UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO

FACULTAD DE INGENIERÍA ELÉCTRICA

TESIS

PROCESAMIENTO DIGITAL DE VIDEO EN TIEMPO REAL CON DISPOSITIVOS ANDROID PARA LOCALIZACIÓN Y SEGUIMIENTO DE FIGURAS GEOMÉTRICAS ESPECÍFICAS

Que para obtener el Título de:

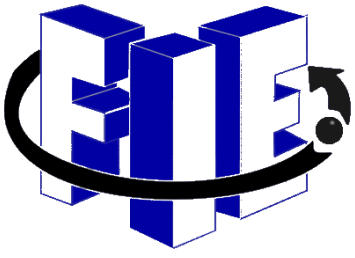
INGENIERO EN ELECTRÓNICA

Presenta:

Adrian Romeo Barajas García

Asesor de Tesis:

M.C. Félix Jiménez Pérez



Morelia, Michoacán, agosto del 2019

Resumen

En este trabajo se presenta el desarrollo de una aplicación programada para dispositivos personales con sistema Operativo Android®, el cual permite el reconocimiento de patrones en secuencias de imágenes en tiempo real, es decir, video.

El software está integrado de cuatro partes básicas:

- Adquisición de imágenes en tiempo real.
- Análisis y procesamiento de las imágenes obtenidas.
- Búsqueda del patrón en las imágenes de video recibidas.
- Seguimiento del patrón una vez localizado.

La imagen procesada se descompone en matrices en el software (app) de Android®, y con la ayuda de la biblioteca OpenCV® se logra hacer un análisis de imágenes buscando un patrón definido, el cual es una figura geométrica determinada.

Operación básica: El usuario debe instalar la aplicación en el dispositivo Android, el cual con ayuda de la cámara integrada, se busca en las imágenes recibidas (video) para poder localizar el patrón. Dependiendo de la información que se obtiene al procesar las imágenes se genera la instrucción necesaria para modificar la posición de la cámara, con la ayuda de los motores integrados al vehículo que servirá como base del desplazamiento del smartphone.

Palabras clave: Tracking, aplicación, imágenes, bluetooth, microcontroladores

Abstract

This paper presents the development of a programmed application for personal devices with Android® Operative System, which allows the recognition of patterns in sequences of images in real time, that is, video.

The software it‘s integrated by four basic parts:

- Real time images adquisition.
- Analysis and processing of the images obtained.
- Search for the pattern in the received video images.
- Pattern tracking once located.

The processed image is decomposed into matrices in the Android® software, with the help of the library OpenCV® it’s possible to do an images analysis looking for a definited pattern, which it’s a certain geometric figure.

Basic operation: The user must install the application on the Android device, with the help of the integrated camera; the received images are searched to locate the pattern. Depending on the information obtained when processing the images, the necessary instruction is generated to modify the camera position, with the help of the engines integrated into the vehicle that will serve as the basis of the smartphone's movement.

Tabla de Contenido

Capítulo 1
Introducción:

El reconocimiento de patrones en las imágenes está basado en diferentes técnicas y sirven en diversas áreas como seguridad, investigación y en campos diversos como el entretenimiento. El procesamiento digital consiste en convertir las imágenes a matrices de datos los cuales podemos manipular de maneras más simples con ayuda de procesos computacionales.

Los dispositivos con sistema operativo Android® tienen características físicas y de rendimiento suficientes para realizar la tarea de una pequeña computadora con gran eficiencia.

1.1 Antecedentes

1.1.1 Procesamiento de señales

Señal: *Variación de una corriente eléctrica u otra magnitud física que se utiliza para transmitir información.*

El Procesamiento de señales trata de la representación, transformación u manipulación de señales, así mismo, considera la información que nos puedan representar y se busca la recepción, descodificación y entendimiento de la misma, y viceversa.

Para el procesamiento digital de señales es necesario hablar de capacidad de cálculo, entre mayor capacidad de cálculo, es mayor la velocidad de recolección de muestras y, a su vez, de transformarlas. Antes de contar con computadores o dispositivos específicos para dicha tarea, se utilizaban complejos dispositivos mecánicos que resolvían la cantidad de procesos realizados.

Con la necesidad de que el procesamiento fuera cada vez más rápido, se empieza a utilizar computadoras que realicen esta tarea, se tiene entendido que alguno de los primeros usos registrado de procesamiento digital de señales por medio de una computadora se tiene en la década de los cincuentas donde se usó para la prospección petrolífera donde se guardaban los datos sísmicos en cintas magnéticas para su posterior análisis, durante el análisis se pretendía disminuir los costos de extracción de petróleo al poder interpretar los datos de manera eficiente antes de perforar, aunque como se entiende el problema era que no se podía analizar en tiempo real, lo cual cambiaba costos por tiempo.

En los años ochenta, los microprocesadores eran demasiado lentos para implementarlo en tiempo real, de esta manera nace el ***DSP*** *(Digital Signal Processor)* que son circuitos integrados con capacidad de microcomputadores.

1.1.2 Procesamiento Digital de Señales

En el procesamiento digital de imágenes, se utiliza una gran cantidad de recursos computacionales, de manera común se utilizan los microprocesadores, de tal forma que es lento e ineficaz pero de fácil adquisición, por otro lado, los **DSP**, tienen un gran mercado y precio elevado, parecieren un mal necesario, aunque actualmente existen diversas opciones.

Figura 1 Ejemplo de un DSP creado por TI.

Los **DSP,** como el que se muestra en la **Figura 1**pueden ser sustituidos con facilidad por los microprocesadores de propósito general, esto debido al gran crecimiento de su mercado es sencillo conseguirlos a costos moderados, con poder de cálculo muy similar al utilizado por los **DSP.**

Aún con estos dispositivos digitales viene el problema central, no se puede procesar imágenes en tiempo real, es necesario que las imágenes estén previamente capturadas y posteriormente ser analizadas, de aquí parte la idea, del rápido crecimiento de hardware y cada vez mayor capacidad de almacenamiento y de procesos computacionales, se ha logrado explotar esta parte y se ha comenzado a analizar imágenes con mayor velocidad.

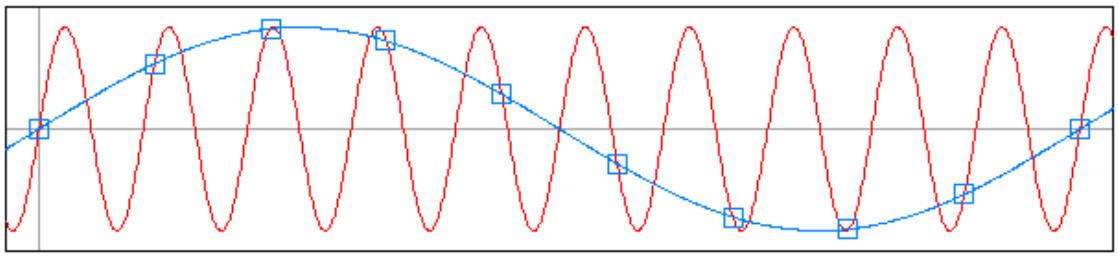


Figura 2Muestreo lento de una senoidal.

En la **Figura 2**se observa el problema al capturar una señal a una velocidad inferior a la velocidad de la máxima frecuencia de la señal a capturar

1.1.3 Procesamiento Digital de Imágenes

Las imágenes pueden ser representadas como matrices, una para cada color, es decir, para imágenes de color son 3 matrices (RGB) una para los colores Rojos, otra para colores Verdes y una última para colores Azules, adicionalmente algunos procesadores utilizan un matriz conocida como Alfa que se utiliza para las transparencias.

Una imagen en escala de grises es una función de una matriz con dos dimensiones (X y Y) donde se el valor de estos representa la intensidad en la escala de grises.

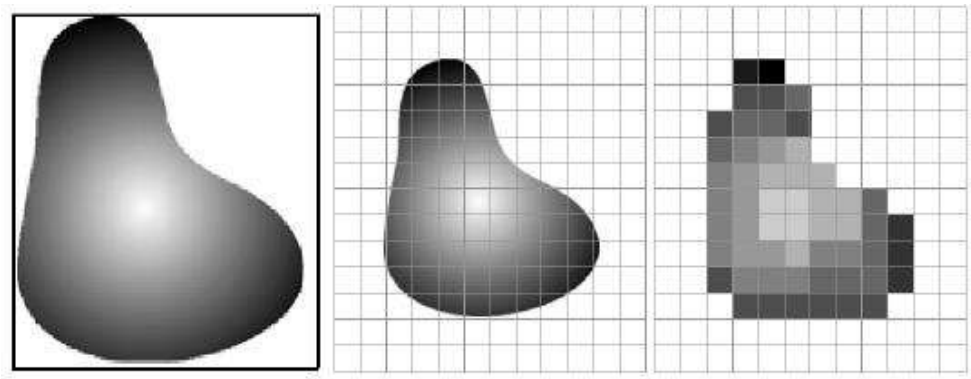


Figura 3Escala de grises vectorizada en X y Y representando pixeles

En la **Figura 3** se observa un ejemplo de cómo se representa una figura en escala de grises en un matriz

1.1.4 Procesamiento Digital de Video.

Partiendo de que un video son imágenes en secuencia (en filmaciones de películas suelen ser 24 imágenes por segundo), para el procesamiento de video se requiere de un dispositivo con gran poder de cálculo, en la actualidad se utilizan computadores de gran capacidad de cálculo para realizar dicha actividad, sin embargo, el problema continua y solo se pueden analizar videos que están previamente almacenados, por lo que los últimos 15 años se han enfocado en realizarlo en tiempo real, es decir, que mientras se toma un video se esté analizando.

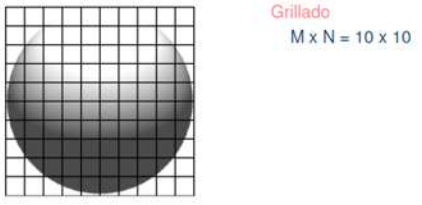


Figura 5 Segmentación de una imagen en escala de grises con todos los tonos.

Figura 4 Clasificación de tonos de 0 a 255

En la **Figura 5** se muestra la captura de una imagen en un sensor, el sensor tiene un área de 10 x10 pixeles, los sensores codifican cada pixel en un rango de 0 (negro) a 255(blanco) como se observa en la **Figura 1**.

El procesamiento digital de video consta de algunas etapas bien definidas:

- **Captura:** Diseño de las propiedades de captura, como la cámara y los colores a utilizar.
- **Pre-Procesamiento:** En la imagen capturada se enfoca al objeto de interés, eliminando el entorno que no nos sirve.
- **Segmentación:** Reconocer los objetos de la imagen.
- **Extracción:** Seleccionar las características y patrones que son de interés.
- **Identificación:** Comparar el paso anterior con una imagen ya guardada confirmar su presencia y su similitud.

Con la adopción de las redes sociales y la dependencia a dispositivos electrónicos de uso personal, es más común ver el procesamiento de imágenes para el reconocimiento facial y gesticular. [1]

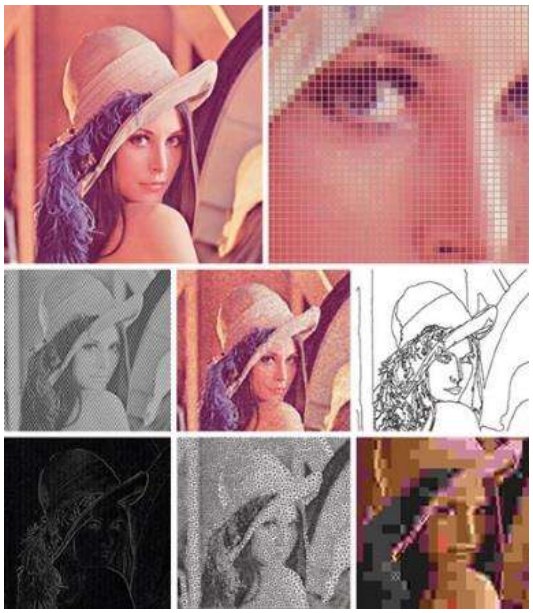


Figura 6 Imagen procesada con distintos filtros, demostrando la gran utilidad de dicha herramienta. Fotografía de la modelo Lena Södenberg.

En la **Figura 6** se aprecia la imagen de la modelo Lena, utilizada ampliamente en el procesamiento digital de imágenes, puesto que presenta una gran cantidad de texturas, en la figura se aprecian los distintos tipos de procesamientos que se le hacen a la fotografía.

