal o en un extremo distante tendremos dos valores distintos. Para esto tenemos los conceptos de diafonía de extremo cercano (paradiafonía) y diafonía de extremo lejano (telediafonía).

Este hecho puede provocar variaciones en los circuitos y el sensor del sistema, por lo que hay que tener muy en cuenta que para realizar una adecuada retroalimentación de nuestra señal infrarroja de medición se opta por implementar un sensor con características que resuelvan este posible problema, por lo cual utilizamos el sensor TeraRanger One.



---

#### 1.4.4 LIDAR y tecnología de tiempo de vuelo (ToF)

[3]"El LIDAR (Light Detection and Ranging o Laser Imaging Detection and Ranging) es una tecnología de medición que nos permite determinar la distancia desde un emisor láser a un objeto o superficie utilizando un haz láser pulsado. " Un sistema con tecnología de tiempo de vuelo "time of flight" es un sistema que por medio de un sensor electrónico emisor (láser, infrarrojo, etc.) y una cámara especial permite obtener una distancia; El cálculo de esta se basa en la velocidad conocida de la luz, midiendo el tiempo de vuelo de una señal luminosa entre la cámara y un punto medido de una superficie. Un sistema con tecnología ToF es una clase de LIDAR.

Los dispositivos que utilizan la tecnología ToF enfocados a aplicaciones civiles comenzaron a emerger alrededor del año 2000, pues los procesos del semiconductor llegaron a ser bastante rápidos para tales dispositivos. Estos sistemas utilizan diferentes técnicas para el sensado de una superficie como: lineal, zigzag, elíptico, fibra óptica, etc. (figura 1.9). Los sistemas cubren rangos de unos milímetros hasta kilómetros. La distancia se determina de manera rápida, mientras que el rango mínimo inicia aproximadamente en 1 mm.

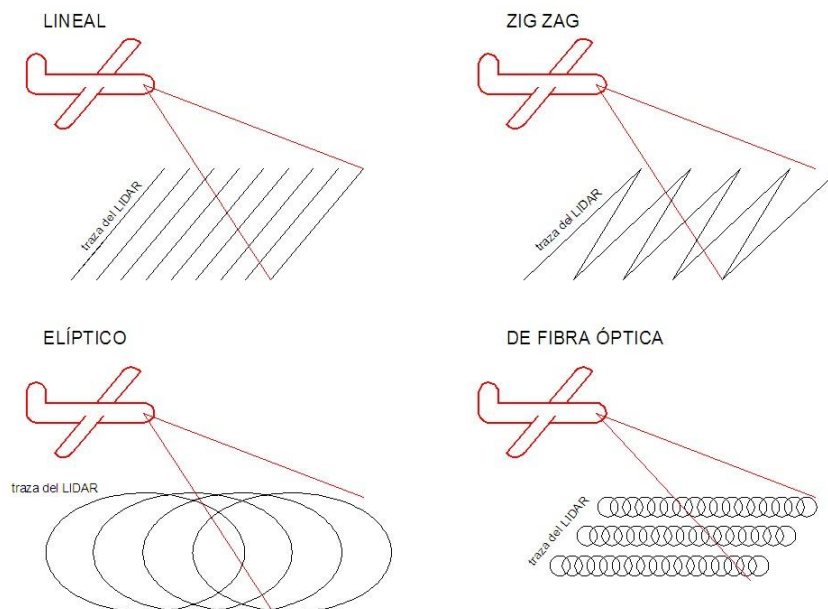


Figura 1.9 Tipos de sensado de sistemas LIDAR

---

## Capítulo 2

### 2.1 Sistema de medición con puntos fijos (SMPF)

El sistema de medición con puntos fijos hace uso de diferentes dispositivos electrónicos para obtener la medición entre dos puntos A y B; El sensor principal es de tipo infrarrojo que utiliza la tecnología ToF (Time of flight) para determinar una distancia hacia un punto fijo, una cámara para capturar la imagen de la superficie medida, y un acelerómetro para calcular el ángulo  $\Theta$  entre los dos puntos medidos; Estas tres variables de entradas son procesadas dentro de una aplicación que finalmente arroja la medida final (figura 2.0). Habiendo descrito la parte teórica de cada uno de los procesos para determinar una medición, ahora describiremos las características de cada uno de los dispositivos electrónicos, su función y finalmente su acoplamiento para dar forma al sistema de medición de puntos fijos.

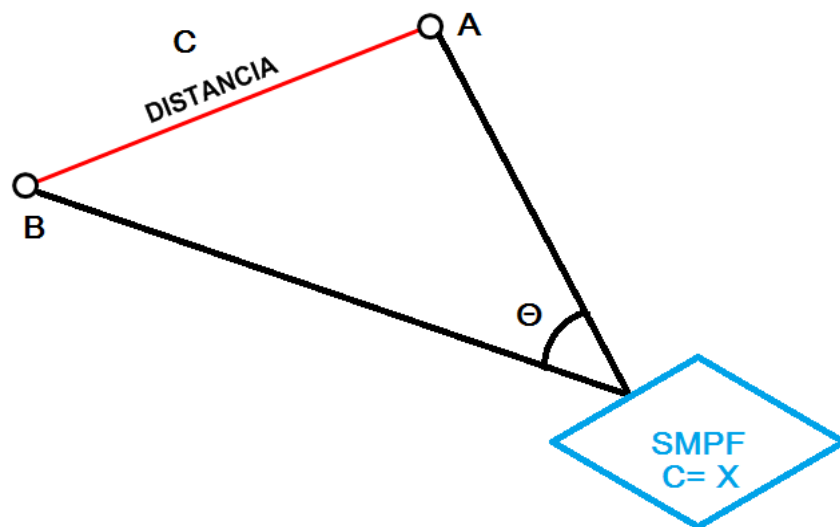


Figura 2.0 Principio de funcionamiento del SMPF

## 2.2 Planteamiento del problema

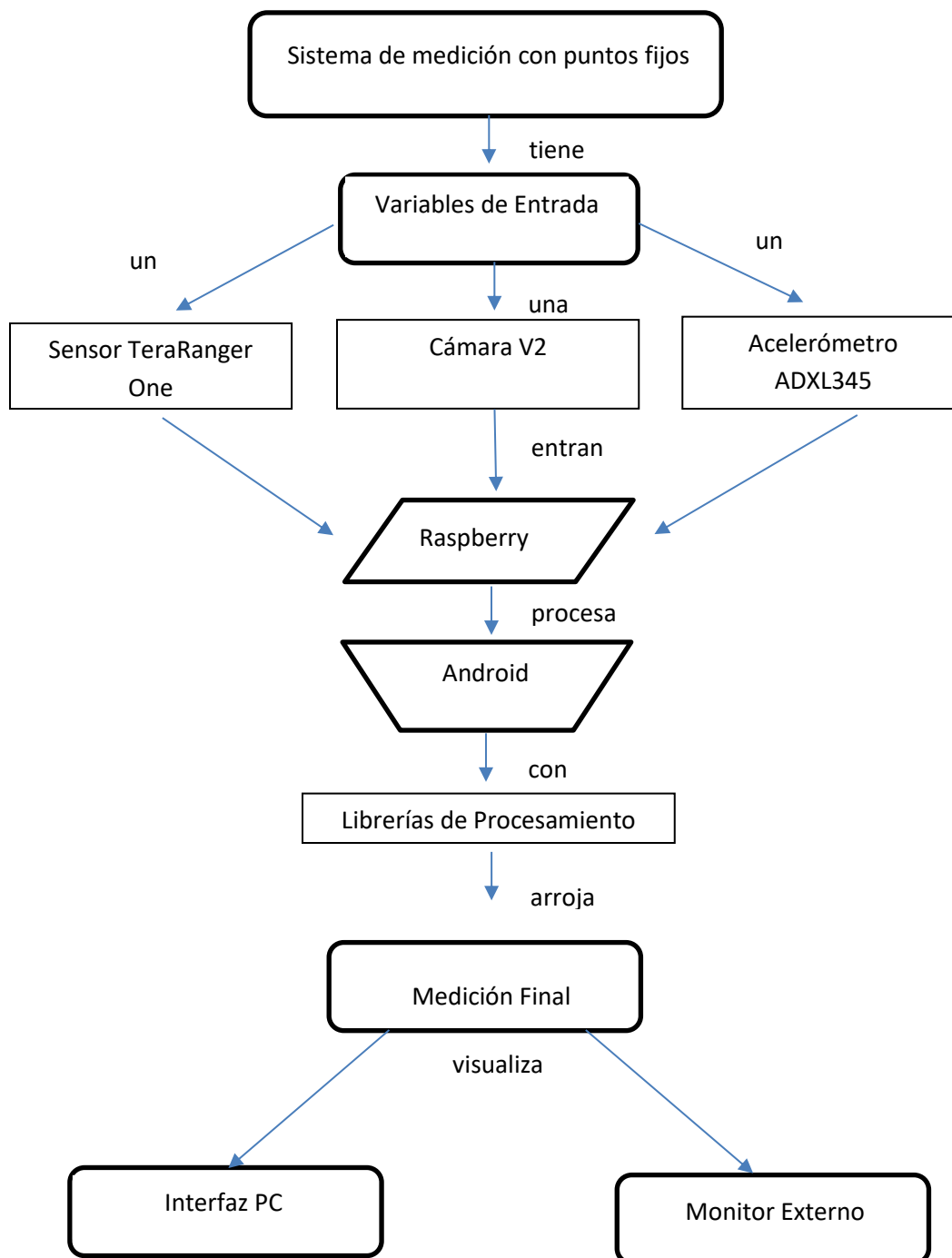
En el siguiente esquema se muestra el planteamiento del problema al medir dos puntos A -B y la solución con el SMPF



---

### 2.3 Diagrama de flujo

El siguiente diagrama de flujo con bloques funcionales representa la estructura de funcionamiento del sistema de medición con puntos fijos.



---

## 2.4 Sensor TeraRanger One

TeraRanger One es un sensor infrarrojo de medición de distancia ligero y de alto rendimiento que unido a la tecnología de tiempo de vuelo (ToF) time-of-flight camera proporciona un cálculo más rápido que un sensor de ultrasonido, y es mucho más pequeño y ligero que los sistemas basados en láser. Su pequeño tamaño, alto rendimiento y bajo consumo de energía lo hacen ideal para nuestro sistema de medición con puntos fijos, mientras que su rentabilidad también lo hace un aliado aun todavía mayor. Para la prevención de cambios de luz en múltiples entornos de medición existen dos versiones del sensor, tipo A “Box” y tipo B “Spider” (figura 2.1), que pueden funcionar simultáneamente sin afectar la lectura de cada uno. Los sensores A y B son muy similares. En este sistema se ha utilizado el sensor tipo B “Spider” para mediciones principalmente en exteriores ya que es más estable ante rayos ultra-violeta y condiciones ambientales especiales.

### 2.4.1 Características Generales del Sensor

El sensor cuenta con características de alto rendimiento que permitirán un excelente sensado hacia los puntos A y B (tabla 2.1).

Principle:	Infrared Time-of-Flight (ToF)
Range:	Up to 14m indoors (At least 5 to 6m in sunlight)
Update rate:	1000Hz in fast mode (Up to 600Hz in precision mode)
Range resolution:	0.5cm
Accuracy:	± 4cm in precision mode
Field of view:	3°
Supply voltage:	10V DC recommended (10 to 20V DC accepted)
Supply current:	50mA average (110mA peak @12V)
Interfaces:	1. UART (+5V level, up to 115200,8,N,1) 2. TWI (I2C compatible, +5V level, up to 400kHz, configurable address)
Connector:	15 pin DF13 (open-ended, 1.27mm pitch flat ribbon cable provided)
Designs:	Box, Spider, Frame
Weight:	8g (Spider, Frame) or 10g (Box)
Sensor types:	Type A (Box, Spider) and Type B (Box, Frame)

Tabla 2.1 Especificaciones del sensor

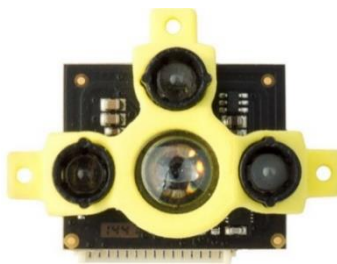


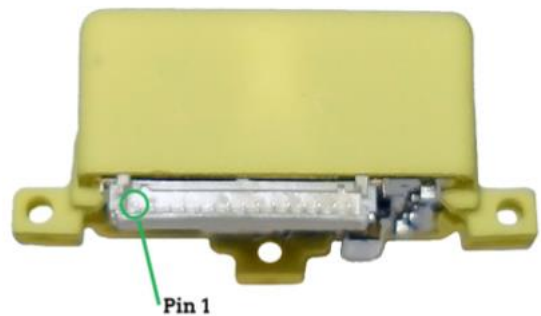
Figura 2.1 Sensor TeraRanger One “Spider”

---

## 2.4.2 Puertos de comunicación

Los principales puertos de comunicación del sensor son los pines en los cuales se realiza una conexión de tipo serial con el módulo de transferencia de datos (figura 2.2), estos permiten una comunicación de forma bidireccional por lo que se puede recibir y transmitir información entre el sensor, el módulo de transferencia y el pc en tiempo real.

Pin	Función
15	GND
14	Vcc (Rango de 10v a 20v)
13	Salida Serial TXD a 5 v
12	Entrada Serial RXD a 5v
11	TWI- SDA
10	TWI-SCL
9	I/O 1
8	I/O 2
7	Reset (Arduino)
6	Reset
5	MOSI (SPI para Flasheo)
4	MISO (SPI para Flasheo)
3	SCLK (SPI para Flasheo)
2	Salida 5v (para programar)
1	GND



*Figura 2.2 Pines del sensor*

---

### 2.4.3 Módulo de transferencia

El módulo de transferencia es principalmente un circuito electrónico que nos permite realizar la comunicación de tipo bidireccional entre el sensor TeraRanger One y una computadora a través de un puerto USB, usando una velocidad de 12 Mb por segundo nos da la oportunidad de tener un registro de los datos de la medición realizada en tiempo real, este módulo de transferencia está constituido principalmente de un microcontrolador Philips ISP1581 con las siguientes características:

- \* USB 2.0
- \* Soporte para la detección de velocidad automática (high y full)
- \* 7 Endpoints de entrada, 7 de salida. Soporta double buffer.
- \* Los endpoints pueden ser interrupt, bulk, o isochronous.
- \* 8Kb de memoria FIFO integrada.

Interfaces:

- \* Interface de bus independiente para la mayoría de los microcontroladores/microprocesadores (12.5 Mb/s)
- \* Interface DMA de alta velocidad (12.8 Mb/s)
- \* Interface directa con periféricos ATA/ATAPI
- \* Conexión al bus USB controlada por software (SoftConnect tm)
- \* Data transceiver y regulador de voltaje de 3.3 V integrados.

Se incorpora dentro del mismo microcontrolador el hardware necesario para conectarse al sensor a través del driver proporcionado por el fabricante. La comunicación con dicho modulo se realiza por medio de registros especiales y de memoria compartida (dual port RAM) y línea de interrupciones dedicada. Las comunicaciones que se manejan son de nivel punto a punto.

---

## 2.4.4 Interfaz

La interfaz del sensor TeraRanger One es de tipo UART (Universal Asynchronous Receiver-Transmitter), en español: Transmisor-Receptor Asíncrono Universal, en los pines 12 y 13, el sensor necesita un nivel de tensión de entrada de 5V (lógica alta) para funcionar de forma fiable, el nivel de tensión de salida es 5V. Se realiza la comunicación de la interfaz es a través del módulo de comunicación con el Software Hterm, esta es la estación dónde se programa el sensor para determinar el rango de distancia a medir, también es la interfaz gráfica dónde se aprecia la información de la medición realizada en datos binarios codificados para evitar el efecto Doppler.

En la siguiente figura se aprecia la interfaz inicial del software Hterm

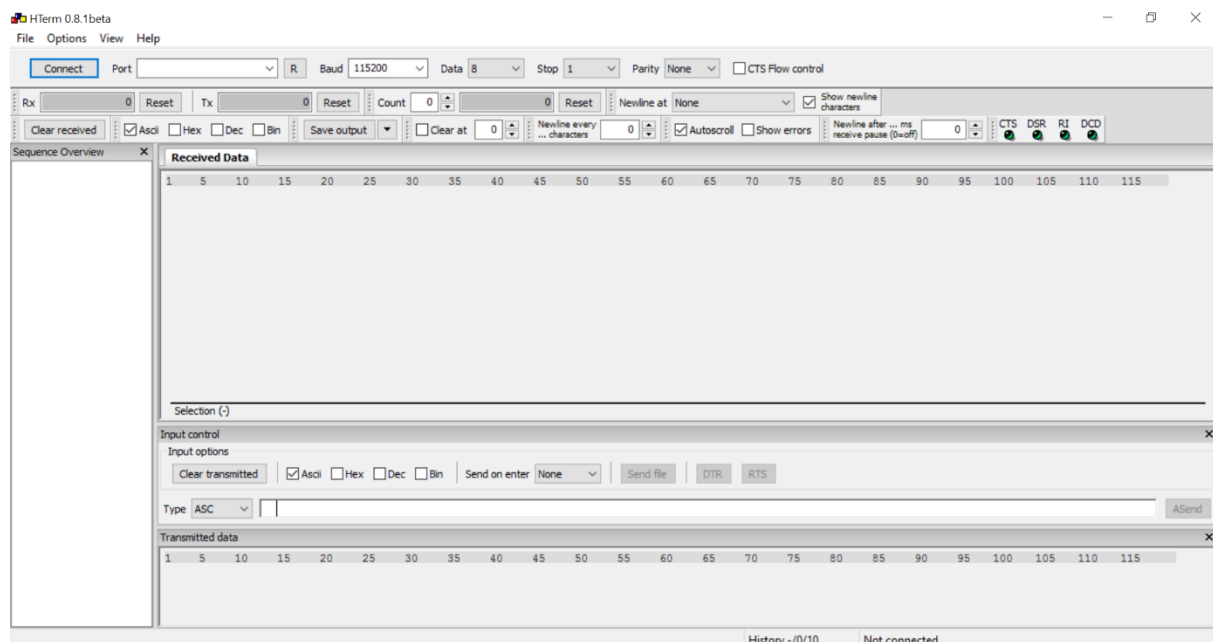


Figura 2.3 Interfaz del software Hterm

Al iniciar la secuencia del sensor a una distancia deseada el programa empieza a recibir la información censada y la representa en datos binarios codificados (figura 2.4), al mismo tiempo podemos ver la representación en mm (figura 2.5).

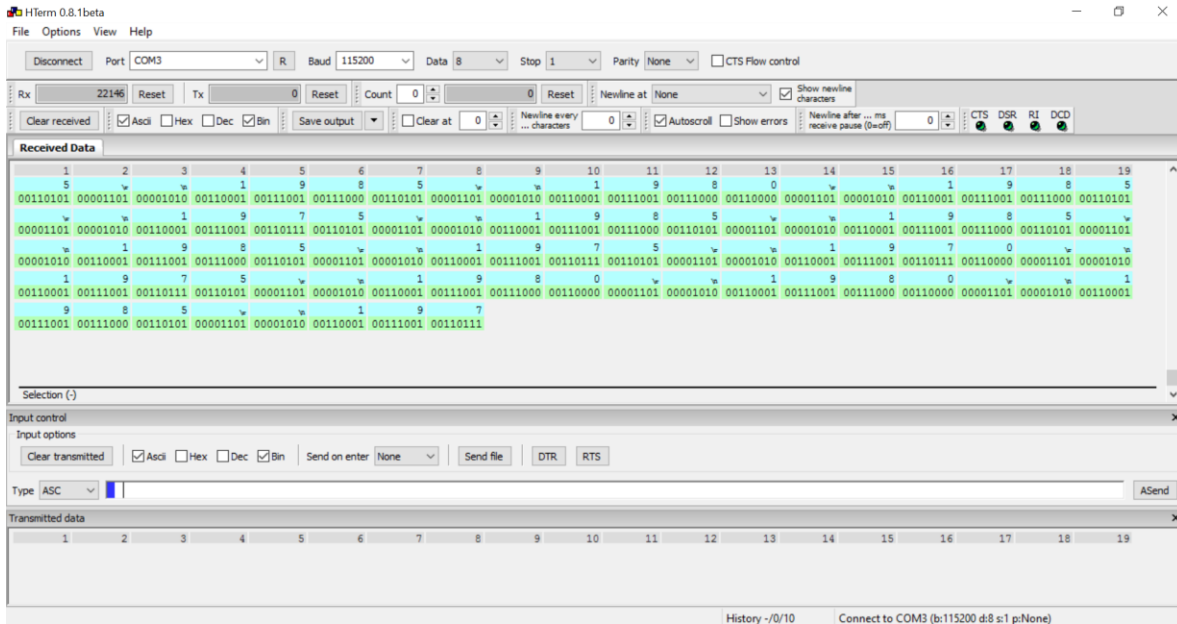


Figura 2.4 Datos binarios del sensor

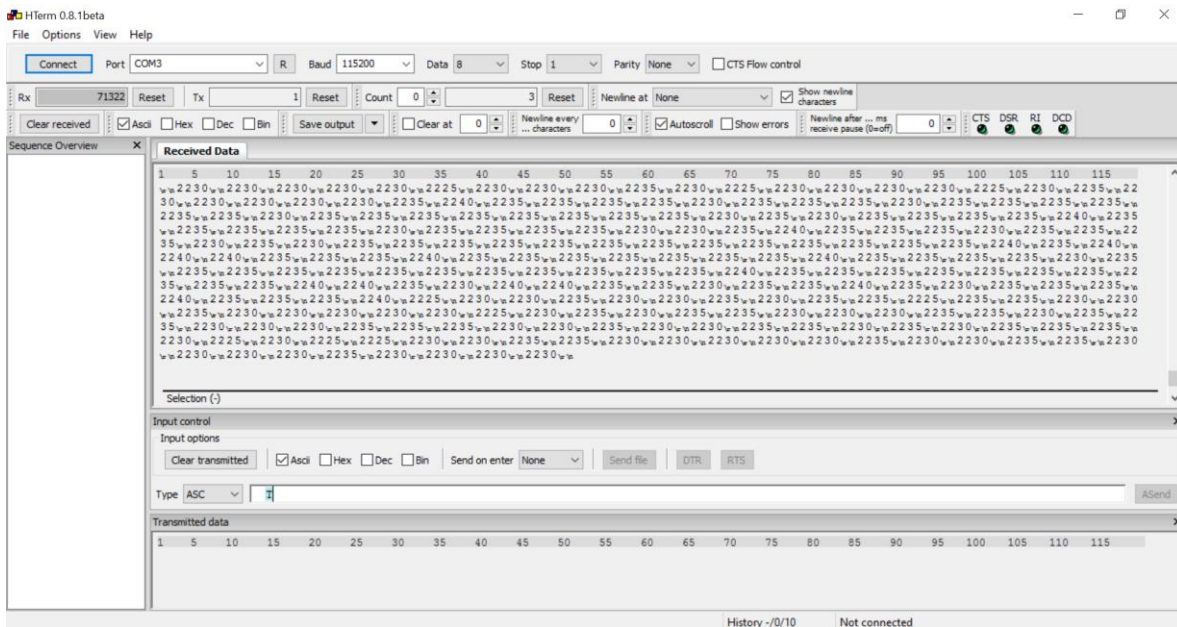


Figura 2.5 Medida obtenida por el sensor en milímetros

---

## 2.5 Módulo de cámara V2

Se implementa el módulo de la cámara V2 de la Raspberry Pi para tomar imágenes. El módulo tiene una cámara fija de foco de cinco megapíxeles que admite distintos modos de video, a 1080p con 30 frames por segundos, de 720p con 60 frames por segundo y de VGA. Se conecta a través de un cable a los pines de comunicación de la Raspberry Pi (figura 2.6).

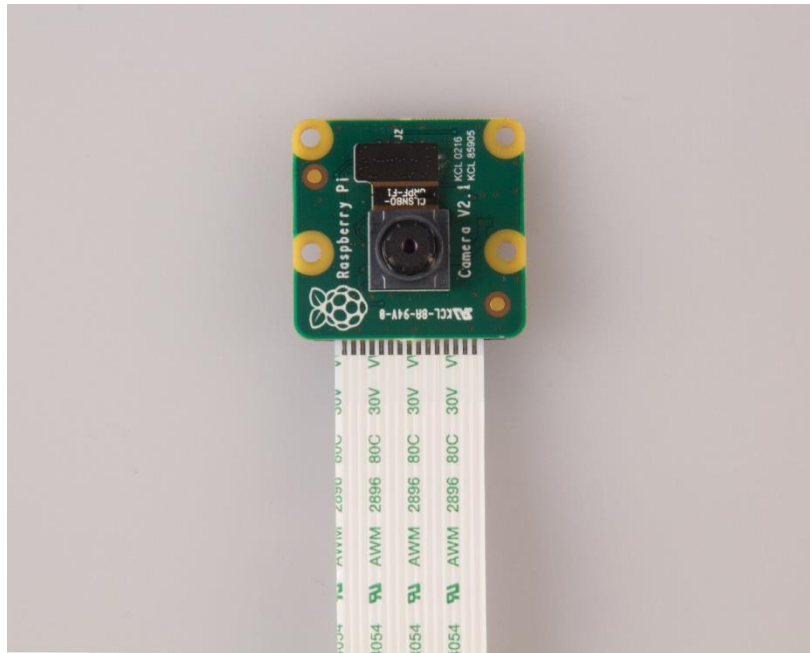


Figura 2.6 Modulo de cámara V2

El uso de esta cámara modular es principalmente para obtener una imagen de la superficie que se está midiendo y marcar en la imagen los dos vectores de referencia **A** y **B** para obtener la medida, esto se logra gracias a la librería de procesamiento de imágenes que está integrada dentro de la programación de la aplicación móvil, el marcar los puntos **A** y **B** sirve principalmente para establecer dos puntos de referencia en la imagen de la superficie capturada, la información de estos dos puntos es añadida a la proporcionada por el sensor TeraRanger One entre ambos puntos marcados y este., Estas tres referencias generan un “triángulo” entre los puntos marcados y el sensor, posteriormente por medio de la librería de Matlab para Android y los datos proporcionados por el acelerómetro ADXL345 se realiza un cálculo matemático basado en la ley de los cosenos para determinar la distancia entre los vectores de referencia, dando así la medición lineal de la superficie capturada.

---

## 2.6 Acelerómetro ADXL345

El acelerómetro ADXL345 es un tipo de sensor electrónico que detecta el movimiento o el giro, es decir, es capaz de responder con una señal eléctrica ante una perturbación inducida por la aplicación de una fuerza o la gravedad (figura 2.7). Este dispositivo es capaz de detectar si está en posición horizontal, vertical o si ocurre un cambio de posición en cuanto su punto de referencia.

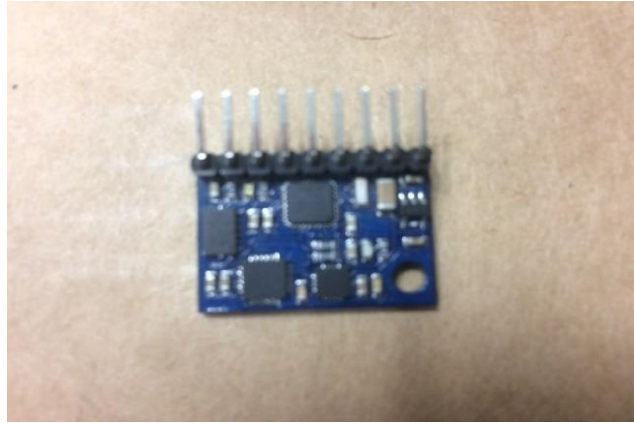


Figura 2.7 Acelerómetro ADXL345

El acelerómetro ADXL345 es un dispositivo electrónico de 3 ejes de alta resolución (figura 2.8), puede medir un ángulo de hasta  $60^\circ$  de apertura con una precisión de  $\pm 5^\circ$ ; Es pequeño, delgado, y tiene un bajo consumo de energía. El formato de los datos de salida es de complemento a dos de 16 bit. Gracias a su estructura interna (figura 2.9) los datos se pueden obtener a través de los pines de comunicación del dispositivo, es adecuado para medir la aceleración estática de la gravedad para aplicaciones de detección de inclinación, así como la aceleración dinámica ocasionada por algún movimiento o impacto. Su alta resolución (4 mg/LSB) permite medir cambios de inclinación de menos de  $1.0^\circ$ .

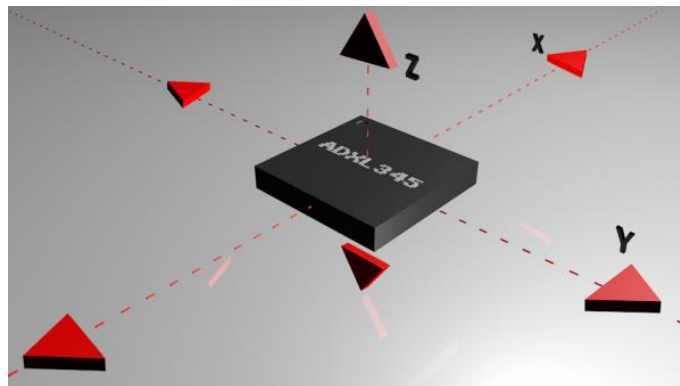


Figura 2.8 Ejes x, y, z que utiliza el acelerómetro

---

El uso del acelerómetro para nuestro sistema es principalmente para calcular el ángulo que se va a formar en el vértice que supondrá el sensor TeraRanger One, es muy importante tener en cuenta la lectura del acelerómetro ya que nos proporcionará el valor del ángulo de este vértice en todo momento en que se esté realizando la medición para poder determinar el valor que habrá en nuestros puntos de referencia, la programación se incluye dentro una librería originalmente basada en Linux modificada para Android de acuerdo al principio de funcionamiento de este, se puede monitorizar el ángulo de 3 ejes a partir de una referencia estática.

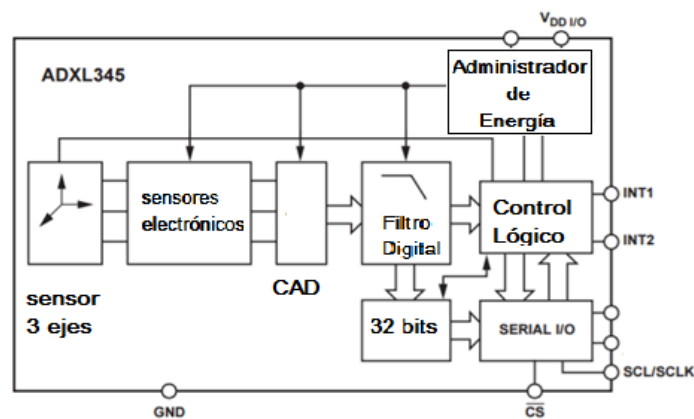


Figura 2.9 Diagrama a bloques del funcionamiento del acelerómetro

INT 1: Entrada

INT 2: SALIDA

SCL/SCLK: SEÑAL DE RELOJ

GND: TIERRA

VDD I/0: ALIMENTACIÓN

---

## 2.7 Raspberry Pi

Raspberry Pi es un computador de placa simple SBC (Single Board Computer) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi (figura 2.10). La implementación de la Raspberry Pi es principalmente al soporte que nos brinda electrónicamente, ya que nos otorga un alto número de posibilidades en aplicaciones de software y hardware, esta tarjeta de desarrollo cuenta con la integración de un slot para una memoria ROM y una memoria RAM integrada, por lo que es factible cargar en él un sistema operativo, en este caso el sistema operativo móvil Android que será la base dónde se ejecutará posteriormente la aplicación que realiza la medición en conjunto con el sensor TeraRanger One, el módulo de cámara V2, y el acelerómetro ADXL345.

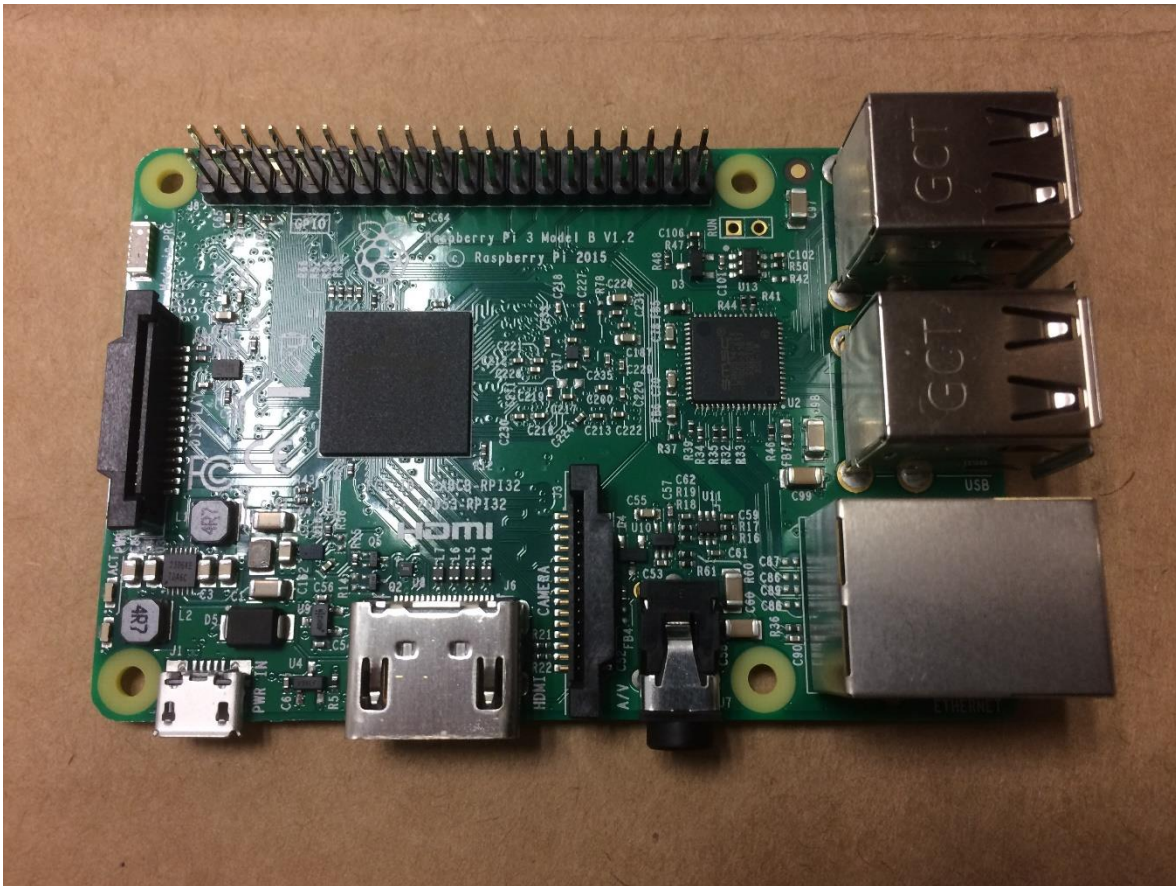


Figura 2.10 Raspberry Pi 3 B

---

## 2.7.1 Características Generales

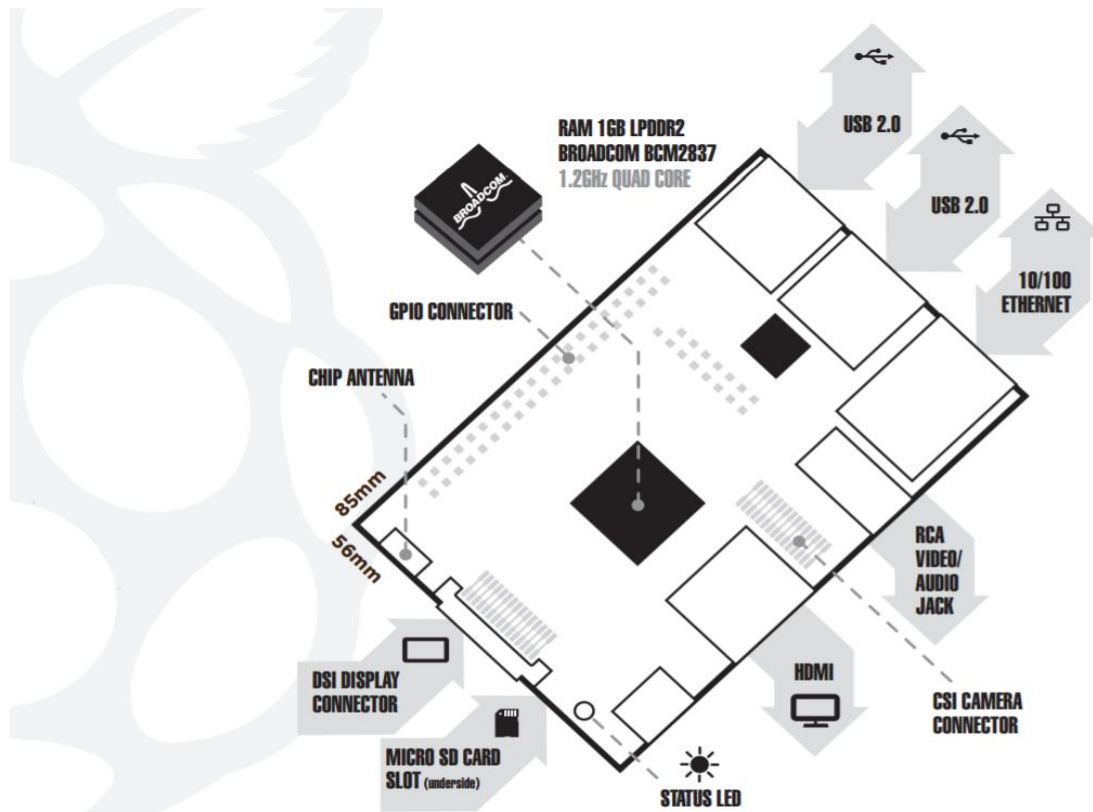


Figura 2.11 Características internas de la Raspberry Pi 3 B

### Procesador

Chip Broadcom BCM2837

1.2GHz Quad-Core ARM Cortex-A53

802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Bluetooth Classic y LE)

### GPU

Dual Core VideoCore IV® Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware y 1080p30 H.264 de alto perfil para descodificar.

Capaz de 1Gpixel / s, 1.5Gtexel / s o 24GFLOPs con filtro de textura e Infraestructura de DMA

---

## **Memoria RAM**

1GB LPDDR2

## **Dimensiones**

85 x 56 x 17mm

## **Alimentación**

Micro USB socket 5V1, 2.5A

## **Conectores**

Conector Ethernet 10/100 BaseT Ethernet

## **Salida de vídeo**

HDMI (rev 1.3 & 1.4 Composite RCA (PAL y NTSC)

## **Salida de audio**

Conector de 3,5 mm, HDMI

## **USB**

3 Conectores USB 2.0

## **GPIO**

Conector de 40 pines 2,54 mm (100 mil) cabezal de expansión: 2x20 tira. Proporcionando 27 clavijas GPIO así como líneas de suministro de +3,3 V, +5 V y GND

## **Conector de la cámara**

Interfaz serie de la cámara MIPI de 15 contactos (CSI-2)

## **Conector de Display**

Display Interfaz serie (DSI) Conector de cable plano flexible de 15 vías con dos datos Carriles y un carril de reloj.

## **Ranura para tarjeta de memoria**

Push / pull Micro SDIO

---

### **Beneficios clave**

- Formato consistente de la tarjeta
- Procesamiento 10x más rápido
- Conectividad añadida

El Raspberry Pi 3 Modelo B es la tercera generación de tarjetas Raspberry pi. Reemplazó al modelo anterior Raspberry Pi 2 B en febrero de 2016, contado con características electrónicas más completas, el modelo 3 B nos permitirá hacer uso de su arquitectura ARM (Advanced RISC Machines) para usarla como un computador móvil con recursos necesarios para ejecutar el sistema operativo Android y así usar la aplicación de medición que se describirá posteriormente.

---

## 2.7.2 Puertos de comunicación de la Raspberry Pi 3 B

La tarjeta cuenta principalmente con dos puertos USB y uno HDMI, uno de los puertos USB será usado para la comunicación de la tarjeta con la computadora, será por medio de este puerto que se enviará la información de la medición capturada por el sistema a la interfaz visual de la computadora, el sensor TeraRanger One, el acelerómetro ADXL345 y el módulo de cámara V2 estarán conectados a los pines de comunicación de la tarjeta mientras que el slot de tarjetas micro SD será utilizado para llevar la memoria en donde se estará ejecutando el sistema operativo Android y finalmente el puerto HDMI será para tener monitorizado en una pantalla externa con puerto de esta característica el seguimiento de la medición que se esté haciendo en tiempo real.