1.2 Objetivo

1.2.1 Objetivos Generales

Creación y diseño de un software para dispositivos móviles (APP) que realice el procesamiento digital de video en tiempo real para el reconocimiento de figuras geométricas especificas con suficiente poder de cálculo, demostrando que los dispositivos móviles actuales tienen características suficientes para tareas poderosas como el procesamiento digital de imágenes.

1.2.2 Objetivos Particulares

- Programar una App para dispositivos Android® con el programa **AndroidStudio®**.
- Uso de la biblioteca **OpenCV®**.
- Búsqueda y detección de círculos con diámetro predefinido.
- Controlar un vehículo sin conexión física que realice la búsqueda de una figura.
- El software se comunicará con un módulo bluetooth para el movimiento de un vehículo que realiza movimiento con dos motores.

1.3 Justificación

La tecnología avanzando a pasos agigantados permite resolver la problemática de procesamiento digital de video utilizando herramientas de bajo costo y al alcance de todos.

Considerando la popularidad de los dispositivos inteligentes, se decidió desarrollar una aplicación para dispositivos con sistemas operativos Android® aprovechando los recursos mencionados.

Teniendo como eje principal del proyecto la eficiencia y practicidad, se decidió desarrollar la aplicación móvil para que la mayoría de los dispositivos con sistema operativo Android pueda descargarlo y hacer uso de él, sin la necesidad de tener que conseguir un DSP que es costoso y más complicado de adquirir que un celular o tableta con Android® y no es de propósito general.

Además de contar con una herramienta de procesamiento digital de video en tiempo real, los alumnos de la Facultad de Ingeniería Eléctrica podrán obtener conocimiento de un medio de procesamiento que no se ve en el plan de estudios actual. Y, tienen la oportunidad de conocer los aspectos de programación básica para el desarrollo de aplicaciones en esta plataforma. Es una buena oportunidad de contribuir al desarrollo de los planes de estudio de la Facultad, y ser la base para una aplicación comercial en un futuro próximo.

1.4 Metodología

Investigación bibliográfica del Procesamiento de señales, tanto analógicas como digitales para posteriormente aplicarlo en un video. Asesoramiento en programación de Android, consulta en libros respecto al tema y búsquedas en internet sobre la biblioteca llamada **OpenCV®.**

Se efectuaron diversas pruebas una vez instalado el software (APP) en el dispositivo con sistema Operativo Android® para evaluar la eficiencia y fidelidad del video procesado y la comparación con el procesamiento de imágenes previamente guardadas.

1.5 Contenido de la Tesis

En el capítulo 1 se presenta una breve introducción al trabajo a desarrollar, así como antecedentes históricos de señales y el procesamiento de ellas a la par de mencionar las mejoras. Se describen los objetivos y justificación del motivo que impulsó el desarrollo de este tema de tesis.

En el capítulo 2 se muestran los actuales diseños y avances sobre el tema, una introducción a Android, historia de Android y detalles del procesamiento de señales, como funciona y cuál ha sido la causa de su creación.

En el capítulo 3 se realiza la descripción detallada del procesamiento digital de imágenes, el procesamiento necesario para este proyecto, sumado, una introducción a Android Studio, programación en base JAVA, **OpenCV®,** y protocolo de comunicación bluetooth que realiza la tarea de enviar las señales de control del dispositivo Android hacia el vehículo móvil.

En el capítulo 4 se describen los componentes físicos que conformaran el vehículo a controlar, abarcando una breve introducción a los distintos tipos de motores, controladores, microcontrolador, lenguaje de programación C, la adquisición de un soporte para los componentes, diseño de pcb, modulo bluetooth y fuente de alimentación.

En el capítulo 5 se habla de las pruebas y resultados obtenidos de la experimentación del presente proyecto, en relación a los objetivos y la metodología planteada para atacar el problema a solucionar.

En el capítulo 6 se plasmaron las conclusiones obtenidas de la experimentación, del avance obtenido y el cumplimiento de los objetivos particulares y generales mostrados al inicio del presente documento.

En el capítulo 7 se desarrollan las problemáticas a solucionar, para realizarse en trabajos futuros a la publicación de este documento.

Capítulo 2
Android

2.1 Historia

Android® es el sistema operativo más usado actualmente, se estima que cerca de un millón de dispositivos nuevos al día son adquiridos [2].

Desarrollada gracias a la contribución de la comunidad de Código Libre de Linux, lo cual le permitió ser el sistema operativo con más alto porcentaje de crecimiento, con lo que se tiene un alcance global y se da la oportunidad de crear tu propias aplicaciones y distribuirlas de manera inalámbrica a través del mundo gracias a su base de datos que es la más grande del mundo, permitiendo que actualmente 1.5 mil millones de aplicaciones sean descargadas mensualmente por sus usuarios.

En 2003, Andy Rubin, Rich Miner, Nick Sears y Chris White fundaron **Android Inc.** en Palo Alto, California, pasando desapercibida más que como una pequeña empresa dedicada al desarrollo de software móvil, hasta que en el 2005 Google compró la pequeña empresa, aunque, sin tener aún algo sólido hasta que en 2007 se funda **OHA (**Open Handset Alliance**)** que era una alianza entre varias empresas con relación a la telefonía celular y telecomunicaciones, dentro de **OHA** se encuentran empresas como LG, Qualcomm, Intel, T-Mobile, Sprint Nextel, Moto, Texas Instruments o HTC, Samsung entre otros, en la cual Google sería el líder [3], cuya finalidad es el desarrollo de código libre basados en el kernel de Linux, ese mismo año se da a conocer la primera versión de lo que hoy conocemos como **Android** con su versión **1.0 Apple Pie.** [4]

Agregado a esto, Google ofrece la posibilidad de distribuir y/o vender tu aplicación de Android por medio de ***Google Play*** que permite monetizar ya sea cobrando por descarga o por usuarios.

De manera sencilla se puede programar usando como lenguaje base JAVA con una gran cantidad de bibliotecas a disposición pública. [2]

En la actualidad, los dispositivos Android disponen de una gran cantidad de recursos físicos para mejorar la experiencia de uso, sensores de movimiento, pantallas adaptables, wifi, bluetooth, GPS, pantalla táctil, sensores de posicionamiento, módulo de tarjeta sim para el uso de redes móviles, así como, materiales más ligeros y resistentes.

2.1.1Apple Pie 1.0

Basado en el *Kernel* de Linux 2.6 es el primer sistema operativo móvil gratuito y de código abierto, algunas de las características que incluía y que se vería años después en sistemas operativos como un estándar es:

- Menú desplegable de notificaciones
- *Widgets* de escritorio
- **Android Market,** tienda de apps totalmente gratuita.
- Integra Gmail.
- Navegador, Maps y reproductor de Youtube.

Figura 7 Logo de Android 1.0 apple pie

2.1.2 Banana Bread 1.1

Unos meses después, en febrero del 2009, se lanza a nueva versión de Android que, a excepción de la actualización automática del software, no presenta ninguna novedad pero corrige muchos errores anteriores.

Figura 8 Logo Android 1.1 Banana Bread

2.1.3 Cupcake 1.5

Con la velocidad a la que crece Android y la demanda de los usuarios para tener equipos con mayor eficiencia y capacidades, en abril del mismo año sale la nueva versión llamada **Cupcake** la cual no presenta cambios importantes en la interfaz pero si llega con novedades como lo son [4]:

- Teclado Táctil QWERTY con predicción de texto
- Widget de escritorio de Google para búsquedas directas
- SDK para el desarrollo de widgets de escritorio por parte de terceros
- Funciones del portapapeles ampliadas
- Interfaz para grabar y reproducir vídeos mejorada
- Bluetooth

Figura 9 Logo Android 1.5 cupcake

2.1.4 Donut 1.6

Para septiembre del mismo año se recibe la actualización Donut que sería la 1.6 de muchas que aún no se imaginaba. La interfaz sufre cambios visuales, pero lo importante de esta actualización sería la mejora en el núcleo del Sistema Operativo:

- Se amplía el alcance a los mercados con la adaptación de soporte para DMA/EVDO, 802.1x y VPN.
- Compatible con distintas resoluciones de pantalla.
- Actualización y nuevo diseño del Android Market.
- Rediseño de la interfaz de la aplicación de cámara de fotos.
- Utilidad de búsquedas facilitada en el dispositivo.

Figura 10Logo Android 1.6 Donut

2.1.5 Eclair 2.0

En noviembre, a escasos dos meses de la versión anterior Android decide lanzar esta bomba que representa cambios tanto a nivel de diseño como a nivel de arquitectura interna. Era una versión dirigida a dispositivos de mayor tamaño y que representaba las características:

- Soporte de múltiples cuentas de usuario.
- GPS, Google Maps y navegador gratuito.
- Soporte para más resoluciones de pantallas.
- Navegador predefinido actualizado con HTML5.
- Función ***Text to Speech*** para realizar dictados al dispositivo móvil.
- Zoom Digital.

Figura 11Logo Android 2.0 Eclair

2.1.6 Froyo 2.2

En Mayo del 2010 sale esta nueva versión de Android conocída como ***Froyo*** la cual sería la ***2.2*** la cual llega con una gran diversidad de actualizaciones, algunos similares y otros que marcarían una definición de lo que sería la línea corporativa de la marca Android.

- La pantalla "*home"* con 5 paneles en lugar de 3.
- Nueva galería de imágenes
- Soporte para actuar como hotspot para otros dispositivos, *tethering* de datos
- Soporte para Flash 10.1
- Desbloqueo mediante *PIN*.
- Mejora de velocidad de procesamiento.
- Motor JavaScript V8 de Chrome usado en el Browser

Figura 12Logo Android 2.2 Froyo

Ese mismo año salen al mercado las primeras tabletas con sistema operativo Android como lo serían las proporcionadas por **Samsung**® y se esperaba que fuera la competencia directa del gigante de la manzana.

2.1.7 GingerBread 2.3

A finales de diciembre del mismo año sale esta nueva versión de Android, la cual suponía una continuación en sus políticas de actualizaciones acompañada de una gran gama de teléfonos celulares nuevos, Gingerbread se convertiría en la versión de Android más extendida durante los siguientes años [4] [5]. Con esta versión vienen nuevas actualizaciones interesantes:

- Mejora estética completa desde la pantalla de inicio hasta el menú de ajustes.
- Compatible con resoluciones mayores.
- Opción de copiar y pegar mejorada.
- Teclado de pantalla mejorado.
- Tecnología NFC.
- Herramientas de visualización de consumo y uso de batería.
- Cámara frontal.
- Acceso de bajo nivel para desarrolladores de juegos.
- Sustitución del sistema de archivos YAFFS por ext4.

Gingerbread llegó de la mano de los celulares sin “mouse” de bola de navegación externa, convirtiéndose totalmente en táctil, aquí nace la serie de celulares ***SAMSUNG GALAXY ® serie S*** que se convertiría en la competencia directa del Iphone [3].

Figura 13Logo Android 2.3 Gingerbread

2.1.8 Honeycomb 3.0

Para principios del año 2011 y con el mercado de las tabletas creciendo con gran velocidad, **Android** se ve en la necesidad de lanzar una actualización exclusiva para las mismas, la cual contenía cuestiones de diseño más que de corrección de errores.

- Pantalla de Inicio rediseñada.
- Inclusión de tonos azules en la interfaz en lugar de los tonos verdes.
- Nuevas funciones para el uso de widgets.
- Fin de los botones físicos.
- Actualización automática.
- Multitarea actualizada.
- Aceleración gráfica.
- Soporte para periféricos USB.

Figura 14Logo Android 3.0 HoneyComb

Con su nueva interfaz gráfica trataba que los usuarios pudieran disfrutar de una experiencia en tercera dimensión, gracias también a su procesador destinado para la renderización de los gráficos 3D.

2.1.9 Ice Cream Sandwich 4.0

Meses después y con la gran aceptación de su versión **Honeycomb,** llega el turno de los *smartphones,* y para no errarle, llega **Ice Cream Sandwich** al mercado, la cual es una adaptación de su versión para tabletas con unas cuantas modificaciones muy precisas.

Ese año superó la cuota de mercado de BlackBerry y se convirtió en el sistema operativo móvil más usado del mundo y se considera como el momento en que Android adquirió la fuerza necesaria para ser el sistema operativo por excelencia. [4] [5] [3]

- Nueva fuente tipográfica Robot.
- Interfaz *Holo.*
- Sistema de Notificaciones mejorado.
- Multitarea actualizado.
- Diccionarios para teclado virtual.
- Nuevas funciones y nuevo diseño de la pantalla principal.
- Android Beam, funcionalidad de transferir datos entre dispositivos vía NFC.
- Seguridad mediante Biometría facial.
- Nuevas funciones para la gestión de datos.
- Nueva forma de captura de pantalla vía botones de volumen y el de encendido.
- Soporte MKV.
- Lápiz táctil.