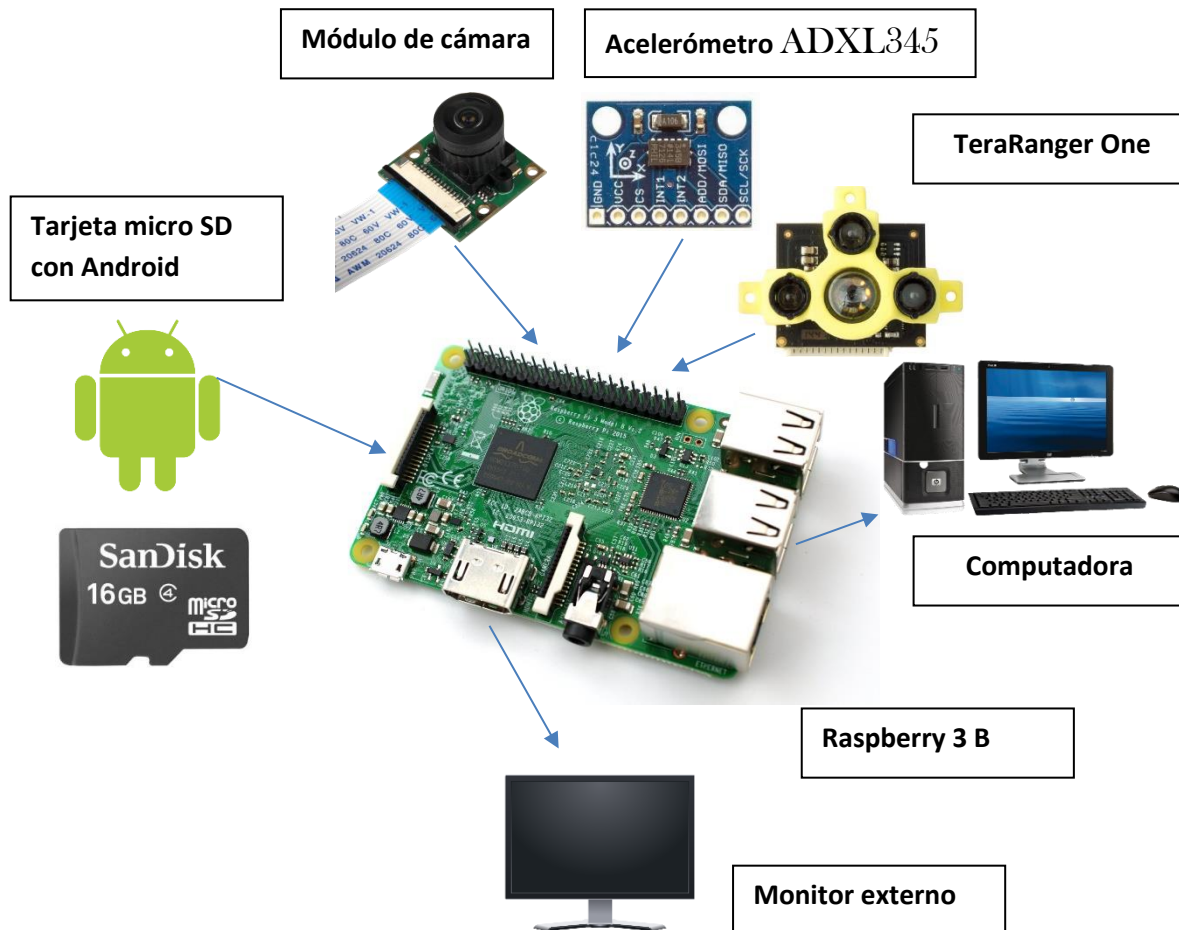


Figura 2.12 Puertos de comunicación

---

## 2.8 Ley de Cosenos

La ley de los cosenos es utilizada para calcular uno de los lados de un triángulo cuando se conocen el ángulo opuesto y los otros dos lados (figura 2.13). Esta definida por la ecuación 2.1.

$$c^2 = a^2 + b^2 - 2ab \cos \theta \quad (2.1)$$

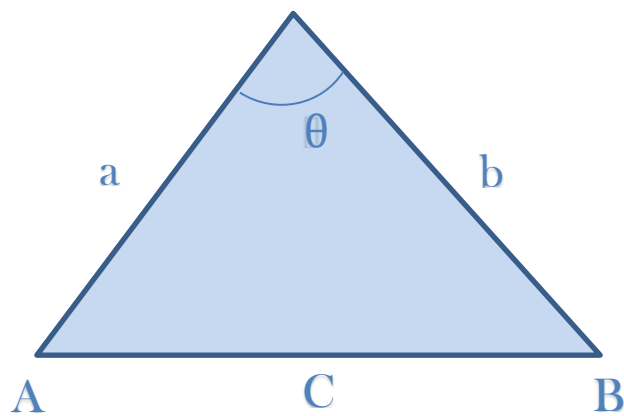


Figura 2.13 Ley de Cosenos

La ley de los cosenos es la utilizada para calcular el segmento C que se forma entre los puntos A y B (figura 2.4) marcados en la imagen capturada por el módulo de cámara V2, dentro de la librería de Matlab existe un algoritmo que aplica esta ley de cosenos a los puntos marcados A y B dentro de la aplicación de medición.

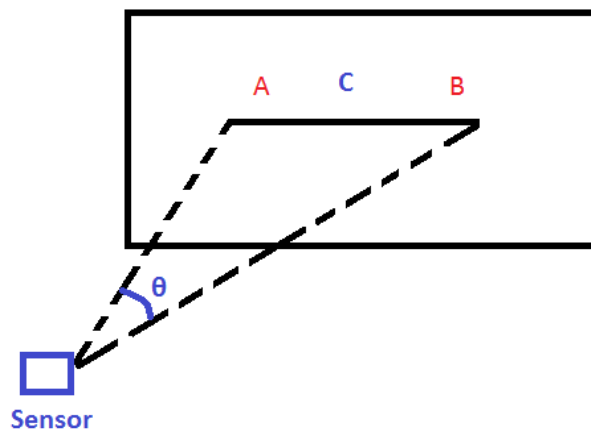
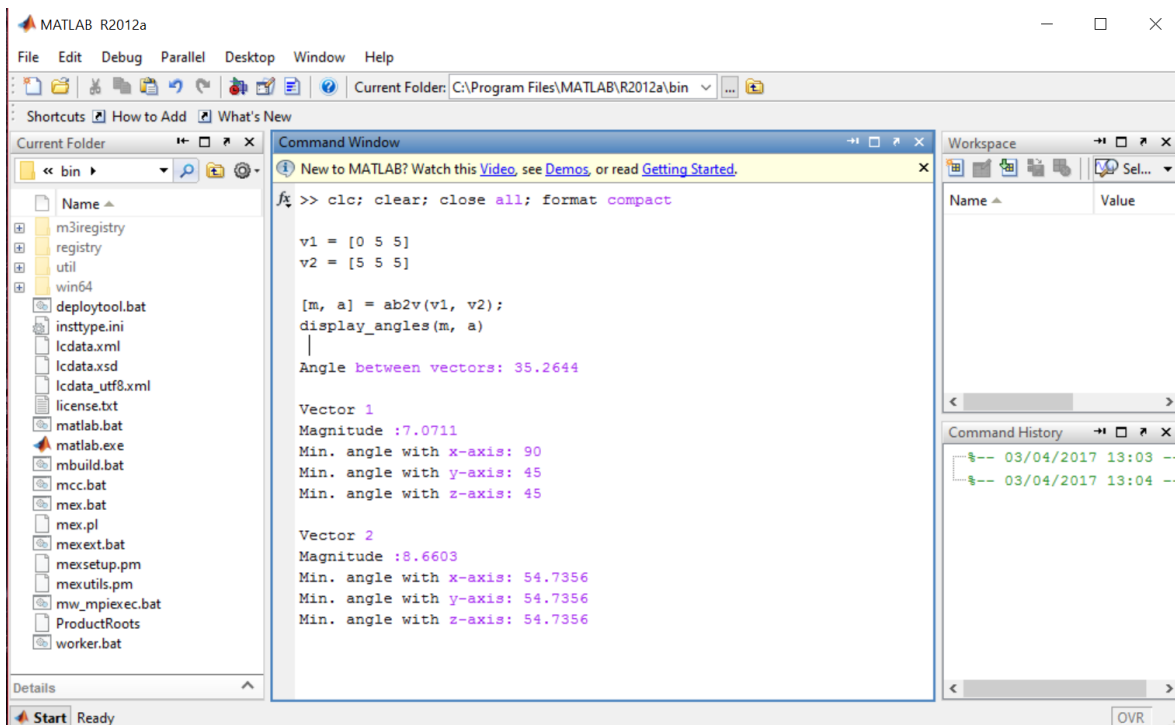


Figura 2.14 Funcionamiento del sistema

## Programa en Matlab

La siguiente simulación en Matlab (figura 2.15), hace referencia al cálculo que nos permite conocer el valor de los vectores de tres dimensiones que corresponden a los puntos A y B, así como el ángulo entre estos, pudiendo determinar el valor del segmento que se forma entre estos puntos manualmente (figura 2.16).



```
MATLAB R2012a
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Program Files\MATLAB\R2012a\bin

Shortcuts How to Add What's New
Current Folder bin
Name
m3registry
registry
util
win64
deploytool.bat
insttype.ini
lclata.xml
lclata.xsd
lclata_utf8.xml
license.txt
matlab.bat
matlab.exe
mbuild.bat
mcc.bat
mex.bat
mex.pl
mexext.bat
mexsetup.pm
mexutils.pm
mw_mpiexec.bat
ProductRoots
worker.bat

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> clc; clear; close all; format compact
v1 = [0 5 5]
v2 = [5 5 5]
[m, a] = ab2v(v1, v2);
display_angles(m, a)
Angle between vectors: 35.2644
Vector 1
Magnitude :7.0711
Min. angle with x-axis: 90
Min. angle with y-axis: 45
Min. angle with z-axis: 45
Vector 2
Magnitude :8.6603
Min. angle with x-axis: 54.7356
Min. angle with y-axis: 54.7356
Min. angle with z-axis: 54.7356

Workspace
Name Value
Command History
03/04/2017 13:03
03/04/2017 13:04
```

Figura 2.15 Programa Matlab

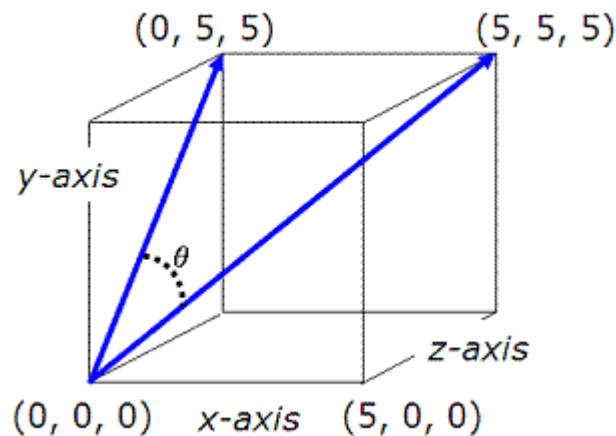


Figura 2.16 Cálculo de los puntos

---

## 2.9 Instalación de Android

El sistema operativo móvil Android es un entorno basado en el núcleo Linux. Este fue diseñado principalmente para dispositivos móviles con pantallas táctiles, como teléfonos celulares inteligentes, tablets o phablets, relojes inteligentes, televisores y automóviles. Desarrollado inicialmente por la compañía Android Inc., empresa inicialmente patrocinada por Google y que respaldó económicamente, que posteriormente fue obtenida por esta última en el año 2005. La primera versión de este (1.0 Apple Pie) fue presentado en el año 2007 junto a la fundación del Open Handset Alliance, un consorcio de compañías de hardware, software y telecomunicaciones, para avanzar tecnológicamente en los estándares abiertos de los dispositivos móviles.

Para la instalación del sistema operativo Android en su versión 4.4 (KitKat) en la placa Raspberry primeramente se obtuvo una copia de la imagen ISO de Android con licencia open source, desarrollada especialmente para el modelo Raspberry 3 B haciendo que el sistema operativo Android fluya adecuadamente de acuerdo a las especificaciones técnicas con las que cuenta la tarjeta, esta imagen se cargó en una memoria micro SD de 16 Gb (previamente formateada en el formato FAT32) con el software LIVE LINUX USB, este software adecua la imagen del sistema para que posteriormente se expanda en su instalación en la memoria ROM en la Raspberry, esta versión de Android está especialmente desarrollada para la placa, la dota de todas las funcionalidades que posee un dispositivo móvil, exceptuado los módulos electrónicos externos que requiera la tarjeta y Android.



*Figura 2.17 Android en tarjeta de memoria Micro SD*

---

## 2.10 Software Development Kit Android

La compañía Google nos ofrece el SDK (Software Development Kit Android) oficial de su sistema operativo móvil Android, que es un conjunto de drivers, herramientas de software y diversos recursos para programar y desarrollar aplicaciones para Android. El kit de desarrollo puede obtenerse de manera gratuita en el paquete Developer Tools donde además se incluye un editor IDE que es una plataforma de software formado por herramientas de programación de código abierto para crear entornos de desarrollo integrados. Con el Android SDK contamos con las herramientas necesarias para crear nuestra aplicación de medición para esta plataforma, junto con distintas APIs (Application Programming Interface) ofrecidas por Google tanto para el control de las funciones del dispositivo como para la integración de servicios, un depurador, un emulador para probar las aplicaciones antes de ser ejecutadas por el dispositivo y toda la documentación necesaria para crear y programar las aplicaciones en Android. Android está basado en Linux, utiliza una máquina virtual para ejecutar las aplicaciones, y ofrece soporte para 3G, Wi-Fi, GPS, pantallas táctiles y cualquier otro componente habitual que hoy en día cuentan todos los smartphones y tablets de cualquier gama.



# Android SDK

*Figura 2.18 Logo Android SDK*

---

### 2.10.1 Editor IDE Eclipse

El editor Eclipse es una plataforma de software formado por un conjunto diverso de herramientas de programación de código abierto (Open Source) multiplataforma para desarrollar lo que se denomina "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores web. Esta plataforma de software, ha sido usada principalmente para desarrollar entornos de desarrollo integrados IDE (Integrated Development Environment), como el IDE del lenguaje Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse y que son usados comúnmente para desarrollar el mismo software de Eclipse. Cabe mencionar que también se puede usar para otros tipos de aplicaciones cliente, como Azureus, BitTorrent o Android.

Eclipse fue creado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. Eclipse fue liberado originalmente bajo la Common Public License, pero después fue re-licenciado bajo la Eclipse Public License. La Free Software Foundation ha dicho que ambas licencias son licencias de software libre, pero son incompatibles con Licencia pública general de GNU y GPL.

[4] Los entornos de desarrollo integrado IDE son parte del software fundamental para desarrollar un APK (Android Application Package) porque la estructura y el lenguaje de programación de estos son de forma fácil y práctica de utilizar, la existencia del IDE Eclipse estaba incluida dentro del SDK antiguo de Android posteriormente fue reemplazado por lo que es hoy Android Studio, que en conjunto con otras herramientas de programación y edición de aplicaciones nos brinda un uso y manejo de este tipo de software para la creación de aplicaciones que se describirá en la sección de desarrollo de la aplicación.

---

## 2.10.2 Android Studio

Habiendo explicado lo que es un IDE y entendiendo que son los entornos de programación dónde se desarrolla paso a paso una aplicación, en este caso para la plataforma Android, describiremos la evolución de lo que el SDK de google se ha convertido a día de hoy, el IDE oficial para desarrollar en esta plataforma es Android Studio, que proporciona las herramientas de programación más rápidas para la creación de aplicaciones en todos los tipos de dispositivos Android. La edición de códigos de primer nivel, la depuración, las herramientas de rendimiento, un sistema de compilación flexible y un sistema instantáneo de compilación e implementación que nos permiten desarrollar de forma más sencilla aplicaciones únicas y de alta calidad. Android Studio fue anunciado el día 16 de mayo del año 2013 en la conferencia de Google I/O, y reemplazó al editor IDE Eclipse que era el entorno oficial para el desarrollo de aplicaciones de Android. La primera versión estable fue publicada en diciembre de 2014.

Android Studio está basado en el software IntelliJ IDEA de JetBrains, y está publicado de manera gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Este entorno de programación es el utilizado para el desarrollo pleno de nuestra aplicación, provee las librerías y herramientas de software que se adaptan a nuestras necesidades, por medio de librerías internas y externas programadas dentro de la aplicación se procesan las distintas etapas del sistema para obtener una medición, en la etapa inicial los valores binarios codificados que genera el sensor entran a la placa y son procesados convirtiéndolos en un valor numérico, en la siguiente etapa son agregados a la imagen capturada de la superficie, finalmente por medio de la librería de Matlab y la lectura del acelerómetro se calcula la distancia entre dos puntos fijos teniendo finalmente una medición.



*Figura 2.19 Logo Android Studio*

---

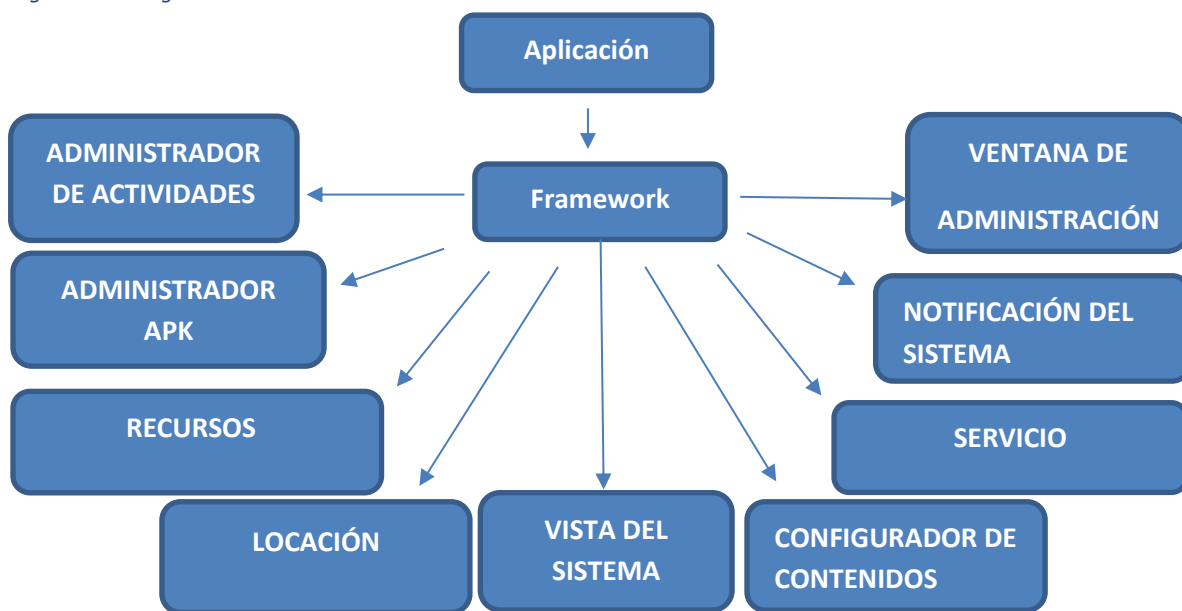
## 2.11 Desarrollo de la Aplicación

La aplicación que desarrollamos toma los valores obtenidos por el sensor TeraRanger One para decodificarlos y unirlos a los vectores marcados de la imagen obtenida por el módulo de cámara V2 en la Raspberry después se calcula la distancia entre los vectores marcados por medio de la librería de Matlab y la lectura del acelerómetro ADXL345 para darnos una medición. Antes de empezar a programar en Android Studio hay que tener en cuenta los componentes primordiales que hacen a una aplicación APK, esencialmente el framework, las librerías y el kernel de Linux.