El diseño de los dispositivos crea la línea maestra para diseños futuros, Android había ido formando una base sólida hasta este momento, donde se crea una personalidad y un camino donde ya se han conseguido seguidores y un público que confía en ellos.

Figura 15 Logo Android 4.0 Ice Cream

2.1.10 Jelly Bean 4.1

En julio del 2012 se presenta una nueva versión que llevaría por nombre **Jelly bean,** donde se reformula la estrategia de Android en las tabletas, a primera vista no presenta cambios tangibles, pero sí muy interesantes:

- Desaparece el soporte para Flash Player.
- Sistema de detección de entrada de datos táctiles.
- Estreno de Google Now, el asistente de voz inteligente de Google.
- Navegador de Google Chrome.
- Mejoras en la corrección ortográfica y la predicción del teclado.
- Dictado de voz offline.

Como estos cambios no eran tan notorio, por lo que, no tardaron en sacar una nueva versión **4.2** con el mismo nombre, pero con más que solo actualizaciones y mejoras a los errores.

- Rendimiento mejorado
- Nuevas animaciones
- Nuevo panel de control
- Acceso a widgets y cámara fotográfica desde la pantalla de bloqueo
- Photosphere, captura de fotografías panorámicas de 360º
- Gestual Mode para personas invidentes

Para marzo del 2013 la división de Android Google cambia de dirección y Sundar Pichai sustituye a Andy Rubin, y ahora la misma persona que dirige el desarrollo de Chrome y de Chrome OS.

Para Julio del mismo año Google anuncia la nueva versión 4.3 que mantiene el mismo nombre de **Jelly Bean,** que con las nuevas actualizaciones espera consolidar a Android como un *SO* capaz de correr juegos.

- Soporte Multiusuarios.
- Soporte *OpenGL ES 3.0.*
- Compatible con *TRIM.*

- Smart Bluetooth.
- Plataforma Google Games
- Mejores servicio de localización por wifi.

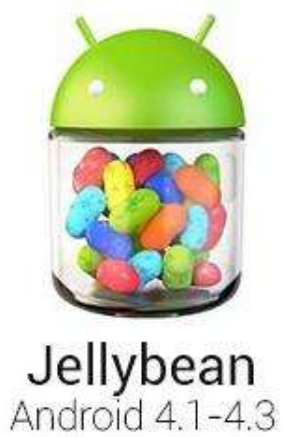


Figura 16 Logo Android 4.1 JellyBean

2.1.11 KitKat 4.4

Se trata de una nueva versión que ofrece al usuario nuevas experiencias y a la par trata de corregir la mayor falla que tiene Android hasta este momento, cada nueva versión que sacaba requería nuevas especificaciones físicas lo cual obligaba a las empresas a crear nuevos dispositivos para soportarlo y con esto muchos usuarios no tenían la opción de actualizarlo.

El nombre se debe a una alianza comercial entre Google y el gigante de productos alimenticios **Nestlé®,** es la primera vez hasta el momento en que se hacía una alianza con una empresa externa, sin embargo tiene cosas que a muchos que ya eran fans no empezaron a gustarles.

- Rebaja de requisitos hardware para corregir la fragmentación de versiones.
- Compatible con memoria ram de 512MB.
- Reducción del consumo de batería mediante la optimización de los sensores.
- Servicios de almacenamiento Online integrados (Google Drive).
- Soporte Infrarrojo para usar el celular como control de TV.
- Soporte Bluetooth HID a través de *GATT*.
- Captura de pantalla en video.

Figura 17 Logo Android 4.4 KitKat

2.1.12 Lollipop 5.0

Para 2014 Android se había expandido de una manera que ya contaban con todo tipo de servicios, Smartphones, Smartwatches, Android Auto y algunos otros dispositivos de entretenimiento.

Para octubre del mismo año se da a conocer que la nueva actualización para todos sus dispositivos sería **Android 5 Lollipop** el cual poseía como rasgo más notorio la inclusión de ***Material Design,*** que es un nuevo lenguaje de diseño con la finalidad de unificar la experiencia de uso en sus distintos dispositivos.

- Interfaz que se adapta a cualquier tamaño.
- Mejoras en lo visual que proporcionan una experiencia más real.
- Nuevo sistema de notificaciones.
- Vista multitarea.
- Aumento del rendimiento energético.
- Función *Android Smart Lock*, que permite emparejar un dispositivo Android con otro, ya sea un reloj inteligente o un automóvil.
- Modo "Invitado" para que otros usuarios accedan a tu teléfono pero no a tu información privada.
- El entorno de ejecución de aplicaciones *ART* (*Android RunTime*) pasa a ser el predeterminado.
- Soporte para sistemas de 64 bits.

En diciembre de ese mismo año aparecían las primeras actualizaciones del sistema (5.0.1 y 5.0.2) enfocadas a la corrección de errores. Android 5.0.1 solucionaba un problema que se producía tras varios intentos fallidos al desbloquear la pantalla del dispositivo. También incluía mejoras en la seguridad Wi-Fi. Mientras que Android 5.0.2 corregía la disminución del rendimiento de los dispositivos cuando se reiniciaban, mejora la gestión de las tarjetas de almacenamiento externo y problemas en la aplicación Alarma, pero no fue hasta marzo del 2015 que por fin salió a la luz la nueva versión de **Lollipop** 5.1, con la que se mejoraba el rendimiento del procesador y traía algunas novedades.

- Soporte Nativo para múltiples SIMs.
- Acceso a opciones de Wi-fi y Bluetooth desde los ajustes rápidos.
- Mejora la gestión de la memoria RAM.
- Nuevas animaciones para la barra de ajustes rápidos y el reloj.
- Soporte para llamadas de alta calidad.

En abril del mismo año llega **Lollipop 5.1.1** que sería la última actualización de esta versión, la cual mejoraba la velocidad de respuesta del dispositivo.

Figura 18 Logo Android 5.0 Lollipop

2.1.13 Marshmallow 6.0

En agosto del mismo año se anuncia la versión 6.0 **Marshmallow,** también conocido como **Android M**. Con diversas actualizaciones como el cambio del cable cargador que se consideraba universal.

- Soporte USB tipo-C.
- Soporte para biometría dactilar.
- Desinstalación rápida de las aplicaciones desde la pantalla de inicio.
- Nueva Gestión de energía con el sistema ***DOZE*.**
- Mayor duración de batería en modo reposo.
- Android Pay que usaría el chip NFC.
- Mejoras de Google Now.
- Control avanzado para los permisos de las aplicaciones.

Figura 19Logo Android 6.0 Marshmallow

2.1.14 Nougat 7.0

Para mayo del 2016 la empresa de **Mountain View** anuncia la llegada de la nueva versión de Android llamado **Nougat 7.0** o **Android N,** siguiendo con el orden alfabético de las versiones que han salido.

Como era de esperar viene acompañado de actualizaciones interesantes, en especial de diseño.

- Rediseño del botón de apps recientes.
- Mejores notificaciones para organizarlas mejor.
- Compatibilidad con más de 100 idiomas.
- Función Multiventana nativa para ejecutar dos apps al mismo tiempo en la misma pantalla.
- Mayor personalización.
- Forma inteligente de bloquear llamadas entrantes.
- Acceso a ***Instant Apps*** sin necesidad de descargarlas ni instalarlas.
- Ahorro de datos inteligente con posibilidad de bloquear consumo de aplicaciones en segundo plano.

Figura 20Logo Android 7.0 nougat

2.1.15 Oreo 8.0

Para inicios del segundo semestre del año 2017 Google Anuncia el nuevo **Android Oreo 8.0** o **Android O** continuando como se había mencionado antes, con el orden alfabetico, presenta nuevo rendimientos y actualizaciones enfocadas a la eficiencia del dispositivo. [5]

- El doble de velocidad en respuesta a las solicitudes del usuario.
- Limitación de las aplicaciones en segundo plano que no uses, para mayor eficiencia.
- Autocompletado de los campos (Con permiso del usuario) para cuentas de usuario.
- Visualización de dos aplicaciones al mismo tiempo.
- Modo Claro y oscuro instantáneo.
- Visualizar estado de batería de los dispositivos Bluetooth.
- Acceso directo a notificaciones sin necesidad de abrir las aplicaciones.
- Puedes poner tus tonos y alarmas de manera sencilla por primera vez.

Figura 21Logo Android 8.0 Oreo

2.2 Uso de Android

Para el correcto uso de esta herramienta es necesario saber que el lenguaje de programación necesitado es *Java.* Es necesario instalar el kit de desarrollo *Android Software Development Kit* (Android SDK), el cual es gratuito y es provisto por la empresa Google.

El SDK de Android® incluye un emulador de un dispositivo inteligente el cual sirve para "interactuar" con nuestro programa recién creado y ver los cambios gráficos que se realizarán, es limitado pues no se puede interactuar con las funciones de comunicaciones, ni de obtención de imágenes, consume muchos recursos del equipo de cómputo, que lo vuelve lento y con algunas fallas, para solucionar esto fue necesario hacer uso de un

dispositivo móvil físico. Cada que se modifica el programa es necesario instalarlo al dispositivo y verificar el desempeño, es un proceso relativamente lento pero eficaz respecto al desempeño del emulador incluido.

2.3 Estructura de Android

Android® utiliza un **kernel** basado en Linux, que permite que sea de código libre, en la **Figura 22** se aprecia la estructura de Android, tiene 5 capas: kernel, bibliotecas, máquina virtual, marco de aplicación y aplicaciones.

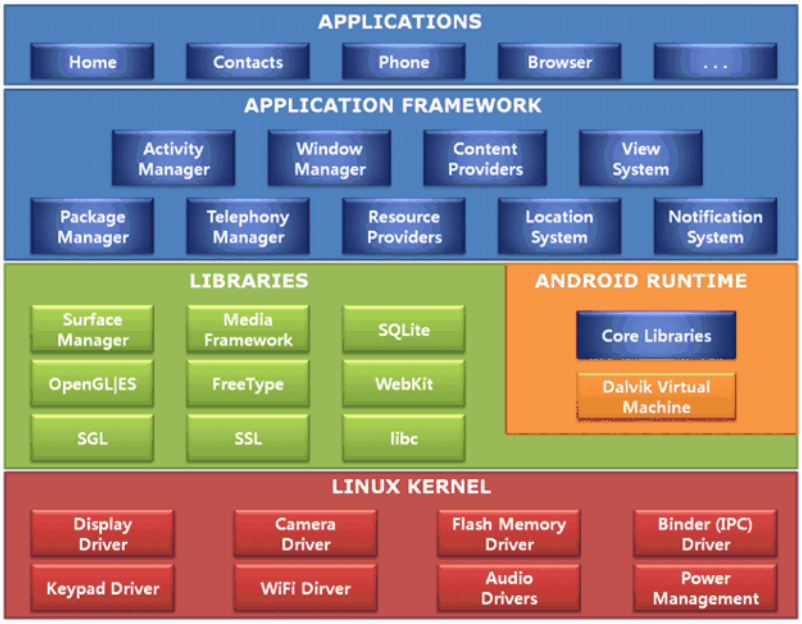


Figura 22Diagrama de la estructura de Android

Kernel: En esta capa se encuentra el corazón de Android: el manejo de memoria, procesos, drivers y manejo de energía, y es donde existe la comunicación entre el hardware y el software. Esta capa al ser tan genérica ayuda a no tomar demasiada importancia por las características de cada dispositivo móvil que existe en el mercado, lo podemos hacer usando los protocolos correctos que nos proporciona este kernel, así mismo, al programar en Android, es necesario seleccionar una versión de Android, ya se mencionó en el **Capítulo 2** algunas características de las distintas versiones que existen actualmente, es de entenderse que entre más reciente sea la versión, menor cantidad de usuarios cuentan con dicha versión, si se busca que la aplicación sea útil a un gran número de usuarios, es necesario que la versión tenga al menos un año de haber salido al mercado, de esta manera los usuarios con dicha versión y a su vez, los usuarios con esta versión podrán tener acceso a la aplicación, Android es retroactivo, por lo que una versión actual puede acceder a diversos recursos de versiones pasadas, sin embargo, no funciona de manera inversa, un dispositivo viejo no puede hacer uso de recursos nuevos.