### Framework

El framework es la base de programación que atiende a sus bloques descendientes manejándolos de una forma estructural o en cascada (figura 2.20), posibilitando una respuesta inmediata ante las necesidades de cada uno de sus miembros, o en secciones de la aplicación, satisfaciendo así las necesidades más comunes del programador en este caso nosotros. En el framework de Android tenemos los principales bloques de la aplicación, cada uno tiene una función específica y cada uno está conectado de forma directa al framework para que administre los recursos y necesidades de cada bloque, entre los principales bloques tenemos los administradores de actividades, ventana, APK (aplicación), recursos, locación, vista del sistema, configurador de contenidos, servicio, y notificación del sistema.

Figura 2.20 Diagrama Framework



---

## Librerías

Las librerías en Android son el conjunto de bloques de implementación funcionales, codificadas en un lenguaje de programación parecido a Java, que ofrecen una interfaz bien definida para la funcionalidad que se solicita (figura 2.21). A diferencia de un programa ejecutable, el comportamiento que implementa una librería es el de no esperar ser utilizada de forma autónoma, un programa ejecutable sí, el fin de una librería es ser utilizada por otros programas, independientes y de forma simultánea. Por otra parte, el comportamiento de una librería no tiene por qué diferenciarse demasiado del que pudiera especificarse en un programa. <sup>[5]</sup>Una librería puede requerir de otras para funcionar, pues el comportamiento que definen refina, o altera, el comportamiento de las librerías originales o bien las hace disponibles para otra tarea programada, las librerías básicas del núcleo Android son SQ Lite, WebKit, SSL entre otras.

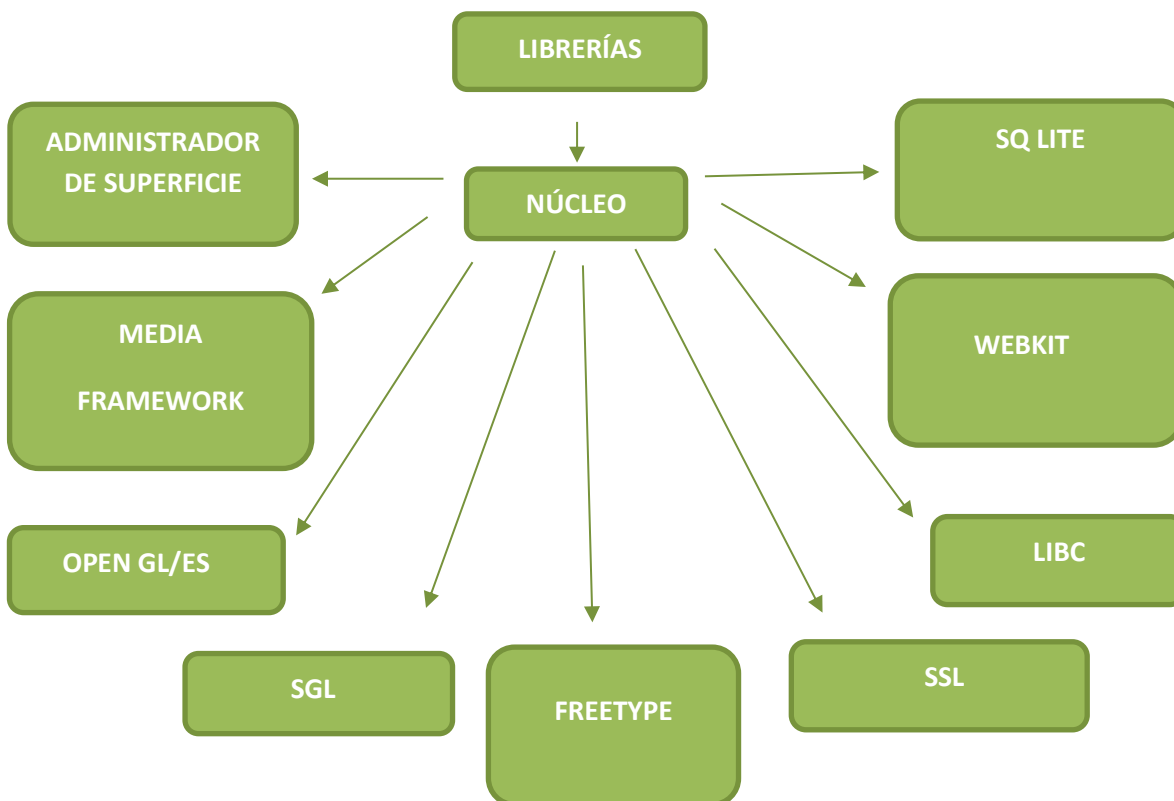


Figura 2.21 Diagrama Librerías

---

## Kernel Linux

El kernel de Linux es un kernel monolítico similar al sistema operativo de Unix, sus características principales es el soporte de la multitarea real preventiva tanto en modo usuario como en modo kernel, su memoria virtual, librerías compartidas, carga de demanda, copia compartida de escritura ejecutables, gestión de memoria, el conjunto de protocolos de enhebrado e internet (figura 2.22).

El kernel maneja y administra los controladores de dispositivos (drivers) y las extensiones del núcleo se ejecutan en el espacio del núcleo, con acceso completo al hardware, aunque algunas excepciones se ejecutan en el espacio de usuario, por ejemplo, sistemas de archivos basados en FUSE / CUSE y partes de UIO. El sistema gráfico que la mayoría de la personas usa con Linux no se ejecuta dentro del kernel. A diferencia de los núcleos monolíticos estándar, los controladores de dispositivo se configuran fácilmente como módulos y se cargan o descargan mientras el sistema se está ejecutando. Además, a diferencia de los núcleos monolíticos estándar, los controladores de dispositivo se pueden evitar en ciertas condiciones; Esta característica se agregó para manejar las interrupciones de hardware correctamente y para soportar mejor el multiprocesamiento simétrico. Por elección, el kernel de Linux en Android no tiene interfaz de kernel binario.

[6]La configuración correcta del Kernel en Android nos permite controlar los bloques de drivers y los bloques del núcleo de las librerías, siendo este fundamental para el correcto funcionamiento del sistema operativo, sus aplicaciones y su hardware (figura 2.23).

---

## Mapa del Kernel Linux

Figura 2.22 Diagrama Kernel Linux

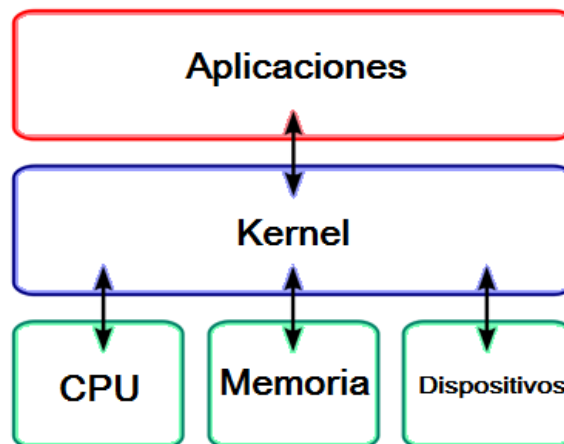
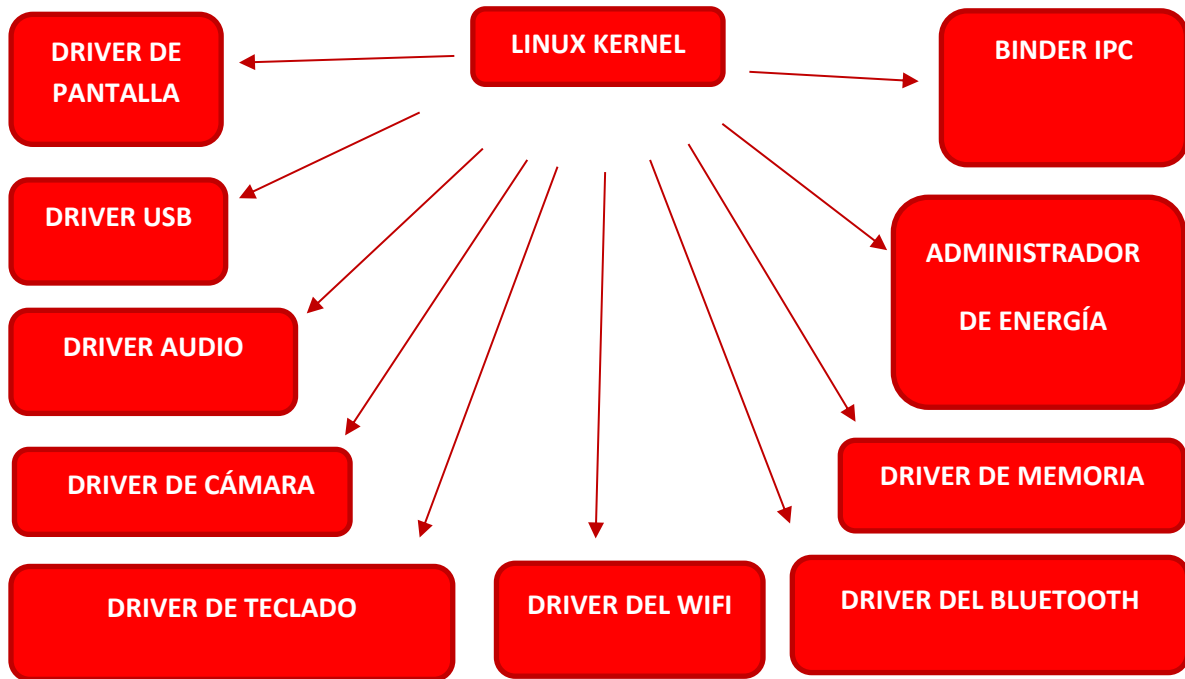


Figura 2.23 Distribución

---

Entendidos los conceptos de programación que hacen a una APK, procedimos a crear la aplicación de medición en el Android Studio, para eso en la ventana del programa creamos el nuevo proyecto “MEDICION” (figura 2.24).

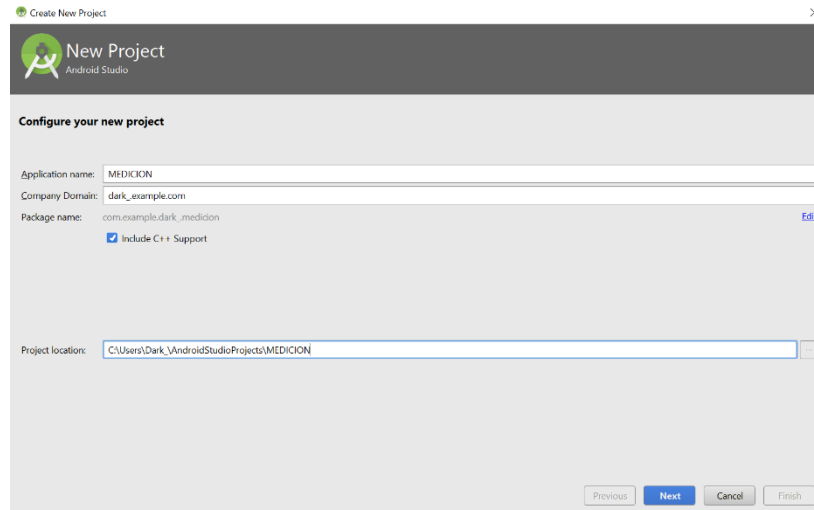


Figura 2.24 Creación de nuevo proyecto

Asignamos la versión de Android que tendrá la compatibilidad con nuestra aplicación, así como los APIs que usaremos (figura 2.25).

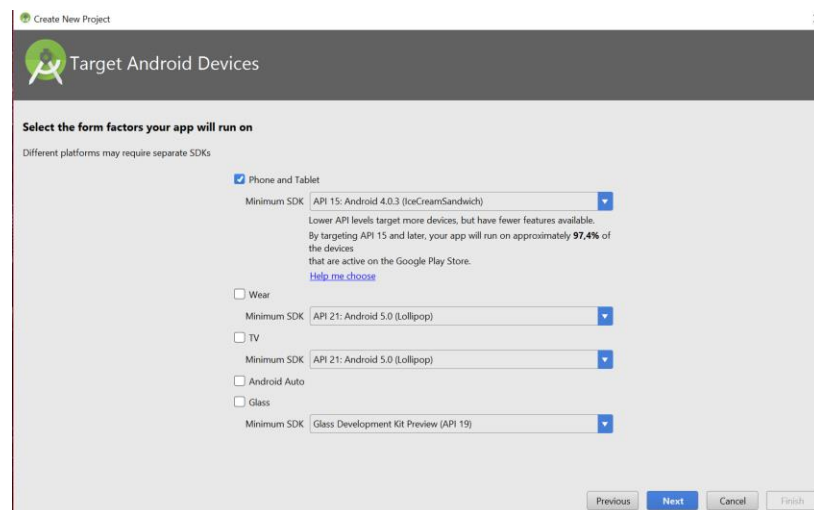


Figura 2.25 Configuración de compatibilidad

---

Nuestra placa Raspberry será utilizada como un dispositivo móvil así que usamos la configuración de actividad básica (figura 2.26).

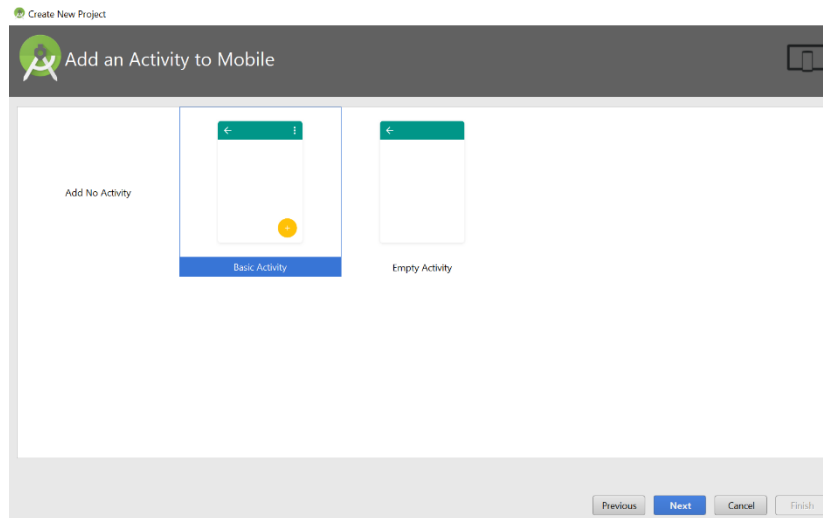


Figura 2.26 Actividad básica

Se configura la ventana de actividad principal figura (2.27).

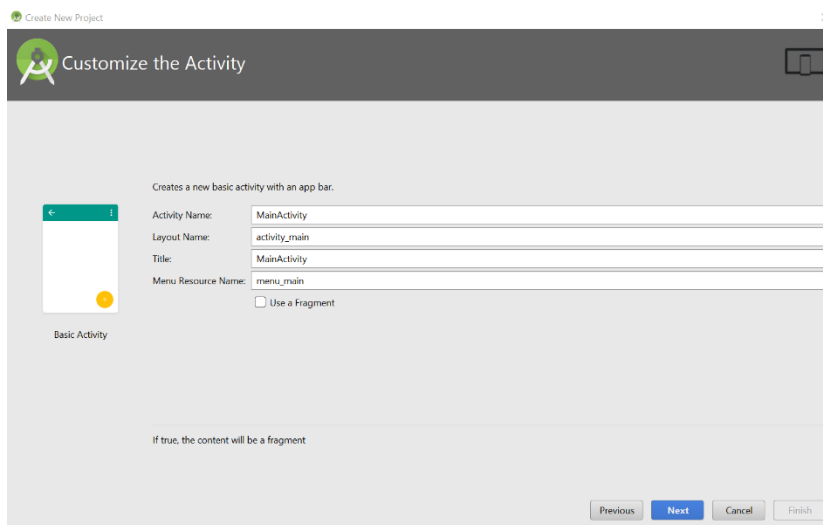


Figura 2.27 Actividad principal

Como usaremos librerías externas, damos soporte al lenguaje de C++ (figura 2.28).

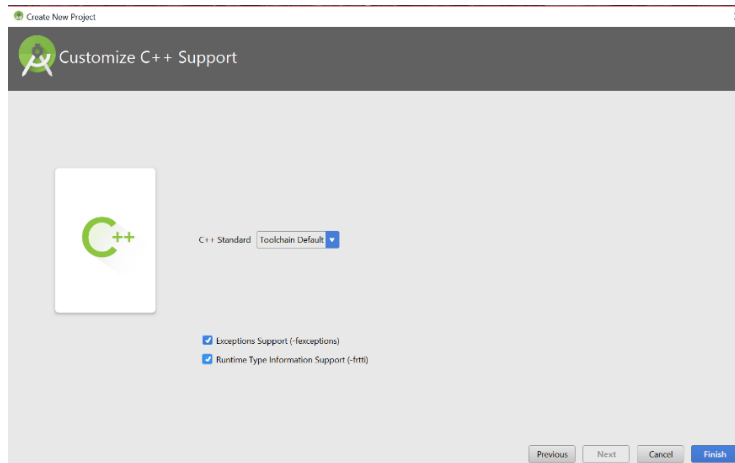


Figura 2.28 Soporte para lenguaje C++

Una vez en la interfaz configuramos las librerías externas, que son los módulos de procesamiento de imagen, la librería de decodificación, la librería del acelerómetro ADXL345, la librería de unificación y la librería de Matlab (figura 2.29).