Blibiotecas: Esta capa contiene las bibliotecas nativas de Android, las cuales están codificadas en C o C++, tienen tareas específicas como:

- *Surface manager*: gestión del acceso a la pantalla.
- *Media Framework*: reproducción de imágenes, audio y vídeo.
- *SQLite*: bases de datos.
- *Webkit*: navegador.
- *SGL*: gráficos 2D.
- *OpenGL*: gráficos 2D intensos y gráficos 3D.
- *Freetype*: renderización de vectores o imágenes.

Se tiene acceso a ellas, y permiten disponer de sus utilidades de manera muy sencilla, solo conociendo el funcionamiento de las bibliotecas y como llamarlas dentro de nuestros programas, esto no quiere decir que sean las únicas bibliotecas que se pueden usar, solo indica las bibliotecas adheridas a los dispositivos Android, inclusive algunas bibliotecas externas, utilizan las nativas como soporte.

Máquina virtual: En esta capa se tiene la máquina virtual de Android, llamada, *Dalvik*, esta máquina virtual es distinta de la de Java, conocida como *JVM*, esta máquina virtual es la que ejecuta las aplicaciones en el celular, su lenguaje de programación es JAVA, *Dalvik* de cada dispositivo Android, cada fabricante puede usar el procesador que más desee.

Algunas de las características de *Dalvik* son:

- Trabaja en entorno con restricción de memoria y procesador.
- Ejecuta el formato .dex.
- Convierte .class en .dex.

Marco de aplicación: Para el desarrollador esta es la capa que se encuentra palpable, es donde se encuentran la mayoría de los componentes para el desarrollo, como son:

- *Activity Manager*: rige las actividades de nuestra aplicación y el ciclo de vida.
- *View System*: administra la interfaz gráfica e interacción con el usuario.
- *Notification System*: dispone las notificaciones.
- *Telephony Manager*: administra telefonía, llamadas, mensajes.
- *Resource Provider*: gestiona los recursos de la aplicación, como los xml, imágenes, sonido.
- *Location System*: gestiona la posición geográfica.

Aplicaciones: Aquí se encuentran las aplicaciones que vienen integradas con el dispositivo móvil: reproductor de música, gestor de correos, mensajes, galería, reloj, etc.

2.4 Aplicación y ciclo de vida

Una aplicación es administrada según las actividades (*Activity*) que tenga, las actividades son cómo el usuario interactúa con la aplicación, esto quiere decir, que cada ocasión que se abren distintas ventanas en una aplicación, tiene distintas actividades y es necesario programar una por una en forma de capas, estas capas permiten tener prioridad en el funcionamiento y en las opciones del usuario. Cada actividad puede tener opciones gráficas y otras en cuestión de software no grafico al que el usuario no puede acceder, el cual es solo para el desarrollador.

Para que una actividad sea eficiente, es necesario un orden, ese orden es denominado ciclo de vida, va desde que se inicia una actividad hasta que se cierra, pasando por el estado inactivo. Cada actividad es administrada como *activity stack* (pila de actividades). Cuando una nueva actividad es lanzada, se coloca en la cima de la pila y se convierte en la actividad ejecutada, la actividad anterior queda en un estado de reposo hasta que la nueva actividad finaliza.

La actividad tiene al menos cuatro estados:

- Si la actividad está visible, se encuentra **activa**.
- Si la actividad ha perdido el foco pero aún es visible, está **pausada**. Una actividad en pausa está completamente viva (mantiene el estado e información de sus miembros y se encuentra ligada al administrador de ventanas), pero puede ser finalizada por el sistema en situaciones críticas de baja memoria.
- Si la actividad es completamente cubierta por otra actividad, se encuentra **detenida**. Todavía retiene el estado e información de sus miembros, sin embargo, no se encuentra visible al usuario y su ventana se oculta, típicamente será finalizada por el sistema cuando éste necesite memoria.
- Si la actividad es **pausada o detenida**, el sistema puede quitarla de la memoria finalizándola, o bien pidiendo al usuario que la finalice. Cuando esta actividad vuelva a ser visible, será reiniciada y restaurada a su estado anterior.

Cada actividad consume los limitados recursos del dispositivo móvil, desde la energía, hasta la memoria y procesador, por lo que es necesario colocarla en los estados correctos para administrar de manera óptima nuestro entorno. En la **Figura 23** se visualiza

las rutas de los estados de una actividad. Los rectángulos representan métodos *callback* que se puedan implementar para realizar ciertas operaciones cuando la actividad se mueve entre estados, mientras que los óvalos representan estados importantes en los que la actividad puede encontrarse.

Figura 23Estados de la actividad de una aplicación en Android

Cuando se abre una aplicación se ejecuta el proceso `onCreate()` la cual inicia el ciclo de la aplicación, acto seguido, se ejecuta `onStart`() que es el preámbulo a la visualización de la aplicación, la cual se da en `onResume`(), en ese momento la aplicación ya está ejecutada correctamente y se puede visualizar, si se minimiza la aplicación o se coloca en segundo plano, entra a `onPause`(), en esta opción se puede reiniciar la app desde el inicio o, reanudarla donde se había quedado antes de minimizar, si por el contrario, se quiere dejar de usar la aplicación y se cierra definitivamente, entra a `onDestroy`() que se encarga de eliminar el cache de la aplicación, de cerrarla correctamente y guardar los archivos que se pueden guardar.

Es importante mencionar que los métodos `onStop()` y `onDestroy()` no siempre son llamados, por lo que cualquier información que quiera preservarse deberá hacerse en el método `onPause()`.

2.5 Vista y aplicación

Los objetos conocidos como "***view"*** se emplean de manera específica para dibujar contenido en la pantalla del dispositivo Android

2.5.1 Vista (XML)

Android Studio permite tener diversas maneras de tener una vista, la más sencilla es desplazarnos en la pestaña de "activity_main,xml" la cual tiene dos opciones:

1. Diseño.
2. Texto.

En la primera opción es una manera sencilla, basta con arrastrar los elementos "view" hasta la representación de la pantalla de un dispositivo móvil, en ella podremos ver las referencias de posición y tamaño del objeto que estamos arrastrando, de manera gráfica podemos alterar su tamaño y posición.

En la **Figura 24** se aprecia como el objeto tiene una posición y tamaño predeterminado, su posición dependerá del "***layout"*** que se use en la pantalla del dispositivo, esta define la distribución de los elementos, los ***layouts*** se añaden de la misma manera, ya que, son objetos ***view*** al igual que los botones, imágenes, barras etc.

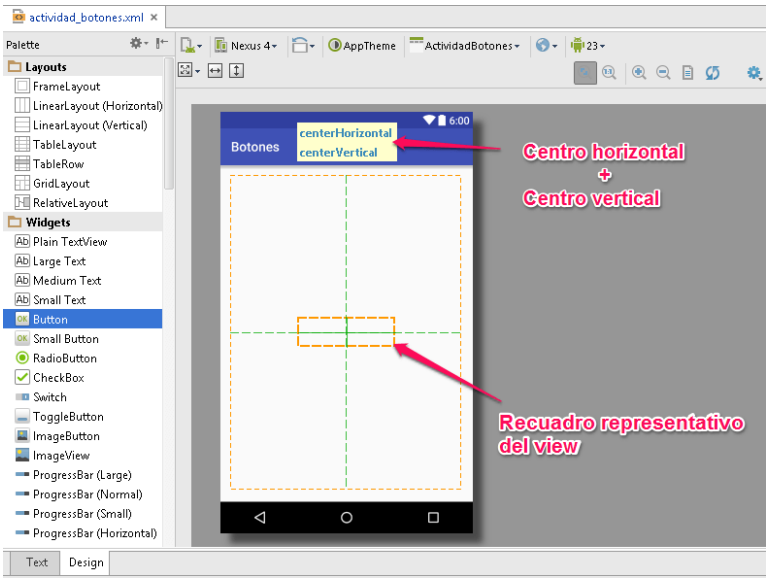


Figura 24 Representación de una vista en Android Studio

Si se tienen problemas para modificar los objetos de manera gráfica, al seleccionar un objeto se despliega en la parte derecha de la pantalla del computador, una lista con opciones que hacen referencia a las características del objeto seleccionado, se puede cambiar características como:

- Posición.
- Efecto de gravedad.

- Orientación.
- Color del fondo.
- Opciones si se hace click sobre el objeto.
- Estilo.
- Ancho.
- Alto.
- Texto en caso de tenerlo.
- Accesibilidad.

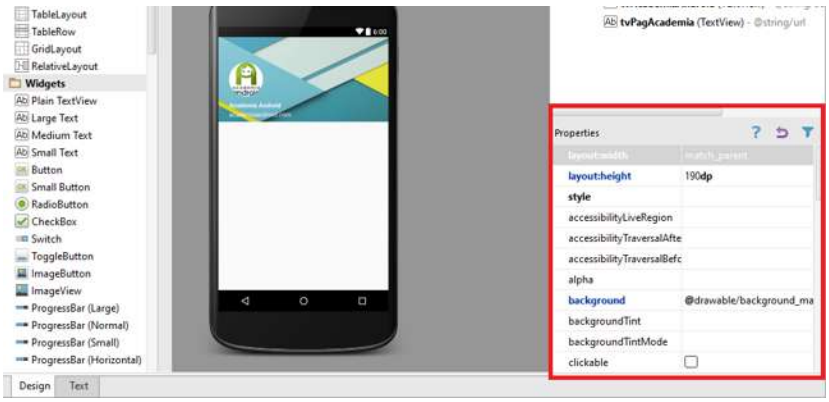


Figura 25Ejemplo de la lista de opciones que aparecen al costado derecho de la pantalla

De igual manera, si se tiene mayor experiencia en programación, es posible elegir la segunda opción, la selección de texto, como se aprecia en la **Figura 26** es posible modificar directamente el tamaño, posición, color, texto a colocar e ir colocando más objetos desde esta ventana, cada que modificamos la pestaña de diseño, se agrega un segmento de código al texto, mientras que si se inserta el texto, se agrega de manera inmediata en la sección de diseño.

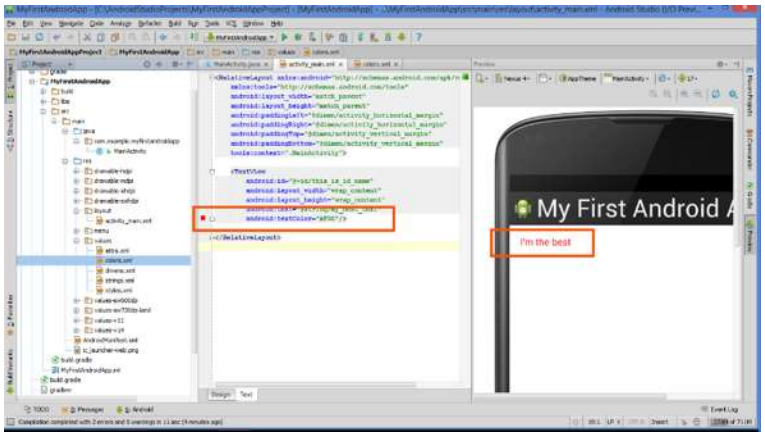


Figura 26 Vistas de android Studio en XML por medio de texto

Usando esta forma de programación tenemos más control en las características generales de los objetos, sumado a que requiere menos procesamiento sin la parte gráfica.

2.5.2 Aplicación (JAVA)

Con las vistas de XML se diseña la interfaz de usuario de la aplicación, y aunque se pueden realizar vistas en codigo de JAVA en la aplicación, no es la mejor manera, sin embargo esta parte representa el verdadero reto de la aplicación.

Como se aprecia en la **Figura 27** en la pestaña **main activity** son creadas todas las actividades que tendrán las aplicaciones, por ejemplo, si se requiere que la aplicación tome fotos, es aquí donde se hace, cada acción que se realice será nombrada ***activity,*** una aplicación puede tener varias actividades, una primaria y diversas secundarias. La actividad primaria es aquella que inicia cuando el usuario selecciona la aplicación, las actividades secundarias son aquellas que el usuario elige dentro de la función principal de la app.

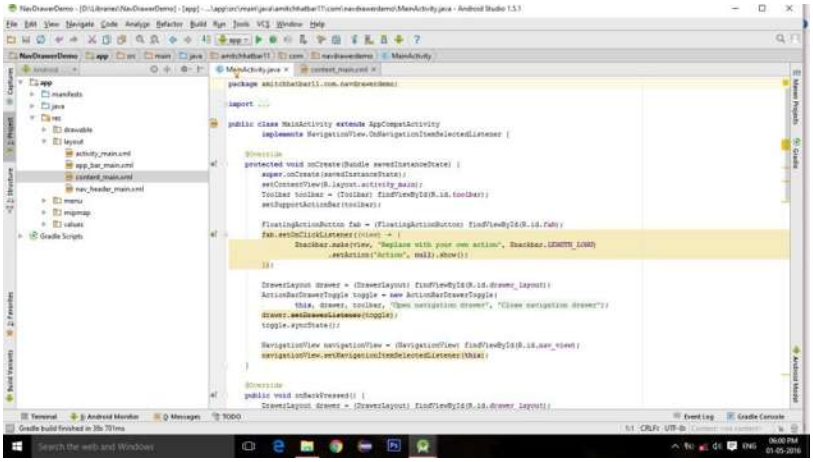


Figura 27Main Activity de la aplicacion

Como se vio en **2.4,** las aplicaciones tienen un ciclo de vida, lo mismo sucede con las actividades, cuando se abre una actividad, es necesario que se realice un **onCreate(),** que se inicie y que sea visible, a su vez, es necesario que se pause o se destruya la actividad cuando se deja de usar.

En esta parte de la aplicación, es necesario que se cree un puente de conexión entre el XML y JAVA, pueden existir de manera independiente sin interacción, pero la aplicación no realizara ninguna acción.

Para crear este puente es necesario crear instancias, las instancias es donde creamos un objeto en JAVA que tendrá las propiedades que se le otorgaron a un objeto en el XML, las vistas como los botones, los textos o hasta la captura de imágenes, cada objeto que coloquemos en XML tiene un ID único que lo reconoce, de esa manera podemos hacer uso de sus propiedades en un objeto en java por medio de la función ***get.ID.***

Se puede hacer uso de funciones exclusivas para cada botón que se presione en la vista del dispositivo, solo es necesario otorgar la función ***onClick*** la cual indica en la vista

que al hacer uso de ese recurso, hará un ***callback*** a una función especial que el programador escoja.

En este apartado se realizan todas las funciones, incluyendo aquellas en las que el usuario no tiene interferencia, como los protocolos de conexión, los casos a realizar cuando se conecta o desconecta un dispositivo, se hace uso de los recursos del dispositivo móvil a los cuales se les otorguen permisos.

Se puede hacer uso de recursos de programación básica como comparaciones, ciclos, funciones, contadores, cadenas, vectores etc. Lo que sea necesario para el correcto funcionamiento de nuestra aplicación.

Es necesario en este apartado el entendimiento del ciclo de vida de una aplicación, es de suma importancia que las funciones sean creadas de manera correcta y que se destruyan al finalizar, de lo contrario la aplicación consumirá recursos del dispositivo móvil en segundo plano, con la posibilidad de crear archivos basura y consumiendo capacidad de batería.

2.6 Permisos

Para que las aplicaciones funcionen de la mejor manera, deben utilizar recursos que el celular ya ofrece, con la finalidad de no tener que usar dispositivos extra y se explote el potencial al máximo de los smartphone.

Google velando por la seguridad, no permite que las aplicaciones que se instalen en el celular use de manera indiscriminada cualquier recurso que requiera, solamente solicita un permiso al administrador del dispositivo móvil para usar dichos recursos y es responsabilidad de cada usuario el leer que recursos ocupa y, decidir otorgar el permiso o no.

Hasta antes de la versión 6.0 de Android, al aceptar la instalación de una aplicación, accedías a todos los permisos que se solicitaba, de lo contrario, no se podia hacer uso de la aplicación, lo cual era injusto y unilateral, pero con la llegada de la versión 6 se permitió al usuario más control de estos permisos de manera individual para que este se sienta en total confianza, seguridad y control de los recursos que permite utilizar, y si una petición no le parece adecuada puede declinarla sin tener que dejar de usar la aplicación.

Los permisos que se pueden solicitar son los siguientes:

- Almacenamiento
- Calendario
- Contactos
- Cámara
- Micrófono
- SMS
- Wi-Fi

- Sensores corporales
- Sensores de ubicación espacial
- Teléfono
- Configuración del Dispositivo
- Ubicación
- Galería
- Redes sociales
- Correo electrónico
- Cuentas bancarias (En caso de tener instaladas)

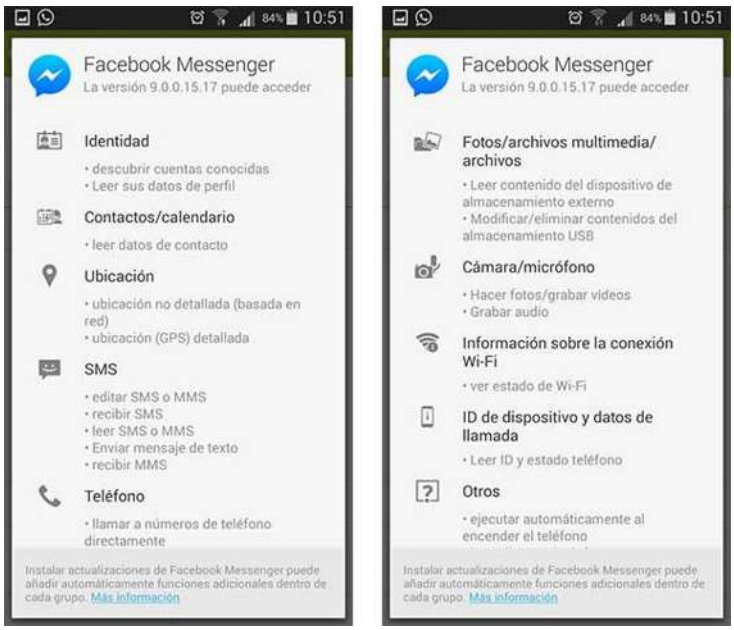


Figura 28Ejemplo de los requerimientos de una app para su funcionamiento

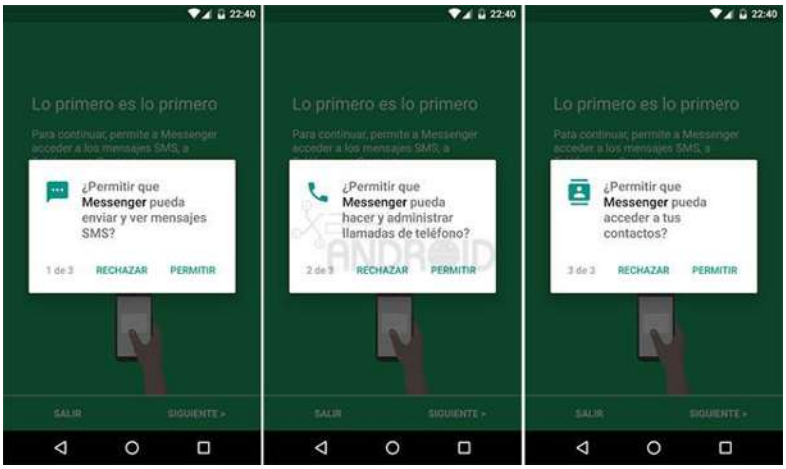


Figura 29Ejemplo de las peticiones para que el usuario otorge los permisos necesarios

2.6.1 Permisos propios de la aplicación

Como ya se explicó en **2.2 ,** Android tiene un método de seguridad permite al usuario el uso de los recursos del celular y el usuario debe aprobarlos para que estos se instalen y puedan usarse, esto se programa desde la aplicación y cada una debe tener los recursos necesarios para su correcto funcionamiento, si se usa más o menos recursos del equipo puede afectar el rendimiento del dispositivo.

Al instalar una aplicación, se muestra al usuario una lista con los permisos que requiere la aplicación y el usuario puede decidir entre proceder con la instalación o cancelarla

Para el desarrollo *tracking* se necesitan los siguientes permisos para los siguientes recursos:

- `android.permission.BLUETOOTH`: necesario para establecer comunicación Bluetooth con los dispositivos sincronizados, el cual es necesario para la comunicación entre el dispositivo móvil y el vehículo a controlar.
- `android.permission.BLUETOOTH_ADMIN`: necesario para encender y apagar el adaptador Bluetooth, escanear nuevos dispositivos y para sincronizarse con ellos.
- `android.permission.CAMERA`: necesario para que el dispositivo utilice la cámara integrada y se visible en la actividad correspondiente.

Para usar algunas funciones extras de la cámara es necesario dar permisos de hardware adicionales como lo son:

- `android.hardware.camera`: para permitir al programa usar las funciones adicionales.
- `android.hardware.camera.autofocus`: permite que la cámara enfoque de manera automática.

Es importante señalar que si no se declaran los permisos necesarios para una determinada acción, al ejecutarla, la aplicación se cerrará debido a la falta de permiso del recurso.

2.7 OpenCV

Figura 30Logo de la biblioteca OpenCV

Open Source Computer Visión Library, una biblioteca de código libre para visión computacional, es desarrollada por **Intel®** en el año de 1999 [19] y se lanza bajo una licencia **BDS (*Berkeley Software Distribution*)**, los aportes que le dio la universidad de Berkeley, es de uso gratuito para fines académicos y comerciales, tiene interfaces con *C++*, *Python, Java y Android*, a su vez se puede usar con casi todos los sistemas operativos actuales. [20]

Se diseñó con fines de eficiencia en la computación y con una fuerte tendencia para procesamientos en tiempo real explotando los beneficios de multinucleos de los computadores actuales, así como de los dispositivos móviles.

Gracias a la gran difusión y la gran comunidad de desarrolladores es que esta biblioteca evoluciona de gran manera siendo más eficiente con el tiempo, empresas bien desarrolladas como Google y Yahoo, entre otros, utilizan esta biblioteca, dando pie a muchos proyectos de emprendimiento como lo son Video Surf y Zeiteria [20], que da el ejemplo de lo bien desarrollada de la biblioteca.

2.8 Adquisición de Imagen

2.8.1 Visión

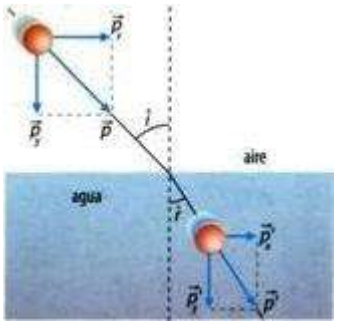
¿Qué es la luz?

Es una radiación electromagnética, cuya energía se trasfiere y mueve gracias a los fotones, aún crea muchos conflictos en la actualidad, sin embargo se habla de la dualidad de la luz.

Sir Isaac Newton defendía la teoría corpuscular de la luz, que dice que a luz tiene una naturaleza de partículas, las cuales, son tan pequeñas que tienen masa insignificante y viajan en línea recta, estas entran al ojo y estimulan la vista.

El argumento más fuerte para esta teoría es la reflexión en un espejo, Newton dice que esto se debe a que las partículas de la luz son elásticas y rebotan, cumpliendo así las **leyes de choque elástico,** como se aprecia en la **Figura 32**. [6]

Figura 31 Refracción de una partícula de luz al pasar de un medio a otro con distintas densidades



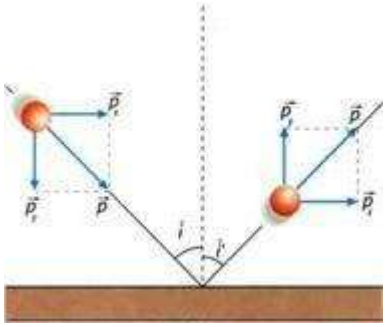


Figura 32 Reflexión de una partícula de luz al impactar con una superficie reflejante

Por otro lado, Christian Huygens apoyaba la teoría Ondulatoria de la luz, que dice que al contrario de la teoría corpuscular, que la luz no tiene naturaleza de partículas, sino, de onda, la cual decía que la luz era una Onda mecánica emitida por una fuente y que se desplaza por un medio muy elástico, impalpable y presente en todo, a lo cual llamaría ***éter.*** [6]

En la **Figura 31** se aprecia una onda pasando de un media a otro, cambiando su dirección y fuerza.

A diferencia de la teoría corpuscular, aquí la energía luminosa no está presente en cada fotón, sino que está a lo largo de toda la onda como se aprecia en la **Figura 33**

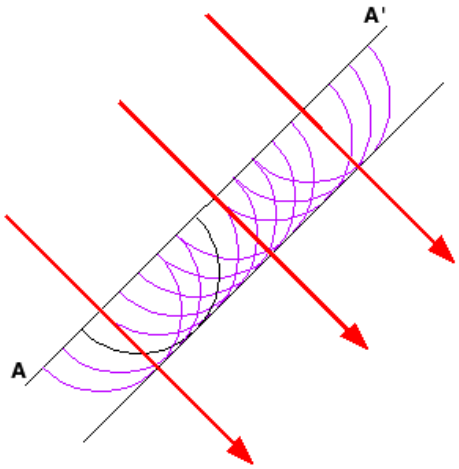


Figura 33 Teoría ondular de la dualidad de la luz

Se defendía esta teoría observando como al dejar caer un objeto en un líquido este generaba ondas en todas las direcciones haciendo que rebotaran e, inclusive, colisionaran

entre sí, si esto ocurriera con materia generaría una sombra al anteponerse materia sobre materia.