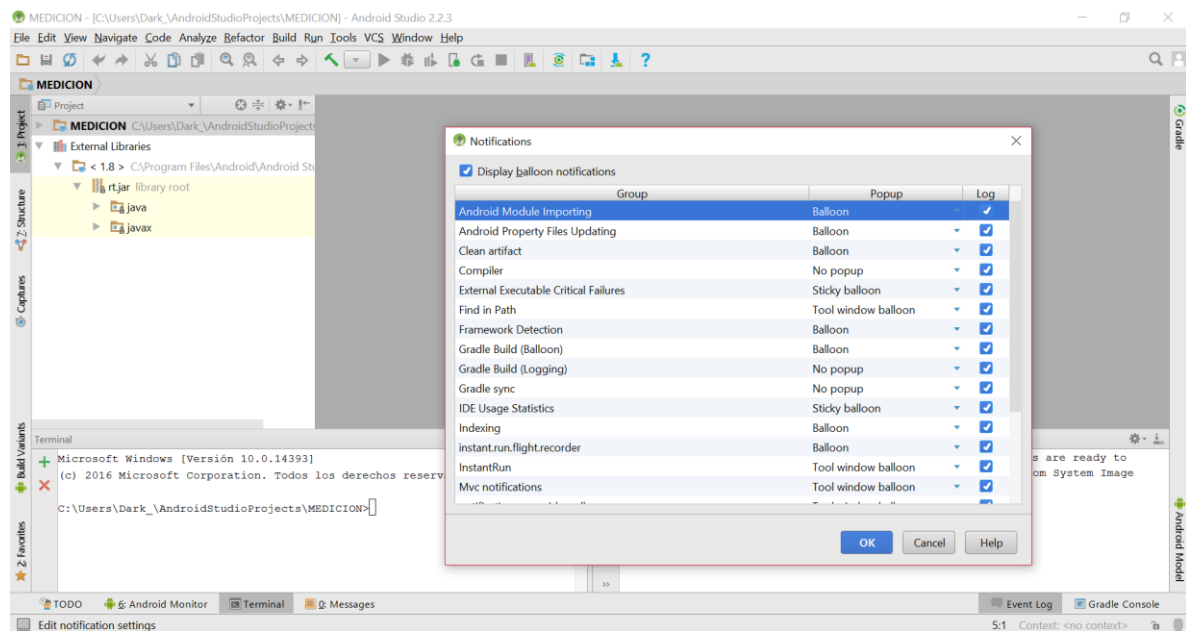
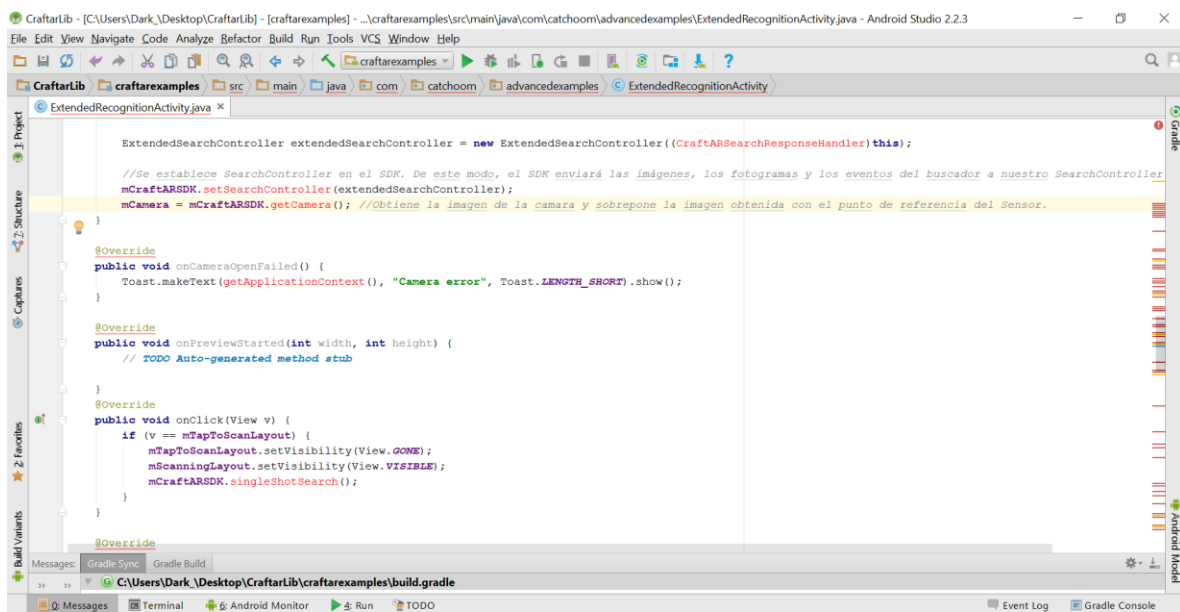


Figura 2.29 Módulos de procesamiento de imagen

Desarrollamos el código de la aplicación que realiza la medición (figura 2.30); Éste se basa en la comparación de los valores obtenidos (vectores) por las librerías de procesamiento de imagen a través del módulo electrónico de cámara V2 en la Raspberry; Cuando esta genera una imagen “normal” dentro de la aplicación se genera otra en mapa de bits, dónde los vectores de referencia y término de la imagen son “marcados” , después la información obtenida por el sensor TeraRanger One se une a estos puntos marcados, la aplicación procesa y une esta información a través de un módulo de unificación basado en lenguaje JAVA que analiza los vectores marcados en la imagen de la cámara y posteriormente agrega la información del sensor infrarrojo de manera gráfica entre el valor de la medición inicial y final en la imagen obtenida, finalmente usando la librería de MATLAB y los valores obtenidos por el acelerómetro ADXL345 calculamos la distancia entre estos puntos por ley de cosenos.



```
ExtendedSearchController extendedSearchController = new ExtendedSearchController((CraftARSearchResponseHandler) this);

//Se establece SearchController en el SDK. De este modo, el SDK enviará las imágenes, los fotogramas y los eventos del buscador a nuestro SearchController
mCraftARSDK.setSearchController(extendedSearchController);
mCamera = mCraftARSDK.getCamera(); //Obtiene la imagen de la camara y sobrepone la imagen obtenida con el punto de referencia del Sensor.
}

@Override
public void onCameraOpenFailed() {
    Toast.makeText(getApplicationContext(), "Camera error", Toast.LENGTH_SHORT).show();
}

@Override
public void onPreviewStarted(int width, int height) {
    // TODO Auto-generated method stub
}

@Override
public void onClick(View v) {
    if (v == mMapToScanLayout) {
        mMapToScanLayout.setVisibility(View.GONE);
        mScanningLayout.setVisibility(View.VISIBLE);
        mCraftARSDK.singleShotSearch();
    }
}

@Override
```

Figura 2.30 Código de compilación

Ahora para visualizar nuestra aplicación como una APK ejecutable configuramos el emulador virtual, en este caso pondremos una Tablet "Nexus 7" (figura 2.31).

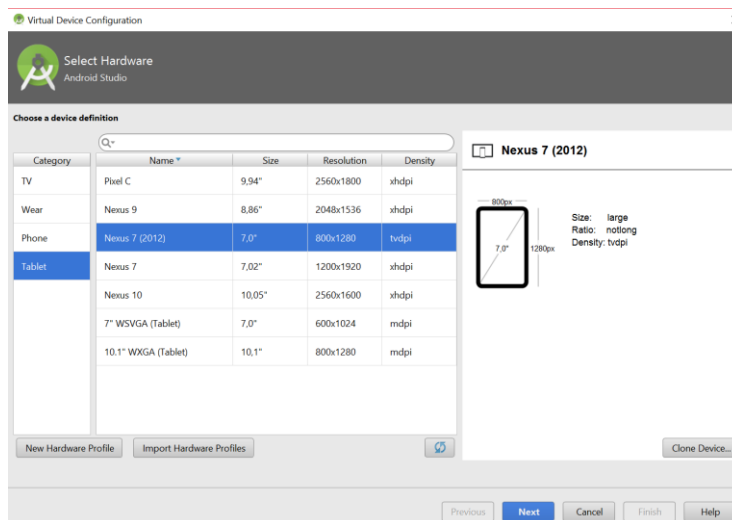


Figura 2.31 Selección de emulador

Asignamos la memoria RAM del emulador a 1 Gb (figura 2.32).

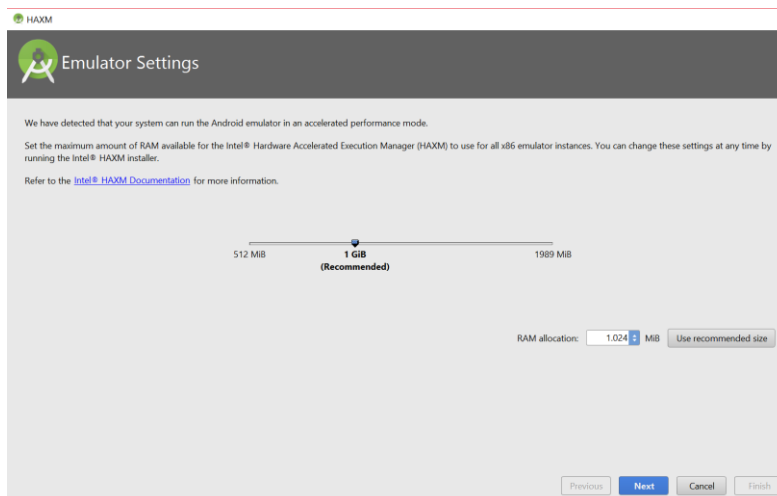


Figura 2.32 Configuración de memoria RAM

Asignamos la versión de Android previamente descargada "KitKat 4.4" (figura 2.33).

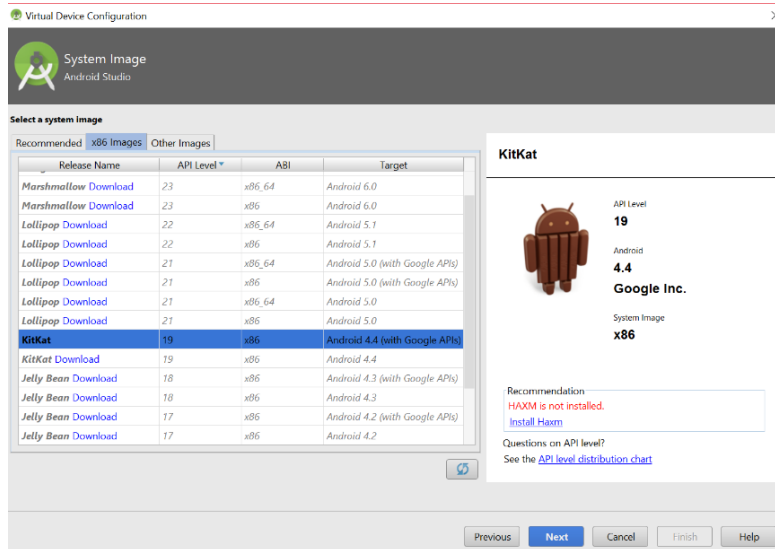


Figura 2.33 Versión de Android

Iniciamos el emulador virtual para instalar y ver la aplicación (figura 2.34).

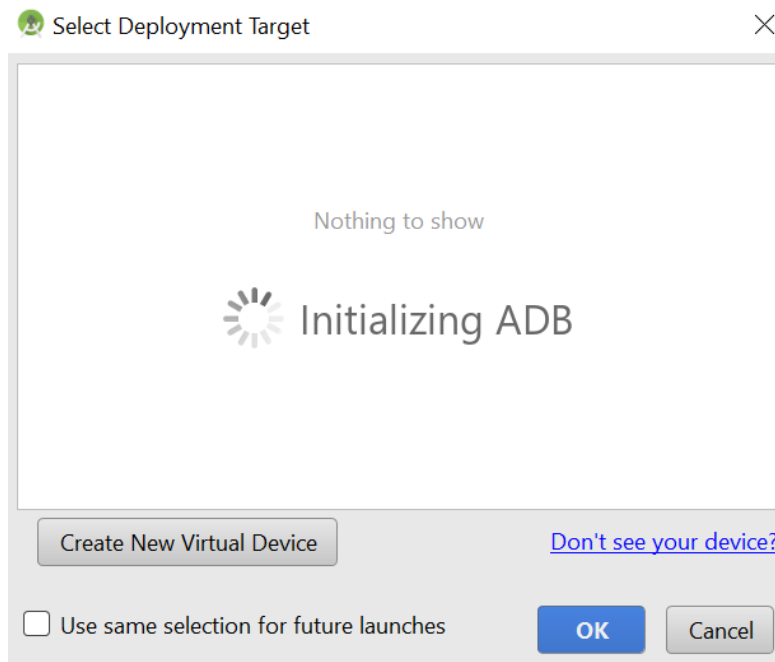


Figura 2.34 Inicio de la emulación

---

Una vez instalada vemos el emulador mostrando la aplicación terminada (figura 2.25).

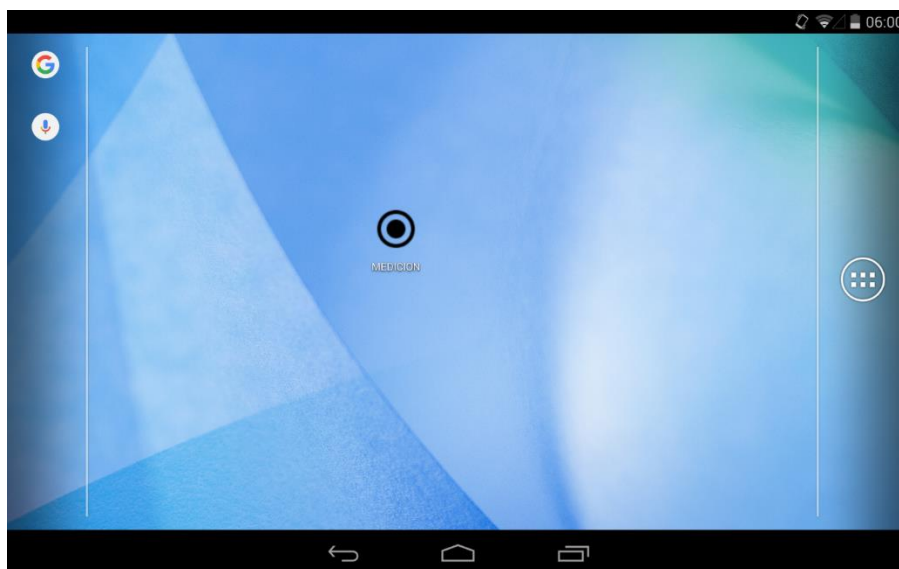


Figura 2.35 Aplicación instalada

### 2.11.1 Diseño gráfico

El diseño gráfico de la aplicación fue realizado en Android Studio, con las herramientas para la interfaz gráfica, los iconos así como los botones virtuales se crearon en Photoshop (figura 2.36); este software de diseño y edición de imágenes posee una gran variedad de herramientas de edición gráfica, en este se llevó a cabo la creación de los iconos, líneas y figuras en el formato <sup>[7]</sup>PNG (Portable Network Graphics) que es la extensión de imagen utilizada por Android para el desarrollo de las aplicaciones.



Figura 2.36 Gráficos diseñados

## Capítulo 3

### 3.1 Estructura de Funcionamiento

El siguiente diagrama representa el funcionamiento interno del SMPF.

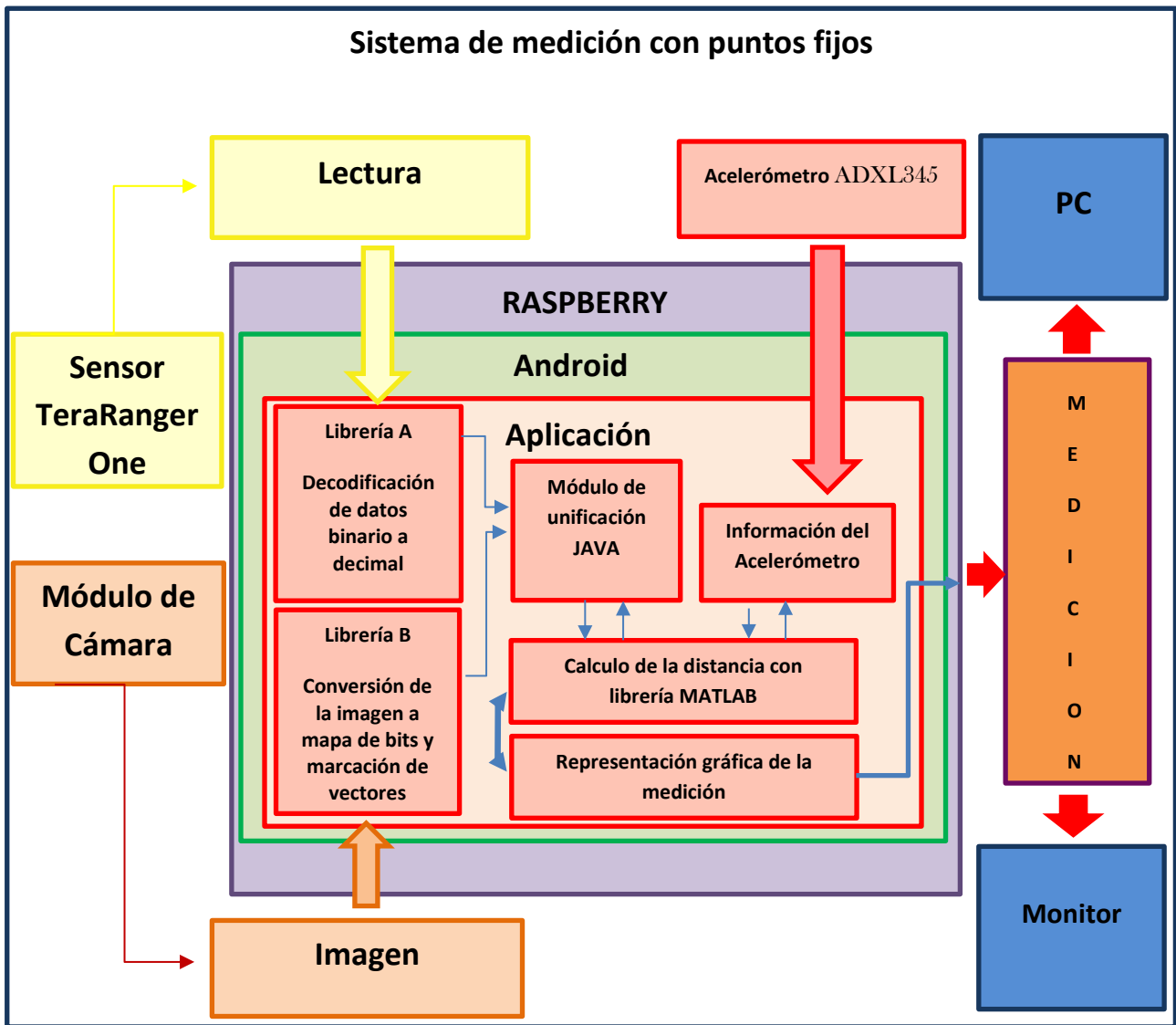


Figura 3.1 Diagrama sistema de medición con puntos fijos

---

### 3.2 Sensor de Distancia

El sensor de distancia TeraRanger One mide la distancia entre el emisor del haz infrarrojo y una superficie, las medidas obtenidas por este son independientes entre los puntos A y B; luego son tomadas como variables de entrada para su proceso en la Raspberry. La precisión del sensor está determinada por la tecnología ToF la cuál utiliza impulsos electromagnéticos que se emiten durante un tiempo muy corto y se reflejan hacia la cámara del sensor para calcular la distancia, en este caso el pulso infrarrojo resultante del sensor es dirigido hacia la superficie y es reflejado hacia el campo de visión de su cámara, la lente reúne la luz reflejada en el sensor y la matriz de plano focal. Dependiendo de la distancia, la señal entrante experimenta un retardo (figura 3.2).

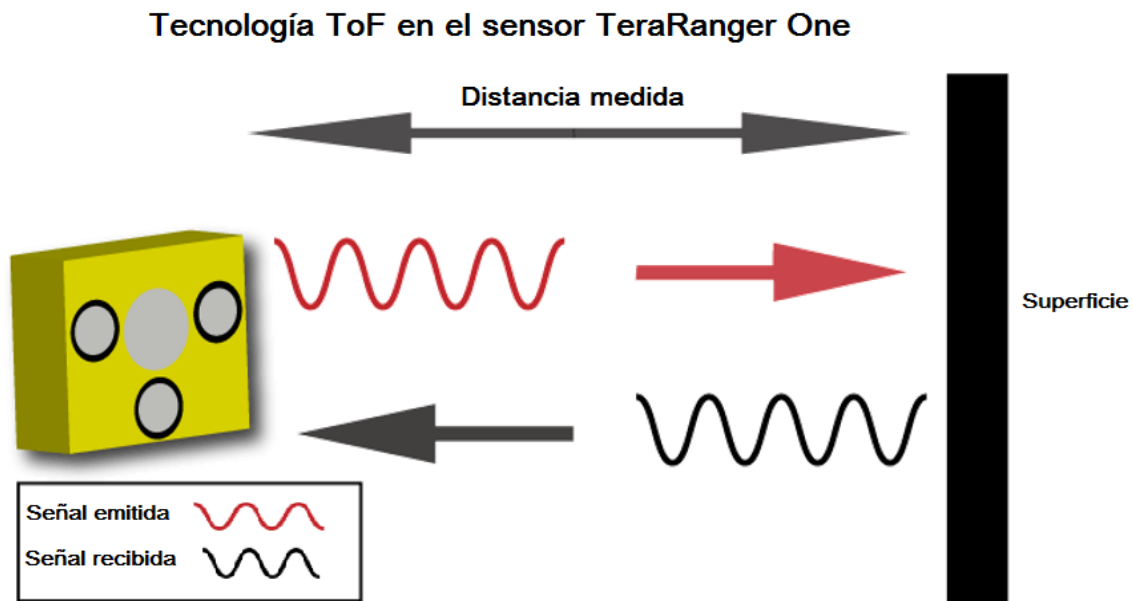


Figura 3.2 Funcionamiento del sensor TeraRanger One

---

### 3.2.2 Programación

El sensor emite una señal infrarroja binaria codificada, esta señal entra en la tarjeta Raspberry Pi 3 donde es decodificada e interpretada, para que se lleve a cabo esta acción se utiliza un algoritmo dentro de la librería Ksoap2 que realiza esta conversión por Soap (Object Access Protocol). El sensor es programado para que pueda sensar distancias entre los 0 mm y los 14 metros, este rango nos brinda la posibilidad de tener una percepción de profundidad considerable por lo que las mediciones podrán abarcar una distancia considerable (figura 3.3). Para esto la interfaz Hterm nos brinda la posibilidad de programar el rango del sensor infrarrojo, una vez realizada la comunicación con el sensor a través del módulo de comunicación bidireccional, comunicáramos el sensor a través del puerto COM (X - es el número de puerto USB que puede variar dependiendo la conexión del usuario), y procederemos a programar la distancia de percepción del sensor con el siguiente código, dónde cada línea esta explicada de forma precisa.

```
long distancia;
long tiempo;
void setup(){
  Serial.begin(9600);
  pinMode(13, OUTPUT); /*activación del pin 13 como salida: para el haz infrarrojo*/
  pinMode(12, INPUT); /*activación del pin 12 como entrada: tiempo del rebote del haz*/
}

void loop(){
  digitalWrite(9,LOW); /* Para la estabilización del sensor*/
  delayMicroseconds(5);
  digitalWrite(9, HIGH); /* envío del haz infrarrojo*/
  delayMicroseconds(10);
  tiempo=pulseIn(8, HIGH); /* Función para medir la longitud del pulso entrante. Mide el tiempo que transcurrido entre el envío
del haz infrarrojo y cuando el sensor recibe el rebote, es decir: desde que el pin 12 empieza a recibir el rebote, HIGH, hasta que
deja de hacerlo, LOW, la longitud del pulso entrante*/
  distancia= int(0.017*tiempo); /*fórmula para calcular la distancia obteniendo un valor entero*/
  /*Monitorización en milímetros por el monitor serial*/
  Serial.println("Distancia ");
  Serial.println(distancia);
  Serial.println(" mm");
  delay(1000);
}
```

*Figura 3.3 Código de programación*

Una vez programado, el código se ‘graba’ en el sensor desde la interfaz de Hterm a través del módulo de comunicación bidireccional.