Para del siglo XX, **Louis De Broglie** de manera atrevida se lanza a decir que la luz tiene un comportamiento dual, es decir, que es onda y que a su vez es una partícula, a la que llamó "***Onda-Corpúsculo"***, basándose en Einstein y su fórmula de relación entre masa y energía, deduciendo así, su propia fórmula, la **Ecuación 1** es la ***Formula de Broglie:***

$$\lambda = \frac{h}{m \times v} \tag{1}$$

Ecuación 1 Formula de Broglie

Donde:

$\boldsymbol{\lambda}$ es la longitud de onda.
$\boldsymbol{h}$ es la constante de Planck.
$\boldsymbol{m}$ es la masa.
$\boldsymbol{v}$ es la velocidad.

Apareció un grave estado de incomodidad al encontrar que la luz se comporta como onda electromagnética en los fenómenos de propagación, interferencias y difracción y como corpúsculo en la interacción con la materia. [7]

Para finales del siglo XIX, a sabiendas que la velocidad de la luz es menor en el agua, (contrario a lo que Newton defendía) Maxwell desarrolló una serie de ecuaciones fundamentales del electromagnetismo, con lo cual pudo deducir que la luz representaba solamente una pequeña porción del espectro de ondas.

A esta pequeña fracción de las ondas electromagnéticas que se pueden ver se les conoce como ***espectro de luz visible, luz blanco*** o simplemente ***luz,*** el ojo humano es capaz de apreciar alrededor, de entre ***380 y 780nm.*** [8]

Y avanza de la luz ultravioleta a la infrarroja pasando por los colores que podríamos distinguir en el arcoíris como se muestra en la **Figura 34** medición del espectro de luz visible y su tamaño**.**

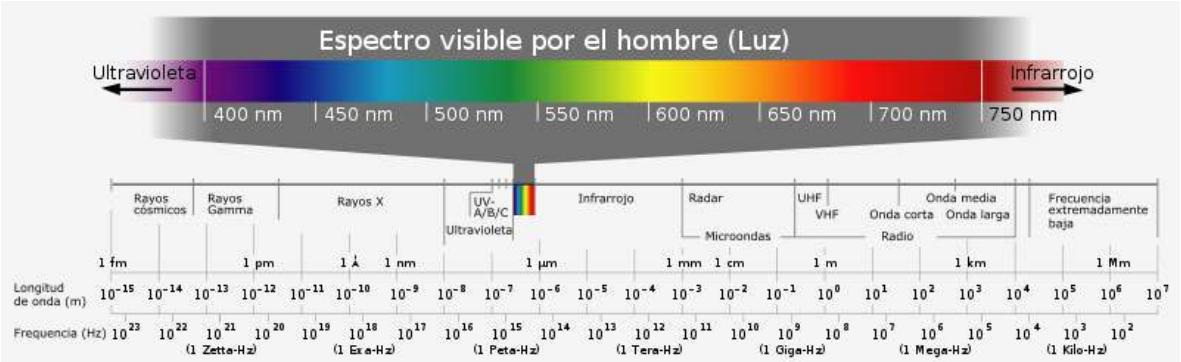


Figura 34 medición del espectro de luz visible y su tamaño

La longitud de onda más baja corresponde al violeta, y la longitud de onda más larga corresponde al rojo. La longitud de onda junto a su velocidad determina la frecuencia de cada onda y cuando más longitud tenga una onda, menor será su frecuencia.

Los estudios científicos de objetos basados en el espectro de luz que emiten es llamado **espectroscopia**, en un inicio Newton decía que este espectro, que veía al hacer incidir luz blanca por un prisma triangular, constaba de 7 colores, que eran el morado, azul, índigo, verde, amarillo, naranja y rojo, sin embargo, la **espectroscopia** actual dice que solamente se muestran 6 colores, todos los anteriores menos el índigo. [9] [8]

También se acostumbra agregar los colores **blanco** y **negro,** que no son colores propiamente dichos, el negro es la ausencia de cualquier color o luz, y el blanco es la unión de todos los colores.

Los dispositivos electrónicos de visualización a color que se conocen y utilizan diariamente como lo son los televisores, teléfonos inteligentes y tabletas, utilizan un principio básico para lograr su cometido, este principio básico se llama ***Teoría de color,*** la cual nos dice que existen tres colores primarios que son rojo, verde y azul, y la mezcla selectiva de estos generan el resto de los colores necesarios. [8] [9]

Existen distintos esquemas de colores, pero, esencialmente se usan dos, el esquema RGB y el CMY como el de la **Figura 35**, la diferencia de estos es sencilla, el primero se utiliza como se ha dicho antes, para dispositivos electrónicos basándose en la mezcla selectiva y armoniosa de los colores, donde la ausencia de color generan negro y la mezcla de todos genera blanco, por otro lado, el segundo se basa en su utilidad en la impresión, usando los colores **cian**, **magenta** y **amarillo,** donde de igual manera la mezcla selectiva produce el resto de los colores, en este esquema la mezcla de todos los colores genera negro y la ausencia de ellos genera blanco (considerando que el fondo para imprimir es blanco). [8]

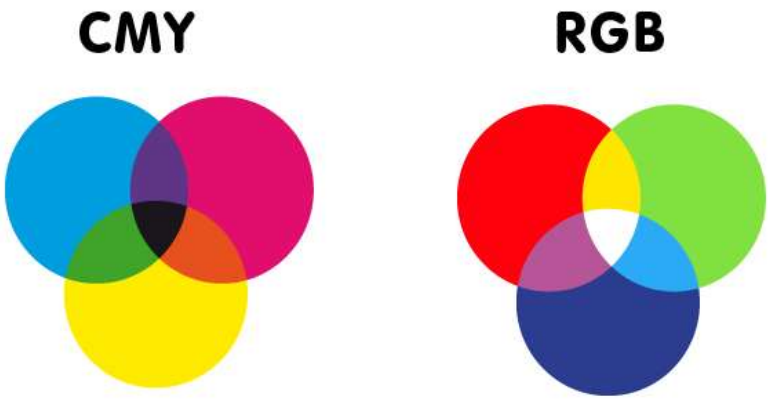


Figura 35Circulo Cromatico

2.8.2 Sensores de luz

¿Qué es un sensor?

Es todo aquel artefacto de instrumentación que permite detectar y medir una magnitud física, en la actualidad la mayoría de los sensores transforma estas magnitudes físicas en impulsos eléctricos para su procesamiento digital.

"*...Los sensores son componente esenciale de la automatización moderna, ya que las instalaciones deben detectar muchas magnitudes físicas. El trabajo de lossensores es de hacer legible las magnitudes físicas como presión, temperatura o fuerza, convirtiendo estas en señales eléctricas."* [10]

En el entendido de que existen diversos tipos de sensores, se harán mención solamente de aquellos sensores que detectan luz, conocidos como ***sensor fotoeléctrico,*** y aquellos que detectan haces de luz ***sensor óptico.***

Los sensores fotoeléctricos son aquellos que miden las variaciones de luz en un área determinada (depende de su colocación), y que podría usarse para medir si es de día o de noche o hasta para el apoyo en el control de iluminación de una casa habitación.

Su composición es básica, mejor conocido como ***fotorresistencia,*** la cual, como su nombre lo sugiere, cambia una resistencia en base a la cantidad de fotones que recibe, específicamente son resistencias cuyo valor de resistividad disminuye a medida que aumenta la energía luminosa incidente sobre ella y viceversa.

Un fotorresistor está hecho de un semiconductor de alta resistencia. Si la luz que incide en el dispositivo es de ***alta frecuencia***, los fotones son absorbidos por la elasticidad del semiconductor dando a los electrones la suficiente energía para saltar de la banda de valencia a la banda de conducción, aumentando así la conductividad del dispositivo y disminuyendo su resistencia.

Como podemos observar en la **Figura 37** se aprecia que al hacer incidir luz en la fotorresistencia, se presenta una excitación de los electrones y huecos, los cuales logran crear una conducción eléctrica y el circuito se cierra, si por el contrario, dejamos de inyectar este haz de luz, el circuito se abre.

En la **Figura 36** se aprecia la muestra de una fotorresistencia común en el mercado.

Figura 36 Fotorresistencia

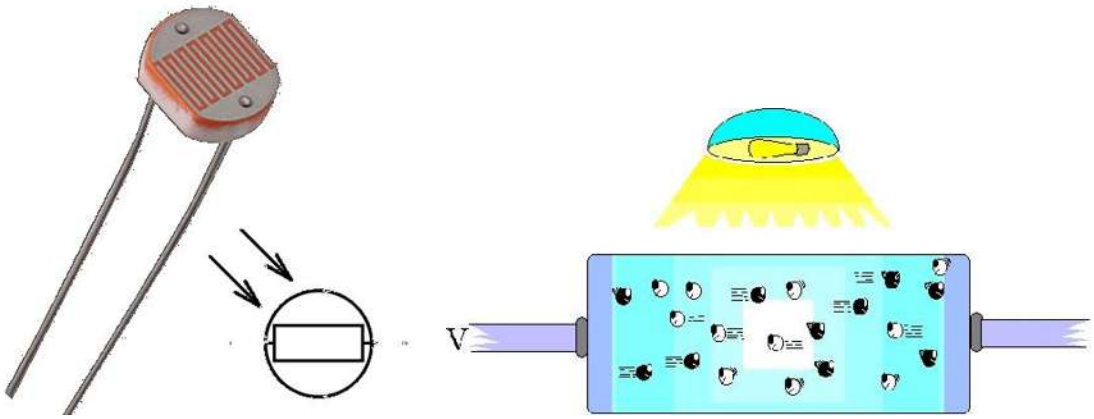


Figura 37 Funcionamiento de una fotorresistencia

Tenemos también a los sensores ópticos, que son aquellos que detectan variaciones de luz, y tienen múltiples aplicaciones en la vida cotidiana, desde la banda automática en la caja de los supermercados hasta en los dispositivos inteligentes.

Su funcionamiento es bastante sencillo, se trata de dos partes, la primera expulsa un haz de luz y la segunda la recibe, si hay alguna alteración en este ir y venir de la luz, el sensor se activará.

Puede hacerse incidir un haz de luz visible o infrarojo, son dispositivos electrónicos que sirven de manera ejemplar de manera binaria con ayuda de adaptaciones digitales.

En la **Figura 38** vemos que sale un halo de luz por el transmisor y al "chocar" con un objeto se refleja y se dirige al receptor, también existen aquellos como en la **Figura 39** en la que se mantienen siempre en esta transmisión y recepción de luz, hasta que un objeto interrumpe esta continuidad, este último se puede encontrar en la banda automática de los cajeros en los supermercados.

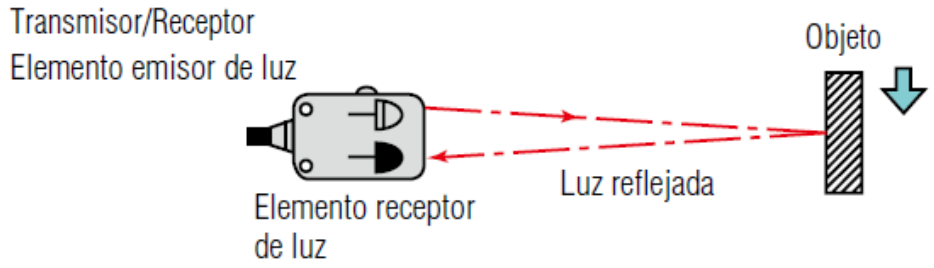


Figura 38Fototrasmisor y receptor unidos

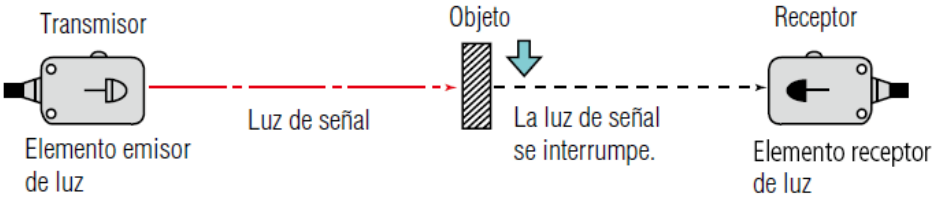


Figura 39Fototransmisor y foto receptor separados

Los sensores también nos sirven para medir temperatura, profundidad e intensidad de alguna magnitud física.

2.8.3 Voxel y Pixel

2.6.3.1 ¿Qué es un pixel?

Viene del acrónimo de dos palabras en inglés "***Picture Element"*** que significa "Elemento de imagen" el cual se reduce a ***Pixel,*** que hace referencia a la unidad más pequeña en una imagen, es decir, que si se aumenta el tamaño de una imagen es posible notar estas pequeñas unidades con forma de pequeños cuadrados como se ve en la **Figura 40**

Figura 40Ejemplo de como los pixeles componen una imagen

Con lo anterior es posible decir que una imagen, entre más pixeles tenga es mayor su resolución y su tamaño para una misma imagen.

En una imagen de color verdadero (*true color*) se usan tres bytes para definir un color, equivalentes a un valor total de 225 colores, cuyo resultado es 16.777.216 opciones de colores diferentes. La diferencia entre esta profundidad de color de 24 bits y una de 32, son 8 bits más para un canal de transparencia o alfa. [11]

En las imágenes digitales, la selección se basa en tres colores principales, Rojo, Verde y Azul y van de 0-255 según la intensidad del color, se expresa de la siguiente manera (255, 255, 255) con la opción de variar estos dígitos según la combinación que se espera, en algunos casos se agrega un extra como se mencionaba antes, el cual es la transparencia, la cual se expresa igual que con los colores (255, 255, 255, 255) y que van en orden de Rojo a Transparencia.

2.6.3.2 ¿Tipos de Pixel?

En el gran mundo de las imágenes digitales existen otros términos utilizados para definir determinados elementos de una imagen entre los cuales se encuentran los **megapixeles**, el **texel** y el **voxel.**

Megapixel: Hablando de una imagen digital, Megapixel se abrevia como Mpx que es el equivalente a un millón de pixeles, seguramente, se ha escuchado este término a lo largo de los años, esto se debe a que en las cámaras fotográficas digitales es una forma de saber el tamaño de la imagen (cantidad de pixeles) y por ende calidad de la misma para poder reproducirla, analizarla y procesarla.

Texel: Este término surge, al igual que pixel, de la unión de dos palabras en inglés, "**Texture Element"** o elemento de textura en español. Se trata de la unidad más pequeña usada en la textura de una imagen.

Como ya se ha dicho, el caso de una imagen digital se encuentra representada por un conjunto de pixeles, mientras que en el caso de una textura, ésta se halla representada a través un grupo de texels. Es por ello, que este término suele utilizarse casi exclusivamente para los gráficos de computadoras, y así, como un texel puede estar conformado por varios pixeles, también puede ser menor a un pixel por lo que resulta imperceptible para el ojo humano.

De manera general, los texel, se agregan a los pixeles para conformar la imagen final en especial en las superficies 3D de imágenes digitales.

Voxel: Viene de "**Volumetric Pixel"** o Pixel volumétrico, hace referencia a la unidad cubica más simple y homogénea que forma parte de un objeto tridimensional. Es el equivalente al Pixel de 2D pero en 3D, se usa para imágenes generadas por computadora, en especial para cierto tipo de gráficos y estudios médicos.

Pixel Muerto: Aquellos pixeles en una imagen que son defectuosos, existen dos tipos, los pixeles muertos cálidos y los sólidos, los primeros se presentan como un pixel blanco en la imagen y los atascados, son aquellos que aparecen en la imagen con un color fijo como Rojo, verde o azul. Ambos defectos son visibles de manera notoria y se deben a problemas con la obtención de la imagen. [12]

2.8.4 RGB

Desde tiempos de Aristóteles (384 - 322AC) se decía que los colores son mezclas, este característico personaje decía que existían cuatro colores, los cuales relacionaba con los cuatro elementos, tierra, fuego, agua y aire.

Tiempo después, **Leonardo Da Vinci (1452 - 1519)** realizó una escala de colores básicos: primero el blanco que permitía recibir a todos los demás colores, luego el amarillo para la tierra, verde para el agua, azul para el cielo, rojo para el fuego y por último el negro, que es el color que nos priva del resto.

No fue hasta que **Sir Isaac Newton** (**1642 - 1727**) presenta un esquema de color y demuestra que la luz del sol al hacerla incidir por un prisma, se divida en los diversos colores del arcoíris.

Con esto se demuestra de los seis colores, los cuales van de los infrarrojos a los ultravioletas.

En el **2.3** se explica de donde proviene el concepto de color, que es la luz y algunas de sus propiedades, por lo que es momento de hablar de las propiedades del color.

2.8.5Propiedades del color

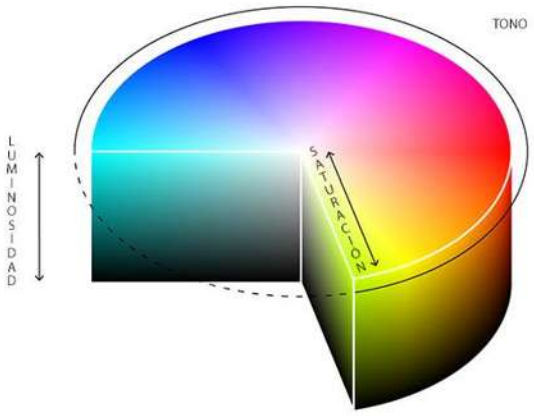


Figura 41cilindro cromatico

El color es un atributo que se percibe de los objetos cuando hay luz. Todo el mundo que rodea es de colores siempre y cuando haya luz.

La luz se propaga por medio de ondas electromagnéticas a la velocidad de 300,000 km por segundo [13]. Debido a las propiedades de la luz que se explican en **2.3.1** los objetos reflejan los rayos que no pueden absorber y el cerebro interpreta que ese objeto es de dicho color que no absorbe, a su vez, otros factores como profundidad o textura se relacionan con los colores que percibimos.

Cada color tiene 3 características que lo definen y lo hacen único, estas características son: el tono, la saturación y el brillo, como se ve en la **Figura 41**.

El tono: También conocido como matiz, es la característica que diferencia un color de otro y con el cual definimos los colores: verde, rojo, azul, etc, y algunos ejemplos se presentan en la **Figura 42**.

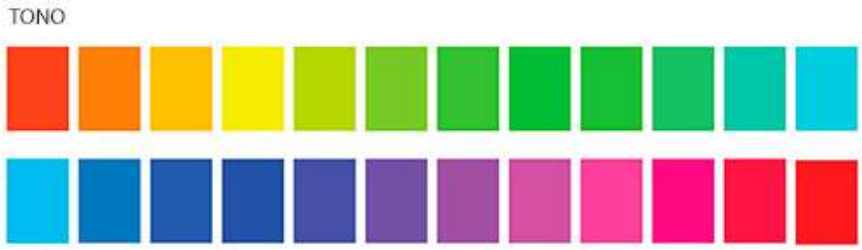


Figura 42Escala de tonos de color

Hace referencia al recorrido de los colores dentro del círculo cromático, es decir, cuando pasa del rojo al amarillo y los colores que existen entre estos dos.

La saturación: Es la intensidad del color. Dicho de otra manera, es la claridad u oscuridad de un color, y se determina por la cantidad de luz que dicho color tiene.

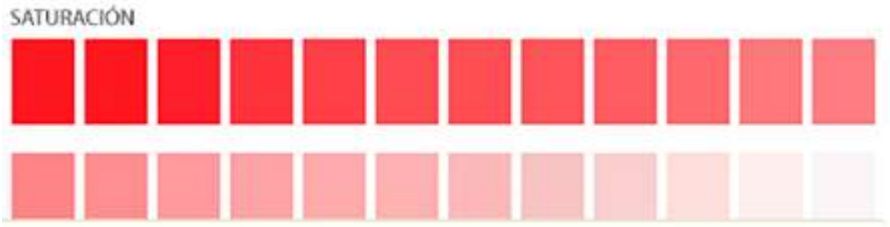


Figura 43Escala de saturación

La saturación cambia de acuerdo a la cantidad de gris que tiene el color, entre más gris tenga tiende a ser más claro o pálido, de manera inversa, entre menos gris el color es más intenso, un ejemplo de esto se muestra en la **Figura 43**.

La luminosidad: Es la cantidad de luz que es reflejada por una superficie en comparación con una superficie blanca en iguales condiciones de iluminación [13].

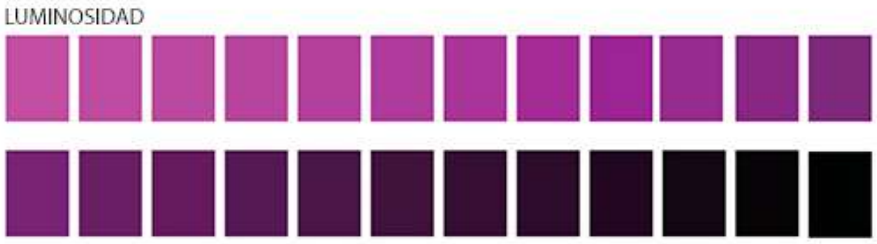


Figura 44Escala de luminosidad

En teoría del color, la luminosidad hace referencia a cuánto de oscuro o de claro es un color. A mayor luminosidad de un color mayor luz reflejará, en la **Figura 44** se aprecia como la escases de luminosidad logra colores de claros a negros.

2.8.6 Imagen (Fotografía)

"*La etimología del vocablo Imagen proviene del latín imago, con el mismo significado. Una imagen es la figura y representación visual o mental de alguna cosa o situación."* [14]

La fotografía es el arte y la ciencia de obtener imágenes visibles de un objeto y fijarlo en una capa material sensible a la luz.

2.8.6.1 Historia de la fotografía.

En 1838, el francés Daguerre lanzó un método práctico: empleó placas de cobre recubiertas con yoduro de plata y expuestas en cámaras de madera y luego sometida a la acción de vapores de mercurio que provocaban la aparición de la imagen latente invisible, formada en el curso de la exposición a la luz.

Este revelado consistía en una gran amplificación del efecto de la luz, con lo cual el tiempo de exposición duraba mucho, alrededor de 30 minutos. El fijado era obtenido por inmersión en agua, saturada de sales marinas.

Con este nuevo invento se motiva a las personas y un año después en 1839 el francés Hippolyte Bayard descubre cómo sacar fotografías en un papel recubierto de cloruro de plata era oscurecido a la luz y luego expuesto en la cámara oscura después de haber sido impregnado en Ioduro de plata, aunque la exposición duraba hasta 2 horas. [15]

En 1841, William Henry Fox Talbot logra duplicar una imagen gracias a la obtención de un negativo intermediario sobre un papel al cloruro de plata, vuelto translúcido gracias a la cera. Como con el invento de Daguerre, la imagen se revelaba luego gracias una solución química compuesta de ácido gálico y de nitrato de plata, a su

vez una segunda hoja de papel que contenía cloruro de plata se exponía a través del negativo previamente obtenido y con eso se tendria la imagen final duplicada.

Utilizando bromuro de plata en lugar del ioduro es como Hippolyte Fizeau logra reducir el tiempo de exposición aún más debido a la sensibilidad a la luz del bromuro, con el cual bastaban de algunos segundos para conseguir el **daguerrotipo,** una fotografía de este dispositivo se encuentra en la **Figura 45.**

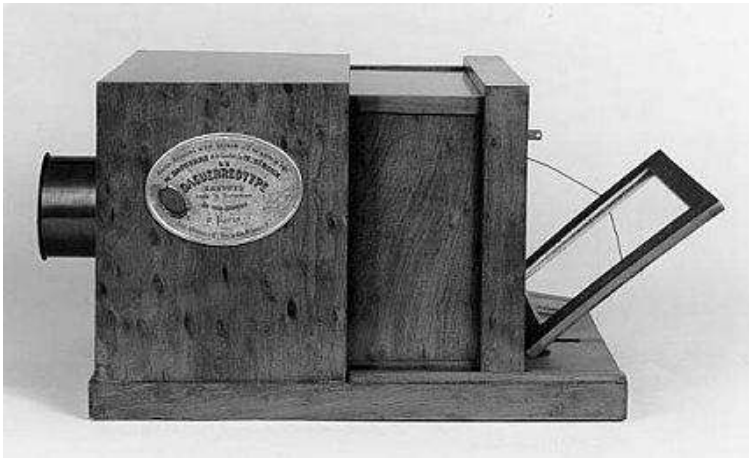


Figura 45 Daguerrotipo, cámara de 1879

Para finales de 1879 y casi rozando 1880, aparece el primer obturador, debido a la sensibilidad de las placas de vidrio utilizadas era necesario que la luz entrara por fracciones de segundo que la mano humana no era capaz de producir, así que por medio de un fotómetro la cámara analizaría que cantidad de luz había y cuánto tiempo se abriría el obturador para dejar pasar solo la luz exacta por el tiempo adecuado. [15]

Casi una década después George Esatman, el fundador de la famosa marca de cámara y fotografías **Kodak,** tiene la idea de usar un medio blando en lugar de esas estorbosas placas de vidrio, así que decide usar rollos de **celuloide.**

Aún faltaba mucho para la fotografía, los colores eran un tema que todos buscaban pero se escapaba de las manos, en 1869 Louis Ducos consigue tomar la primer fotografia a color, usando los principios de Maxwell (Vistos en **2.3.1**) de las ondas de la luz, así que usando filtros de colores rojo, amarillo y azul toma tres fotografías y al superponerlas obtendría la primer fotografía a color.