Para evitar la diafonía y el efecto Doppler el sensor es programado desde la interfaz Hterm para que codifique los datos binarios, así no hay pérdida de información por los cambios de luz que se presentaran en ambientes externos o movimientos que pueda tener el sistema de medición, esta opción viene marcada por la pestaña Asend, dónde tiene los ajustes de codificación de datos (figura 3.4).

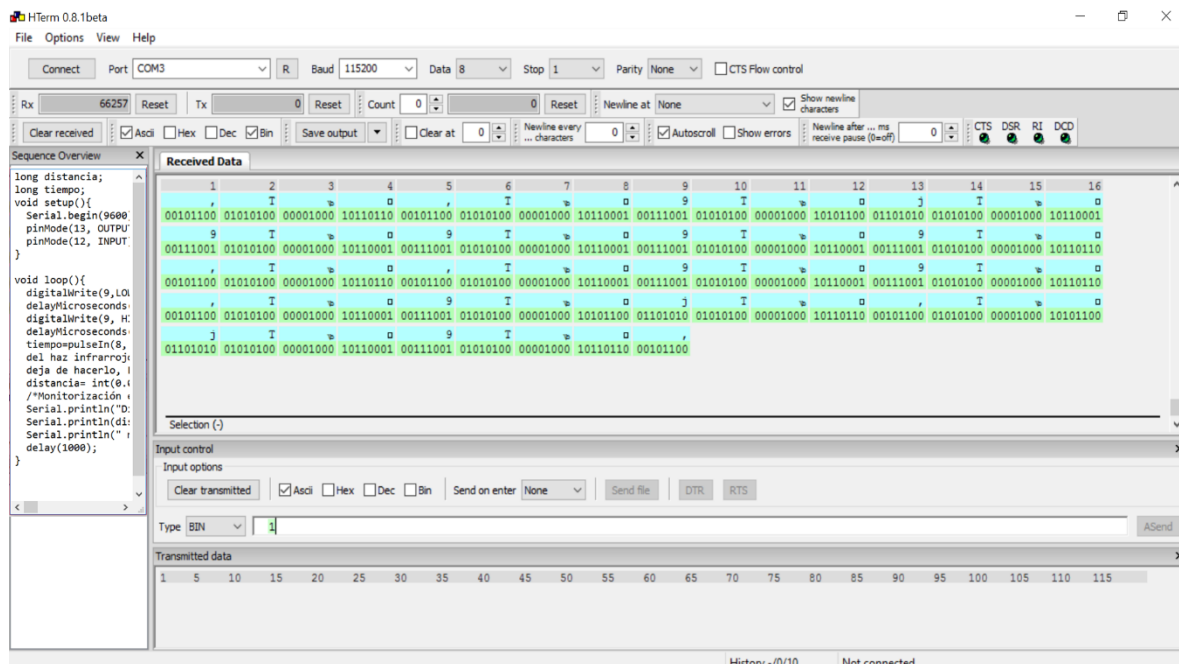


Figura 3.4 Datos binarios en Hterm

Estos datos son enviados a la Raspberry Pi a través de los pines de comunicación, dónde son recibidos y procesados por las librerías de decodificación, éstas convierten los datos binarios a un valor decimal, después de esta conversión ingresan al módulo Java dónde los valores son sumados a la imagen de mapa de bits que se describe a continuación.

---

### 3.3 Cálculo del ángulo con el acelerómetro ADXL345

Para demostrar visualmente el cálculo del ángulo que realiza el acelerómetro ADXL345 se realizó un código y un circuito simple en un Arduino Nano (figura 3.5); los datos del acelerómetro son enviados al Arduino dónde los procesa el algoritmo de programación en este y después los datos pasan a una computadora por medio de un cable USB dónde una interfaz gráfica basada en c# recibe los datos y representa el ángulo de movimiento que tenga el acelerómetro en un círculo, dónde la apertura del vértice representa el ángulo entre los puntos A y B del sistema de medición con puntos fijos (figuras 3.6 y 3.7).

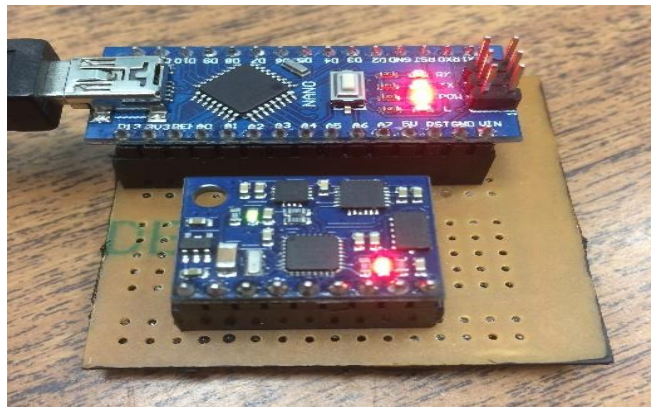


Figura 3.5 Acelerómetro y Arduino Nano

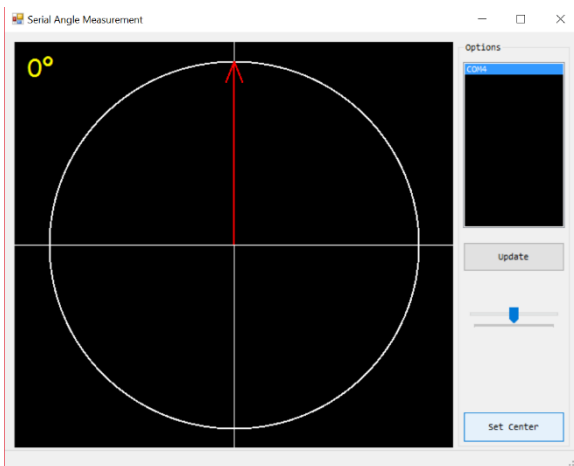


Figura 3.6 Interfaz gráfica con referencia en 0 grados

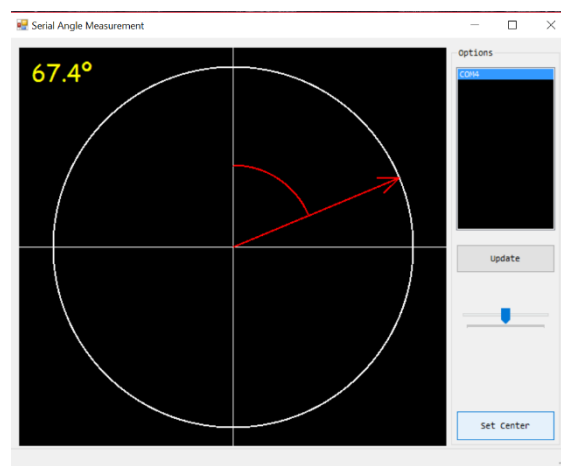


Figura 3.7 Interfaz Gráfica con ángulo medido

---

### 3.4 Conversión de la imagen

En el puerto de la Raspberry recibimos los valores binarios codificados del sensor TeraRanger One; [8] éstos son decodificados y se suman a los vectores de la imagen de mapa de bits (bmp). La imagen bmp es un formato de imagen donde se descompone la imagen original en vectores (figura 3.8); esta se genera a partir de la imagen original obtenida por el módulo de cámara V2, al realizar una fotografía esta es procesada por la librería de procesamiento de imagen donde transforma la imagen de tal manera que se marcan los vectores A y B de referencia de la superficie fotografiada (figura 3.9), haciendo esto los valores del sensor TeraRanger One son sumados a los vectores marcados, posteriormente por medio de un algoritmo matemático y la información proporcionada por el acelerómetro ADXL345 en la tarjeta se concluye una distancia entre estos puntos de referencia.

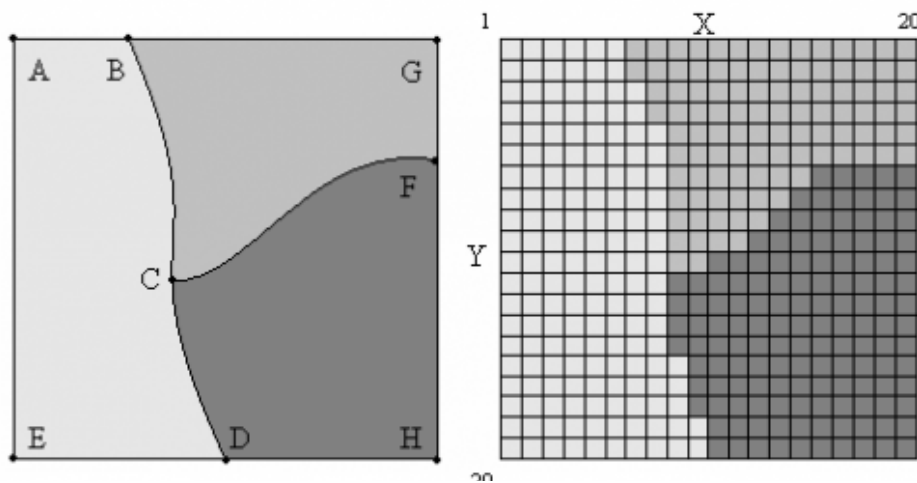


Figura 3.8 Comparación de imagen vectorial con imagen de mapa de bits

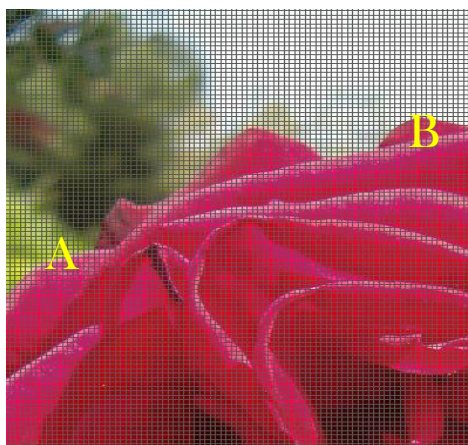


Figura 3.9 Imagen bmp con los puntos A y B

---

### 3.5 Cálculo de la distancia final

Para obtener la distancia entre los puntos A y B que fueron marcados dentro de la imagen bmp, los datos "sumados" que se obtienen previamente del algoritmo de unificación en Java son introducidos a la librería de Matlab, el algoritmo matemático que maneja esta librería está basado en la ley de los cosenos, al tener la distancia entre el sensor con los puntos A y B se genera automáticamente el segmento C, formando un triángulo, el algoritmo calcula el valor de C con los valores recibidos y con el valor del ángulo en el vértice del sensor (figura 3.10), el valor del ángulo es proporcionado por el acelerómetro ADXL345 que está conectado en la tarjeta Raspberry , así al final de procesar esta información nos arroja el valor del segmento C , que es la distancia exacta entre los puntos A y B (figura 3.11).

*\*Consultar programa en el apéndice D pág. 58*

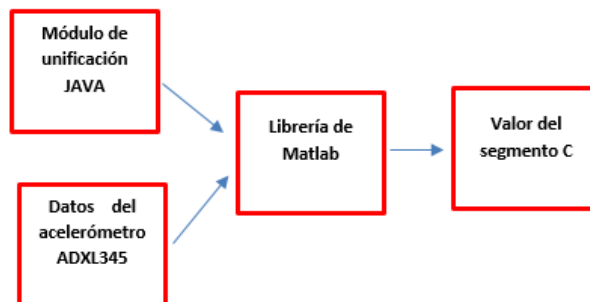


Figura 3.10 Diagrama del procesamiento de la información

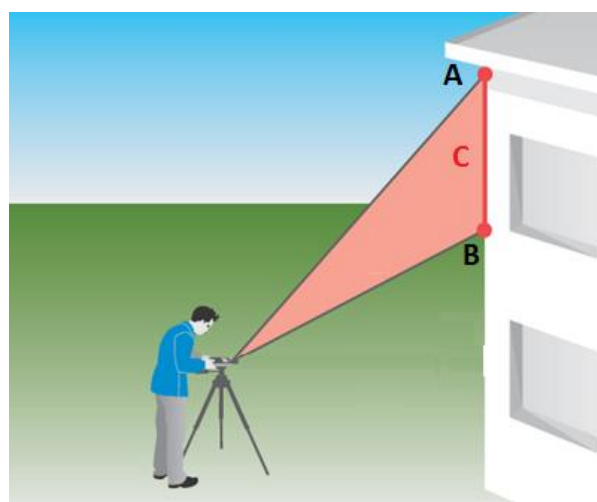


Figura 3.11 Funcionamiento del sistema

---

### 3.6 Unificación de los Sistemas

Para la unificación de la información proporcionada por el sensor TeraRanger One y el módulo de cámara V2 con la Raspberry - Android, el procesamiento de información se hace a través de la unión de los valores de las librerías de procesamiento de imagen CraftArLib y las librerías núcleo armeabi-v7a a través de un módulo de unificación basado en lenguaje JAVA, este “suma” la información recopilada por ambas librerías y envía esta información a la librería de MATLAB dónde procesa esta información con el ángulo proporcionado por un acelerómetro ADXL345 y determina la distancia en la imagen capturada por ley de cosenos, dejando el valor del segmento C entre los puntos A y B como la medida deseada. Cada uno de los dispositivos usados para el SMPF se montaron en una carcasa prototipo, donde se realizan las mediciones a distancia (figura 3.12).

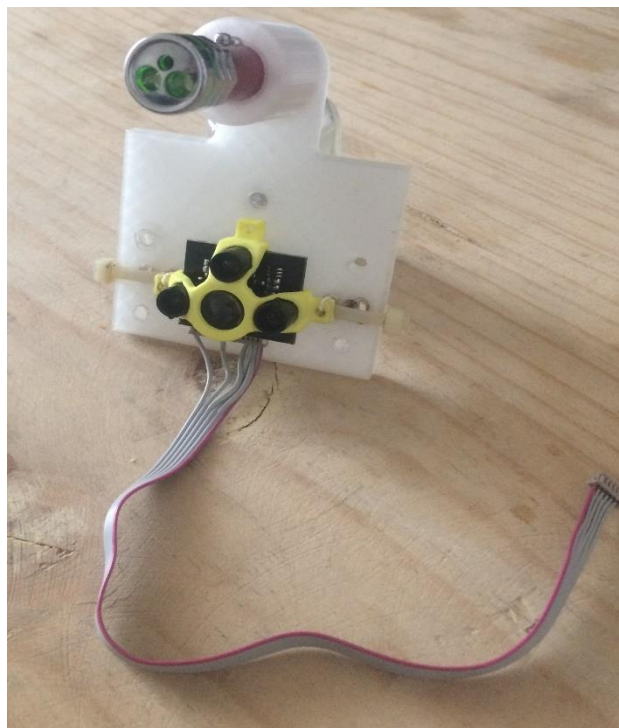


*Figura 3.12 Sistemas unificados en prototipo*

---

### 3.7 Apoyo visual Láser

Como apoyo visual para el usuario del sistema de medición con puntos fijos se colocó de manera externa al sistema un apuntador laser sencillo, este apuntador laser sirve principalmente para ubicar sobre la superficie a medir los puntos A y B, el uso del apuntador no tiene influencia en el funcionamiento interno del sistema, simplemente es una referencia visual para el medidor (figura 3.13); Sin embargo hay que tener en cuenta que para tener un punto de enfoque cercano al del sistema hay que ver la posición del apuntador, al estar en paralelo el apuntador láser con el sensor TeraRanger One se debe tomar en cuenta la distancia que hay entre estos, la cual es de 5 cm, por lo que la referencia visual siempre está a +5 cm sobre la referencia del sistema, la función del apuntador es meramente de apoyo, no influye en el resultado final de la medición en el sistema con puntos fijos y su uso es solamente en casos dónde el observador no pueda alcanzar a ver dónde se van a marcar los puntos A y B de la superficie a medir.



*Figura 3.13 Apoyo visual láser*

---

## 3.8 Pruebas del Sistema

Como prueba final del sistema realizamos una medición de comparación entre el sistema de medición con puntos fijos y una cinta métrica, para esto montamos el sistema frente a una superficie de madera para medirla a lo ancho. Apuntamos con el láser de apoyo visual hacia el punto **A** y posteriormente al punto **B** del ancho de la superficie (figuras 3.14 y 3.15), en la interfaz de la computadora se aprecia la aplicación en ejecución y la imagen captada en tiempo real con el módulo de la cámara V2, dentro de la aplicación se realiza el procesamiento de imagen que detecta los bordes de la superficie a medir, con esto logramos identificar y marcar los puntos **A** y **B** de la superficie dentro de la aplicación de medición.

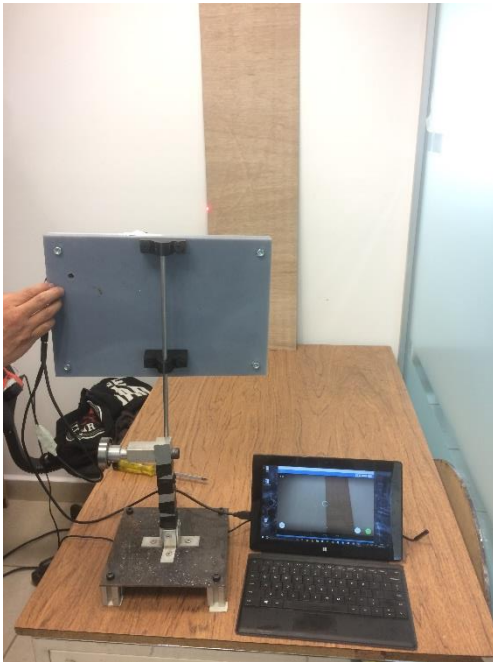


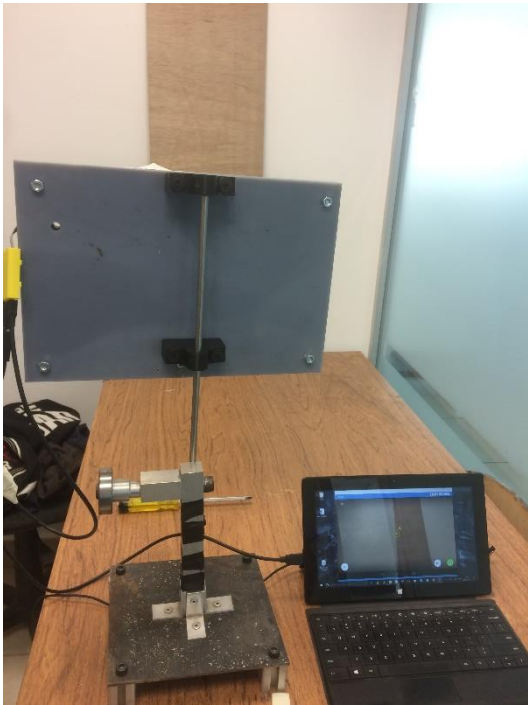
Figura 3.14 Marcación con láser en el punto A



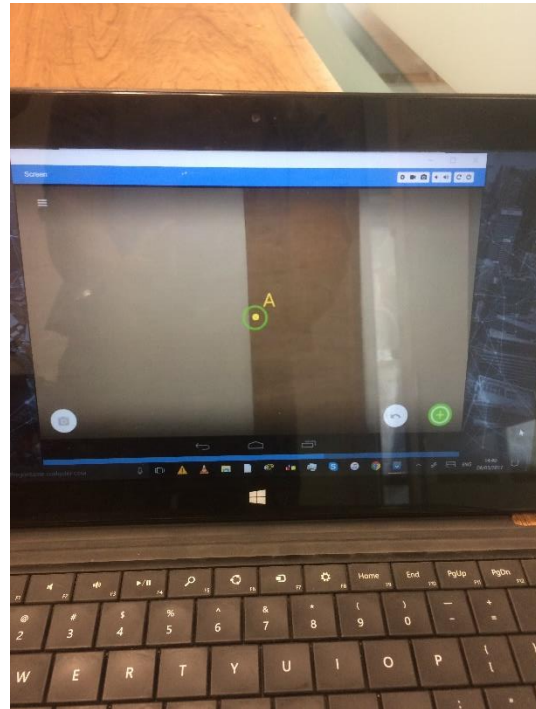
Figura 3.15 Marcación con láser en el punto B

---

Para calcular la medida dentro de la aplicación de medición marcamos con el botón virtual (+) la referencia del punto A de la superficie, que es la referencia de inicio (figuras 3.16 y 3.17).

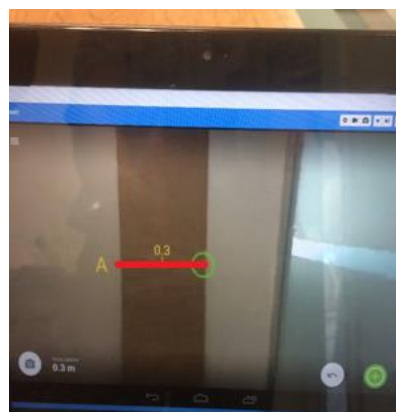


*Figura 3.16 Referencia punto A*



*Figura 3.17 Punto A en la aplicación*

Para completar la medición marcamos el punto B con el mismo botón virtual y la aplicación calcula automáticamente la distancia entre estos dos puntos y nos arroja la medición en metros (figura 3.18).



*Figura 3.18 Medida obtenida por el sistema*

Finalmente comparamos la medición realizada con la medición de una cinta métrica, esta última midió 30 cm y la medición del SMPF midió aproximadamente 30 cm (en promedio) en varias mediciones experimentales, (figuras 3.19 y 3.20). La tabla 3.1 muestra las mediciones calculadas en Matlab, este cálculo se hizo con los valores individuales de cada uno de los sensores del SMPF, y las mediciones experimentales realizadas por el SMPF terminado, marcado así una aproximación entre ambos cálculos.

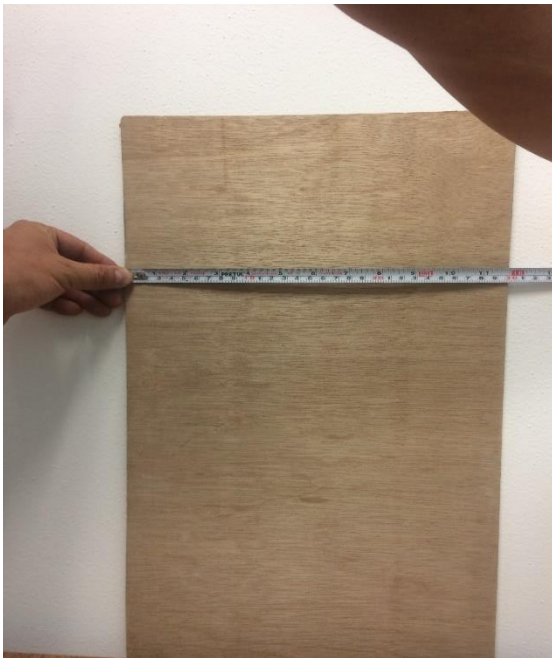


Figura 3.19 Medición con cinta métrica

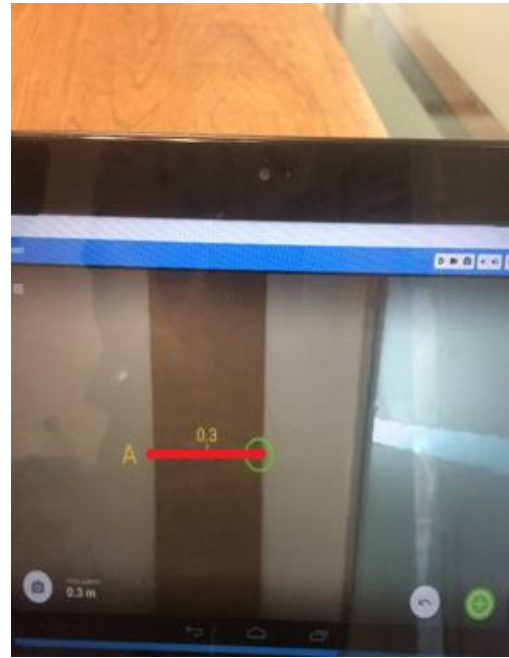


Figura 3.20 Medición Sistema con puntos fijos

Mediciones	Valor del sensor al punto A	Valor del sensor al punto B	Ángulo del acelerómetro	Valor calculado en Matlab	Valor del S.M.P.F	Valor medido cinta métrica
Medición 1	78 cm	76 cm	22 °	29.4500 cm	28 cm	30 cm
Medición 2	77 cm	78 cm	23 °	30.9175 cm	30 cm	30 cm
Medición 3	74 cm	75 cm	21 °	27.1708 cm	27 cm	30 cm
Medición 4	78 cm	78 cm	23 °	31.1013 cm	30 cm	30 cm
Medición 5	77 cm	78 cm	22 °	29.5916 cm	30 cm	30 cm
Medición 6	79 cm	78 cm	23 °	31.3161 cm	31 cm	30 cm
Medición 7	81 cm	82 cm	24 °	33.9037 cm	33 cm	30 cm
Medición 8	83 cm	80 cm	23 °	32.6296 cm	32 cm	30 cm
Medición 9	80 cm	81 cm	21 °	29.35.63 cm	29 cm	30 cm
Medición 10	82 cm	81 cm	22 °	31.1173 cm	30 cm	30 cm

Tabla 3.1 Mediciones calculadas y obtenidas experimentalmente

### 3.9 Comparación con sistemas comerciales y costo del SMPF

La siguiente tabla muestra la comparación de precios entre sistemas de medición.






Medidor	Precio	Rango de distancia	Precisión	Despliegue de la medición	Conexión con PC
<b>SMPF</b> 	\$16700 MXN	Hasta 14 m	$\pm 3$ cm	Aplicación en Android	SI
<b>Láser</b> 	\$8500 MXN	Hasta 14 m	$\pm 1$ cm	Display	NO
<b>Infrarrojo</b> 	\$1400 MXN	Hasta 6 m	$\pm 4$ cm	Interfaz Electrónica	NO
<b>Ultrasónico</b> 	\$4000 MXN	Hasta 8 m	$\pm 3.5$ cm	Display	NO
<b>Microondas</b> 	\$5000 MXN	Hasta 7 m	$\pm 4$ cm	Interfaz Electrónica	NO

Tabla 4.1 Tabla de comparación con sistemas comerciales

---

### Costo del Sistema de Medición con Puntos Fijos (Prototipo)

-Idea innovadora del sistema: + \$5000.00 MXN

-Horas de trabajo invertidas (Tomando \$80.04 MXN como salario mínimo al día)

- 37 días trabajados x \$80.04 MXN

Cálculo total de \$ x horas + \$3000.00 MXN

-Costo del material:

- Sensor TeraRanger One + \$6000.00 MXN
  - Acelerómetro ADXL345 + \$ 400.00 MXN
  - Cámara V2 + \$ 800.00 MXN
  - Raspberry Pi 3B + \$1500.00 MXN
- \$16700.00 MXN

Nota: El costo se determinó empíricamente, para un cálculo más preciso del valor del sistema hay que contactar a un experto en el tema.

### 3.10 Resultados finales del sistema

El sistema funciona como se propuso, este es un sistema de medición entre dos puntos fijos (figura 3.21) que nos proporciona la medida de una superficie plana sin necesidad de emplear una cinta métrica o algún sistema mecánico figura (3.22), la medida de la superficie se agrega a la fotografía obtenida por una cámara V2 la cual está integrada al sistema, conjuntamos la imagen con el dato de la medición, esta tiene un rango de precisión de  $\pm 2$  dm, y tenemos la opción de guardar en un archivo fotográfico la medición realizada con valores representados en metros.

El sistema cuenta con un rango de apertura hasta de  $60^\circ$  y una percepción de profundidad hasta de 5 metros ofreciendo un alcance considerable para calcular distancias por triangulación, con un nivel de precisión confiable.

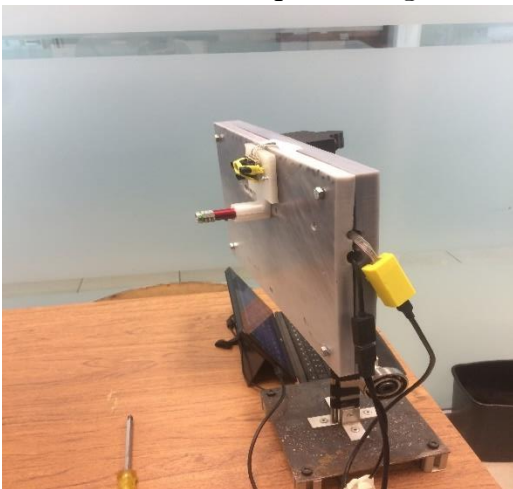


Figura 3.21 Sistema de medición con puntos fijos



Figura 3.22 Imagen con la medición

### 3.11 Rango de medición del sistema con puntos fijos

La siguiente ilustración muestra el rango total de medición que es posible realizar con el SMPF a partir del rango mínimo y máximo (figura 3.23); esta información está basada en una serie de mediciones experimentales que se realizaron con el sistema ya funcional; la tabla 5.1 muestra las mediciones calculadas en Matlab, las del SMPF y las mediciones realizadas con un flexómetro. El cálculo con Matlab se hizo con los valores individuales de cada uno de los sensores del SMPF; las mediciones experimentales se realizaron con el SMPF terminado; y las mediciones realizadas con el flexómetro fueron a mano.

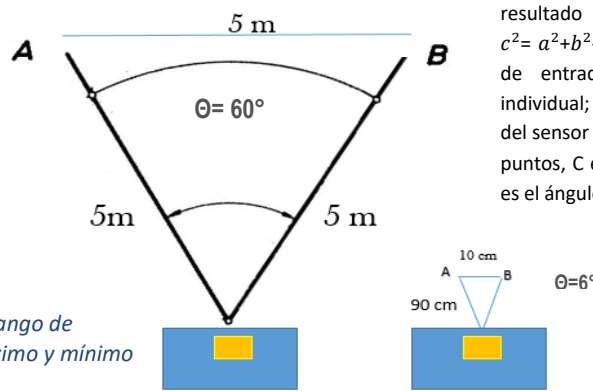


Figura 3.23 Rango de medición máximo y mínimo

Mediciones	Valor del sensor al punto A	Valor del sensor al punto B	$\theta$ = Ángulo del acelerómetro	Valor calculado en Matlab	Valor del SMPF	Valor medido con flexómetro
Medición 1	90 cm	90 cm	6.369477 °	9.99 cm	10 cm	10 cm
Medición 2	100 cm	101 cm	46.9043 °	79.9998 cm	85 cm	80 cm
Medición 3	150 cm	149 cm	55.4 °	138.9905 cm	148 cm	140 cm
Medición 4	200 cm	205 cm	57 °	193.2992 cm	210 cm	200 cm
Medición 5	250 cm	253 cm	57.5 °	241.9516 cm	255 cm	250 cm
Medición 6	300 cm	304 cm	58 °	292.8459 cm	315 cm	300 cm
Medición 7	350 cm	354 cm	58.2 °	342.3979 cm	350 cm	345 cm
Medición 8	400 cm	406 cm	58.5 °	393.8635 cm	405 cm	400 cm
Medición 9	450 cm	452 cm	59 °	444.194 cm	460 cm	450 cm
Medición 10	500 cm	500 cm	60 °	500 cm	510 cm	500 cm

Tabla 5.1 Mediciones experimentales del sistema - precisión  $\pm 2$  dm

---

En la tabla 5.1 se observa una notable diferencia entre los valores del SMPF con los valores calculados en Matlab y los valores medidos con el flexómetro, esto se debe a un error en la medición del sistema; el error de medición surge al existir una falta de calibración entre los sensores: TeraRanger One y acelerómetro ADXL345, con la cámara V2; como se puede observar en la columna amarilla (SMPF), los valores obtenidos por el sistema se aproximan a los valores calculados en la columna naranja (Matlab); los valores obtenidos en la columna naranja son el resultado de aplicar la ley de cosenos (ecu. 2.1) con las variables de entrada: sensor TeraRanger One (puntos A,B) y la variable del acelerómetro ( $\Theta$ ); los valores de la columna amarilla son el resultado de aplicar la ley de cosenos por medio de un algoritmo matemático dentro de la aplicación de medición a las variables de entrada (a, b y  $\theta$ ), sin embargo, la falta de calibración mencionada, produce una perturbación en cada una de las variables de entrada, la alteración producida se determina de la siguiente manera:

$$\mathbf{a} = \mathbf{A} \pm \delta\mathbf{A}, \mathbf{b} = \mathbf{B} \pm \delta\mathbf{B}, \boldsymbol{\theta} = \boldsymbol{\Theta} \pm \delta\boldsymbol{\Theta}$$

Donde: (a, b,  $\theta$ ) son las variables de entrada y ( $\delta\mathbf{A}$ ,  $\delta\mathbf{B}$ ,  $\delta\boldsymbol{\Theta}$ ) son el resultado de la perturbación en las variables de entrada.

Este error de calibración puede llegar a ser corregido si se trabaja más en el ajuste de los sensores.

La medición del SMPF tiene una precisión de  $\pm 2$  dm. Este resultado se obtuvo, determinando el error  $E$  (ecu 3.2), considerando las mediciones sólo en superficies planas. Para obtener  $E$  primero se determinan el valor mínimo y el valor máximo del error  $E_i$  en un conjunto de mediciones experimentales. El error  $E_i$  es la diferencia entre una medida de referencia (en este caso la medida tomada con un flexómetro) y las mediciones experimentales del SMPF; concretamente, se toma el valor de la medida de referencia (flexómetro) y se resta con cada una de las mediciones del SMPF, entonces obtenemos  $E_i$ ; como se observa en la ecuación 3.1.

$$\mathbf{E}_i = \mathbf{x}_r - \mathbf{x}_{SMPF} \quad (3.1)$$

Donde  $E_i$  es el error entre la medición del flexómetro y las mediciones del SMPF, ( $x_r$ ) es la medida dada con flexómetro, y ( $x_{SMPF}$ ) es una de las mediciones del SMPF. Como ejemplo a esto tenemos lo siguiente:

$$x_r = 2.00 \text{ m} \text{ Y } x_{SMPF} = 1.9 \text{ m}; \text{ entonces } E_i = 2\text{m} - 1.9 \text{ m} = 0.1 \text{ m}$$

Para definir la precisión del sistema SPMF, calculamos  $E$  comparando el valor máximo y mínimo de  $E_i$ , como se observa en la ecuación 3.2.

$$\mathbf{E} = [\mathbf{max}(E_i) - \mathbf{min}(E_i)] \quad (3.2)$$

El cálculo del error  $E$  nos permite determinar la precisión del sistema la cual es de  $\pm 2$  dm.

---

---

## CONCLUSIONES

En el presente trabajo de tesis se presentó el sistema de medición con puntos fijos, el cual se construyó para realizar mediciones de forma electrónica cuyo desempeño fue mostrado físicamente y mediante simulación.

Durante su desarrollo se aplicaron diversos tipos de conocimientos en el área de física, matemáticas, electrónica y programación, el sistema engloba diversas partes de tecnología de última generación, el conocimiento y la manipulación de estas permitió unificarlas en nuevo concepto propio, el medir distancias de manera sencilla utilizando una aplicación en Android. Uno de los primeros problemas que se presentó fue la pérdida de información del sensor por cambios en las condiciones de luz, por lo que se optó por codificar la señal durante la medición, y su decodificación en la tarjeta, teniendo así un buen rango de precisión sin pérdida de información. Para el acoplamiento del sensor TeraRanger One, la tarjeta Raspberry Pi 3, el acelerómetro ADXL345 y la cámara V2 se necesitó conocer el funcionamiento de cada uno de estos dispositivos, la forma de fusionar cada uno de ellos para lograr un sistema de medición requirió tiempo y dedicación, la programación utilizada es de alto grado de complejidad, pues el uso de librerías de procesamiento de imagen no es tan accesible así como la programación utilizada para recolectar, procesar y unificar la información entregada por cada dispositivo utilizado.

Se realizaron diferentes pruebas de medición para garantizar la estabilidad del sistema terminado. Principalmente fue realizar una medición a una misma superficie de tres maneras diferentes: matemáticamente con Matlab, físicamente con un flexómetro y una medición realizada con el SMPF; esta comparación de mediciones (pág. 66) nos muestran una notable diferencia entre las mediciones realizadas con el SMPF respecto a las otras mediciones calculadas; esto se debe principalmente a que hay que optimizar y mejorar la aplicación de medición, que es la encargada del procesamiento de las variables de entrada sobre la imagen, sin embargo el trabajo final realiza una medición entre dos puntos fijos a y b con cierta precisión ( $\pm 2 \text{ dm.}$ ), que fue el objetivo final; El valor que toma este trabajo es principalmente de utilizar los conocimientos adquiridos a lo largo de la carrera de electrónica para crear un sistema de medición propio y brindar a las siguientes generaciones una referencia técnica para trabajos relacionados con medición de superficies a través de la electrónica.

---

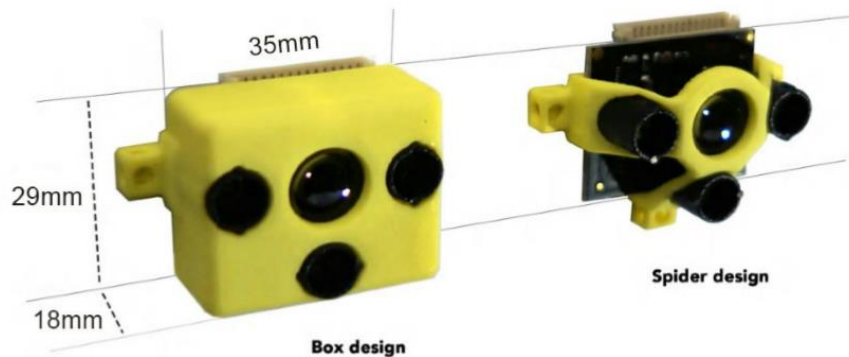
## APÉNDICE A

### Tabla de comparación de sensores de profundidad

	Visión Estereoscópica	Rango Definido	Rango Programable	Tiempo de Vuelo
Precisión de profundidad	mm - cm	mm - cm	$\mu m$ - cm	mm - cm
Tiempo de lectura	media	Rápida	Media/Rápida	Rápida
Rango de distancia	Rango medio	Rango medio/corto	Rango medio/corto	Rango corto/largo
Performance con luz interior	Débil	Bueno	Bueno	Bueno
Performance con luz exterior	Bueno	Débil/Justo	Débil/Justo	Bueno
Complejidad de Software	Alto	Bajo/Medio	Medio/Alto	Alto
Costo de material	Bajo	Medio	Medio/Alto	Medio

Tabla 6.1 Comparación de sensores electrónicos de profundidad

### Comparación de los sensores Box y Spider



---

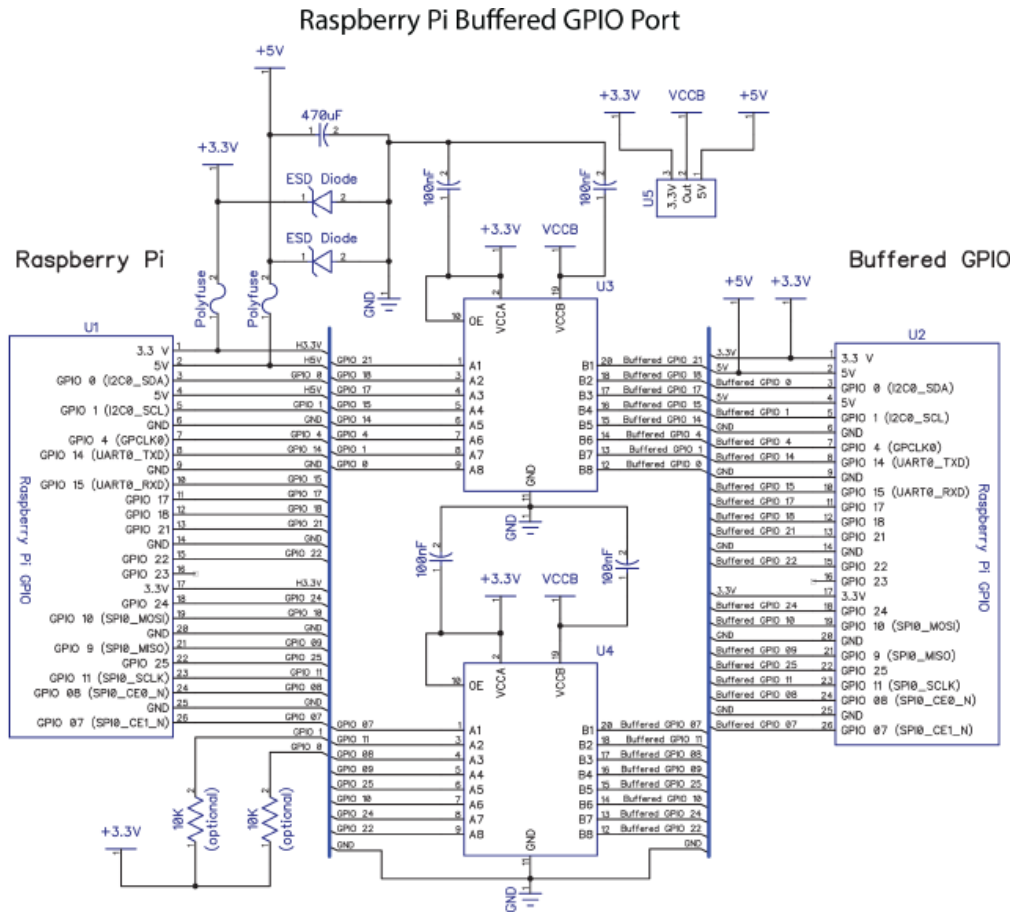
## APÉNDICE B

### Programa del Sensor TeraRanger One

```
long distancia;
long tiempo;
void setup(){
  Serial.begin(9600);
  pinMode(13, OUTPUT); /* activación del pin 13 como salida: para el haz infrarrojo */
  pinMode(12, INPUT); /* activación del pin 12 como entrada: tiempo del rebote del haz */
}
void loop(){
  digitalWrite(9,LOW); /* Para la estabilización del sensor */
  delayMicroseconds(5);
  digitalWrite(9, HIGH); /* envío del haz infrarrojo */
  delayMicroseconds(10);
  tiempo=pulseIn(8, HIGH); /* Función para medir la longitud del pulso entrante. Mide el
tiempo que transcurrido entre el envío
del haz infrarrojo y cuando el sensor recibe el rebote, es decir: desde que el pin 12 empieza
a recibir el rebote, HIGH, hasta que
deja de hacerlo, LOW, la longitud del pulso entrante */
  distancia= int(0.017*tiempo); /* fórmula para calcular la distancia obteniendo un valor
entero */
  /* Monitorización en milímetros por el monitor serial */
  Serial.println("Distancia ");
  Serial.println(distancia);
  Serial.println(" mm");
  delay(1000)
}
```

# APÉNDICE C

## Diagrama interno de la Raspberry Pi 3 B










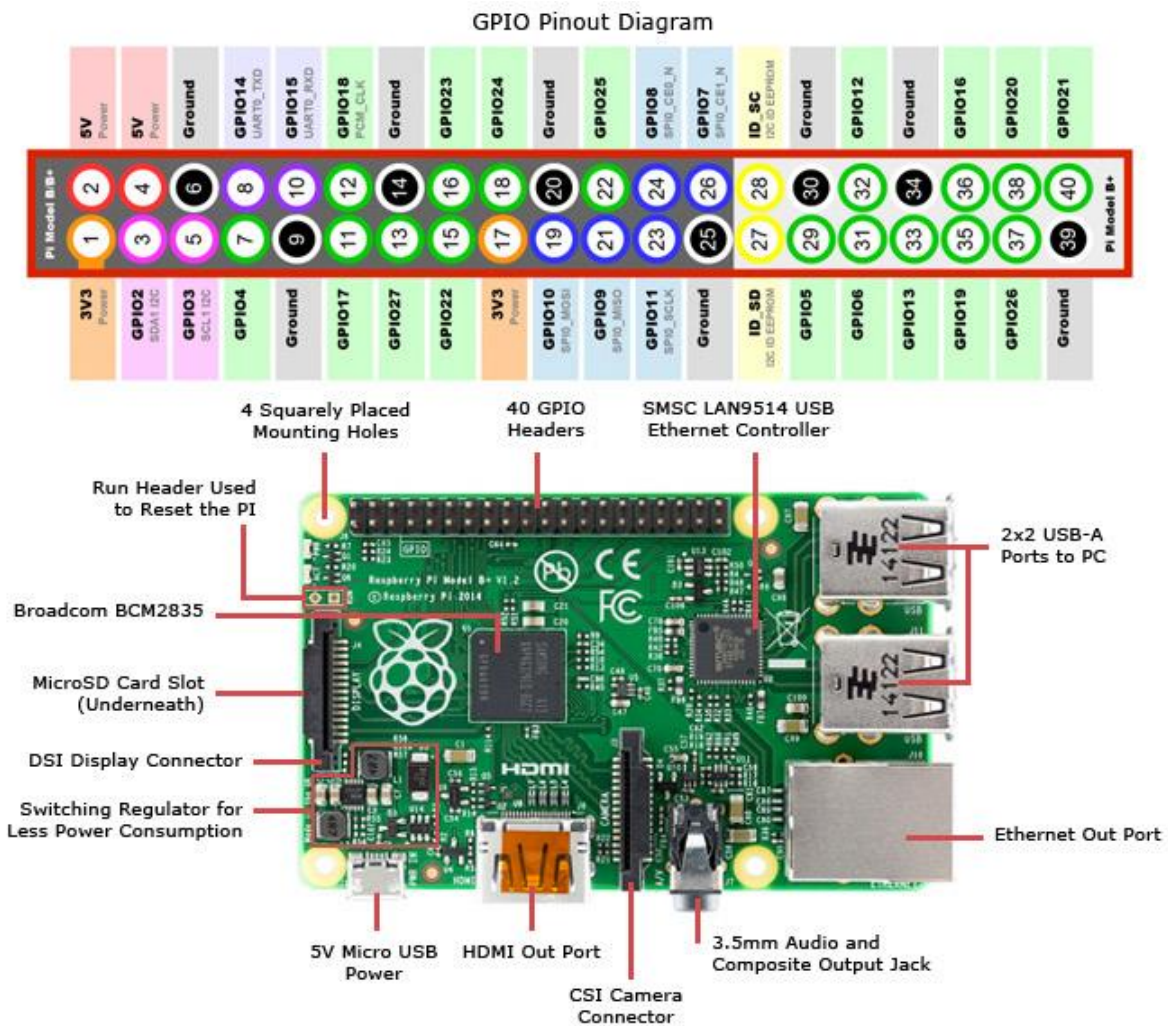
						
Raspberry Pi	Modelo A	Modelo A+	Modelo B	Modelo B+	RPi V2 modelo B	RPi 3 modelo B
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837
CPU	700MHz ARM1176JFZ-S	700MHz ARM1176JFZ-S	700MHz ARM1176JFZ-S	700MHz ARM1176JFZ-S	900MHz Quad-core ARM Cortex-A7	1.2GHz Quad Cortex A53
GPU	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV	250Mhz VideoCore IV	400Mhz VideoCore IV
RAM	256Mb	512Mb	512Mb	512Mb	1Gb	1Gb
USB	1	1	2	4	4	4
Video	RCA, HDMI	Jack, HDMI	RCA, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI
Audio	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI	Jack, HDMI
Boot	Memoria SD	Memoria microSD	Memoria SD	Memoria microSD	Memoria microSD	Memoria microSD
Wireless	No tiene	No tiene	No tiene	No tiene	No tiene	802.11n / Bluetooth 4.1
Red Ethernet	No tiene	No tiene	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100	Ethernet 10/100
Alimentación	5V / 2Amp	5V / 2Amp	5V / 2Amp	5V / 2Amp	5V / 2Amp	5V / 2,5Amp
GPIO	26 pines GPIO	40 pines GPIO	26 pines GPIO	40 pines GPIO	40 pines GPIO	40 pines GPIO
Tamaño	85,6 x 53,98 mm	65 x 56 mm	85,6 x 53,98 mm	85 x 56 x 17 mm	85 x 56 x 17 mm	85 x 56 x 17 mm

Tabla 7.1 Comparación de la familia Raspberry



## Tabla de especificaciones del Acelerómetro ADXL345

### SPECIFICATIONS

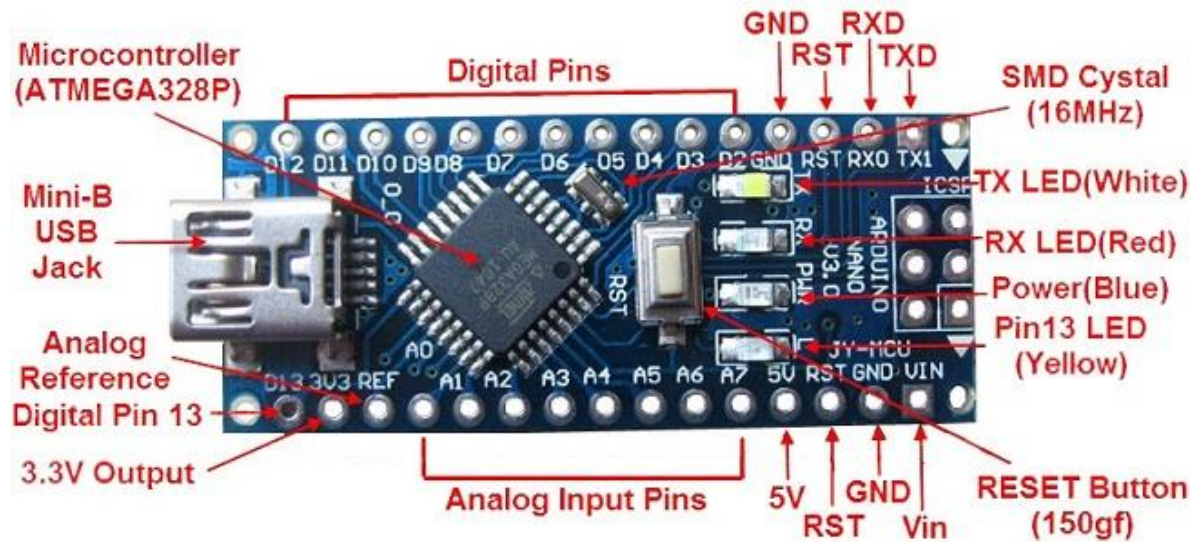
T<sub>A</sub> = 25°C, V<sub>S</sub> = 2.5 V, V<sub>DDIO</sub> = 1.8 V, acceleration = 0 g, C<sub>S</sub> = 1 μF tantalum, C<sub>IO</sub> = 0.1 μF, unless otherwise noted.

Table 1. Specifications<sup>1</sup>

Parameter	Test Conditions	Min	Typ	Max	Unit
<b>SENSOR INPUT</b>					
Measurement Range	Each axis User selectable		±2, ±4, ±8, ±16		g
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity <sup>2</sup>			±1		%
<b>OUTPUT RESOLUTION</b>					
All g Ranges	Each axis 10-bit resolution		10		Bits
±2 g Range	Full resolution		10		Bits
±4 g Range	Full resolution		11		Bits
±8 g Range	Full resolution		12		Bits
±16 g Range	Full resolution		13		Bits
<b>SENSITIVITY</b>					
Sensitivity at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	Each axis ±2 g, 10-bit or full resolution	232	256	286	LSB/g
Scale Factor at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±2 g, 10-bit or full resolution	3.5	3.9	4.3	mg/LSB
Sensitivity at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±4 g, 10-bit resolution	116	128	143	LSB/g
Scale Factor at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±4 g, 10-bit resolution	7.0	7.8	8.6	mg/LSB
Sensitivity at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±8 g, 10-bit resolution	58	64	71	LSB/g
Scale Factor at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±8 g, 10-bit resolution	14.0	15.6	17.2	mg/LSB
Sensitivity at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±16 g, 10-bit resolution	29	32	36	LSB/g
Scale Factor at X <sub>OUT</sub> , Y <sub>OUT</sub> , Z <sub>OUT</sub>	±16 g, 10-bit resolution	28.1	31.2	34.3	mg/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
<b>0 g BIAS LEVEL</b>					
0 g Output for X <sub>OUT</sub> , Y <sub>OUT</sub>	Each axis	-150	±40	+150	mg
0 g Output for Z <sub>OUT</sub>		-250	±80	+250	mg
0 g Offset vs. Temperature for x-, y-Axes			±0.8		mg/°C
0 g Offset vs. Temperature for z-Axis			±4.5		mg/°C
<b>NOISE PERFORMANCE</b>					
Noise (x-, y-Axes)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.0		LSB rms
Noise (z-Axis)	Data rate = 100 Hz for ±2 g, 10-bit or full resolution		<1.5		LSB rms
<b>OUTPUT DATA RATE AND BANDWIDTH</b>					
Measurement Rate <sup>3</sup>	User selectable	6.25		3200	Hz
<b>SELF-TEST<sup>4</sup></b>					
Output Change in x-Axis	Data rate ≥ 100 Hz, 2.0 V ≤ V <sub>S</sub> ≤ 3.6 V	0.20		2.10	g
Output Change in y-Axis		-2.10		-0.20	g
Output Change in z-Axis		0.30		3.40	g
<b>POWER SUPPLY</b>					
Operating Voltage Range (V <sub>S</sub> )		2.0	2.5	3.6	V
Interface Voltage Range (V <sub>DDIO</sub> )	V <sub>S</sub> ≤ 2.5 V	1.7	1.8	V <sub>S</sub>	V
	V <sub>S</sub> ≥ 2.5 V	2.0	2.5	V <sub>S</sub>	V
Supply Current	Data rate > 100 Hz		145		μA
	Data rate < 10 Hz		40		μA
Standby Mode Leakage Current			0.1	2	μA
Turn-On Time <sup>5</sup>	Data rate = 3200 Hz		1.4		ms
<b>TEMPERATURE</b>					
Operating Temperature Range		-40		+85	°C
<b>WEIGHT</b>					
Device Weight			20		mg

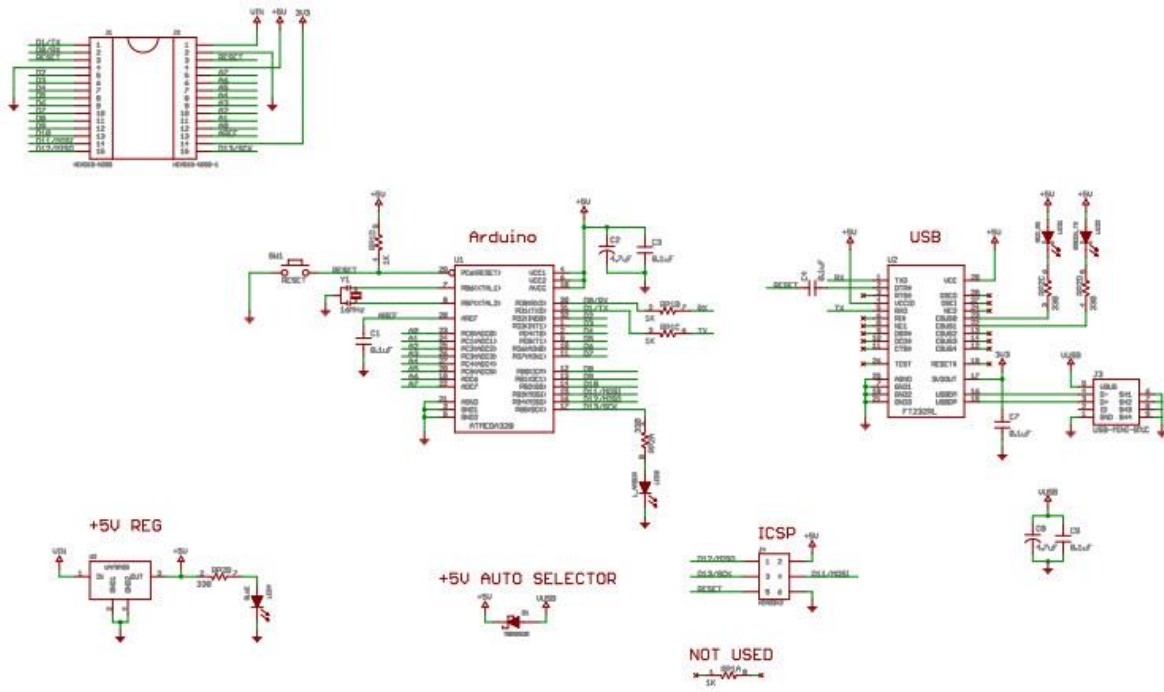
Tabla 8.1 Especificaciones del acelerómetro ADXL345

# Arduino Nano



## Diagrama interno

# Arduino Nano



---

## Librería de programación del acelerómetro

```
#import the adxl345 module
import adxl345

#create ADXL345 object
accel = adxl345.ADXL345()

#get axes as g
axes = accel.getAxes(True)
# to get axes as ms^2 use
#axes = accel.getAxes(False)

#put the axes into variables
x = axes['x']
y = axes['y']
z = axes['z']

#print axes
print x
print y
print z
```

---

## APÉNDICE D

### Algoritmo de cálculo en la librería Matlab

```
function [m, a] = ab2v(v1, v2)
% Magnitud de los vectores A y B
m(1) = norm(v1);
m(2) = norm(v2);

% Ángulo del vector A al vértice del sensor
a(1 : 3) = acosd(v1 / m(1));
% Ángulo del vector B al vértice del sensor
a(4 : 6) = acosd(v2 / m(2));

% Producto punto (v1, v2) igual sum(v1 .* v2)
a(7) = acosd(dot(v1, v2) / m(1) / m(2));

function display_angles(m, a)

disp(' ')
disp(['Angulo entre los vectores: ' num2str(a(7))])

disp(' ')
disp('Vector A')
disp(['Magnitud :' num2str(m(1))])
disp(['Angulo con x-axis: ' num2str(a(1))])
disp(['Angulo con y-axis: ' num2str(a(2))])
disp(['Angulo con z-axis: ' num2str(a(3))])

disp(' ')
disp('Vector B')
disp(['Magnitud :' num2str(m(2))])
disp(['Angulo con x-axis: ' num2str(a(4))])
disp(['Angulo con y-axis: ' num2str(a(5))])
disp(['Angulo con z-axis: ' num2str(a(6))])
```

---

## BIBLIOGRAFÍA

[1] <https://es.wikipedia.org/wiki/Láser>

[2] [https://es.wikipedia.org/wiki/Efecto\\_Doppler](https://es.wikipedia.org/wiki/Efecto_Doppler)

[3] <https://es.wikipedia.org/wiki/LIDAR>

[4] **Mobile Design and Development**, BRIAN FLING, (2009), p 336,

ISBN 978-0-596-15544-5, EDITORIAL O'Reilly Media.

[5] **Android Essentials**, CHRIS HASEMAN, (2008), p 116, ISBN 978-1430210641, EDITORIAL APRESS.

[6] **Hello, Android: Introducing Google's Mobile Development Platform**, ED BURNETTE, (2010), p 300, ISBN 978-1-93435-656-2, EDITORIAL Pragmatic Bookshelf.

[7] **Android Programming Tutorials**, MARK LAWRENCE MURPHY, (2010), p 436, ISBN 978-0981678047, EDITORIAL Commons Ware.

[8] **PROCESAMIENTO Y ANÁLISIS DIGITAL DE IMÁGENES**, RODRIGUEZ MORALES, ROBERTO / SOSSA AZUELA, JUAN HUMBERTO, (2011), p 430, ISBN 978-84-9964-077-8, EDITORIAL RA-MA EDITORIAL.