En 1891 Gabriel Lippman descubre como tomar fotografía a color en una sola placa, pero no fue hasta 1906 que se le otorga el premio Nobel por esta hazaña y ese mismo año los aficionados podrían tomar fotografías a color monoplaca gracias al **autocromo** inventado por los hermanos Lumiere, el cual, consistía del principio usado años atras de la tricomia, pero, en una sola placa gracias a un mosaico de microfiltros de los tres colores realizado con granos de fécula de papa. [15]

A pesar de los grandes avances y la gran nitidez de la fotografía y película actual, no impide que se recurra al bromuro de plata, así como a muchas bases inventadas hace más de 100 años (**AgfaColor** y **Kodachrome**). [15]

2.8.7 Procesamiento digital de imagen

El procesamiento de imágenes tiene como objetivo el alterar el aspecto de las mismas, ya sea mejorándolas, resaltando aspectos importantes que se deseen notar o la degradación de las imágenes originales.

La imagen puede ser generada de formas diversas, por ejemplo, fotográficamente o electrónicamente, por medio de monitores televisivos. Existen dos métodos de procesamiento de imágenes, uno es óptico y el otro, es digital. [16]

El matemático Jean-Baptiste-Joseph Fourier (1768-1830) da por medio de sus análisis una manera sencilla y de cual parte el procesamiento de imágenes, el teorema de Fourier asegura que una gráfica o función, sin discriminar la forma que esta tenga, se puede representar mediante la suma de distribuciones senoidales, de varias frecuencias. Dicho de otra manera, cualquier función, sea o no periódica, se puede representar por la superposición de funciones periódicas de diferentes frecuencias como se ve en la **Figura 46**.

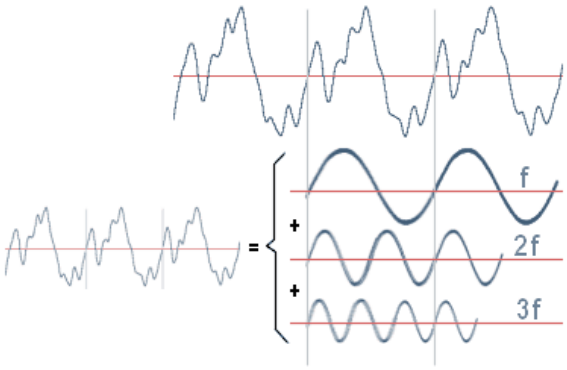


Figura 46Transformada de Fourier aplicada a ondas senoidales

El procesamiento óptico se basa en el hecho de que la imagen de difracción de Fraunhofer de una transparencia colocada en el plano focal frontal de una lente es una distribución luminosa que representa la distribución de las frecuencias de Fourier que componen la imagen, a la que se le llama técnicamente transformada de Fourier.

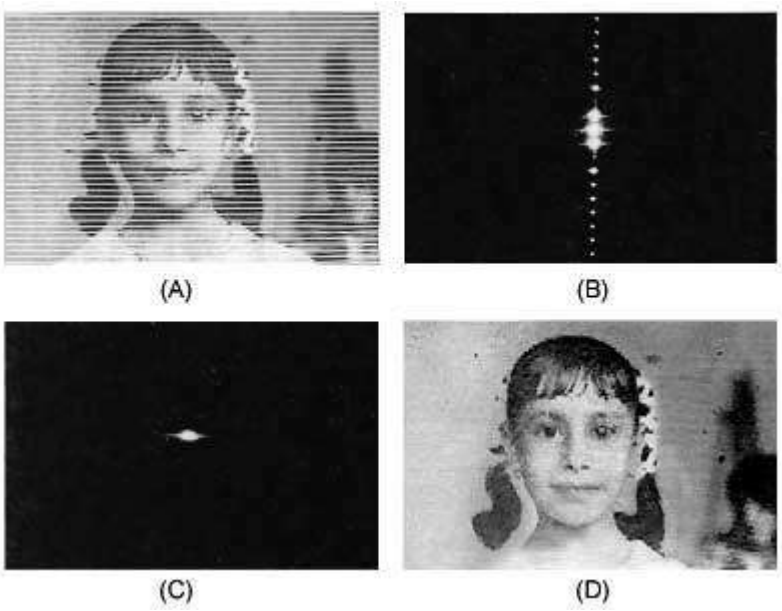


Figura 47Ejemplos prácticos de la transformada de Fourier

En la **Figura 47**. (a) imagen original, con líneas de barrido, tipo imagen de televisión; (b) transformada de Fourier del objeto; (c) transformada de Fourier modificada, después de filtrar e (d) imagen procesada, sin las líneas de barrido.

Procesamiento digital de señales, al igual que el procesamiento óptico, es un campo que tiene años de exploración, sus inicios en la era actual provienen del **Jet Propulsion Laboratory,** quienes trabajan para la NASA y, en 1959 inician con el propósito de mejorar las imágenes enviadas por lo cohetes. Los resultados resultaron tan efectivos, y eficientes que pronto se extendió sus aplicaciones a otras áreas.

Como ya se mencionó en **1.1.3,** el procesamiento se efectúa dividiendo la imagen en un arreglo rectangular de elementos como se muestra en la siguiente imagen.

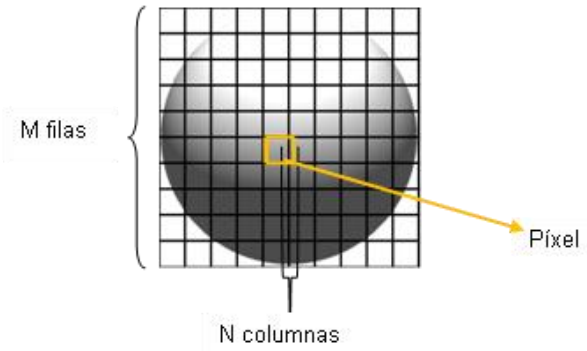


Figura 48Matriz de dos dimensiones representando los pixeles de una imagen

El siguiente paso es asignar un valor numérico a la luminosidad promedio de cada pixel. Así, los valores de la luminosidad de cada pixel, como en la **Figura 48** con sus coordenadas que indican su posición, definen completamente la imagen. Todos estos datos

se almacenan en dispositivos electrónicos, de manera más común en la computadora donde se va a realizar el procesamiento.

Por último, para realizar el procesamiento de imágenes es necesario alterar los valores de cada pixel, estos valores son numéricos y van del 0 que es color blanco al 255 que es saturación del color, un pixel con solo un bit solo puede tener escala de grises como en la **Figura 49.**

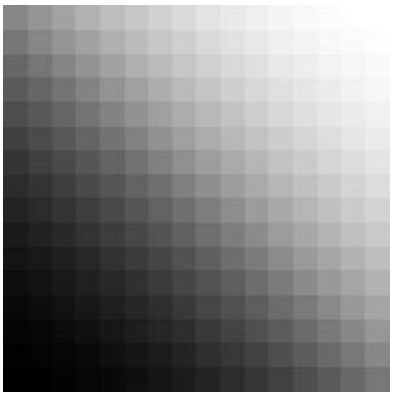


Figura 49Distintos tonos de pixel que van de 0 a 255

Para los pixeles de colores, se utilizan al menos 3 bytes, donde cada bit representa una tonalidad de color, se usan los tres colores primarios rojo, verde y azul, estos pixeles son bien conocidos como pixeles RGB, donde el 0 es el color con toda la intensidad y el 255 es el oscuro, la suma de estos tres puede dar el color blanco como ya se explicó en **2.3.4** y que en la **Figura 50** se puede apreciar.

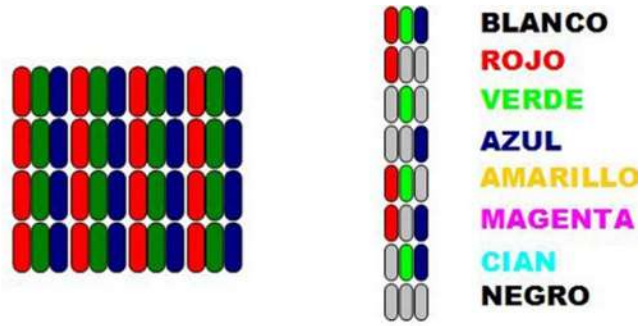


Figura 50Composición de los pixeles RGB

Algunos pixeles digitales tienen un cuarto byte que le otorga transparencia, o bien, profundidad.

Cada uno de estos datos es alterable gracias al procesamiento digital de imágenes, esto no puede ayudar a mejorar la imagen o a resaltar detalles importantes en su representación, por ejemplo en la imagen

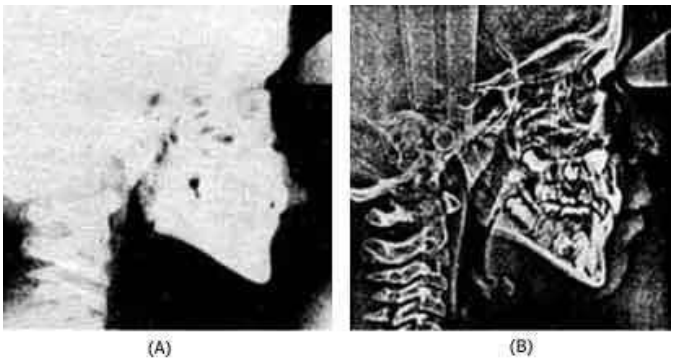


Figura 51Ejemplo de corrección de saturación de iluminación en una imagen

La **Figura 51** (A) es la imagen original donde se aprecia un saturación de blancos y es necesario hacerla más nítida, la **Figura 51** (B) es el resultado luego del filtrado gracias al procesamiento.

2.8.8 Escala de Grises

Es el sistema ordenado y gradual que cubre un rango limitado de valores de luminosidad entre el blanco, el gris y el negro, mucha gente tiende a confundir la escala de grises con el blanco y negro, esto debido, a que son los colores más frecuentes en ambos procesamientos, la diferencia radica en que la escala de grises como su nombre lo indica, empieza en un color negro intenso y va disminuyendo pasando por distintos tonos de grises(mezcla de blanco y negro) hasta llegar a un blanco puro, las imágenes a blanco y negro como su nombre lo indica, es solamente colores binarios, secciones negras y secciones blancas, no hay nada intermedio

Como se explicó en **1.1.3.,** por cada byte pixel que se tenga, se puede tener distintos tonos, donde el valor más alto, el 255 es el color blanco y el 0 es el color negro, donde los intermedios son distintos tonos de grises, en la **Figura 52** se aprecia un ejemplo de estos valores. [16]

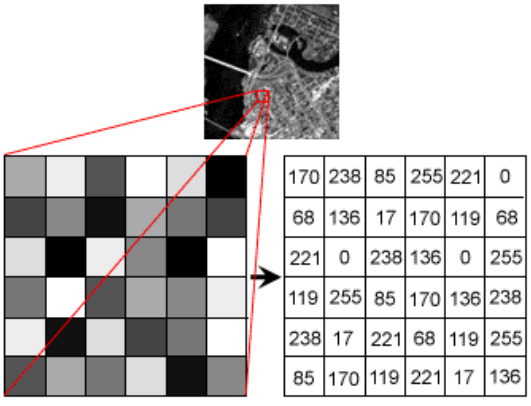


Figura 52 Sector de ina imagen representado en pixeles y esos a su vez, representados con su valor numérico en base a su tono

Dependiendo de la cantidad de bytes de las que se compone cada pixel, se puede tener más tonos de grises, esto se puede determinar con la siguiente **Ecuación 2**:

$$(2^n - 1) \quad (1)$$

Ecuación 2 Tonalidad de grises por pixel

Donde ***n*** es el número de bits por pixel y se le resta una unidad porque se toma en cuenta el valor 0, pero el número total de tonos es el 2^n esto se demuestra en la **Figura 53**. [17]

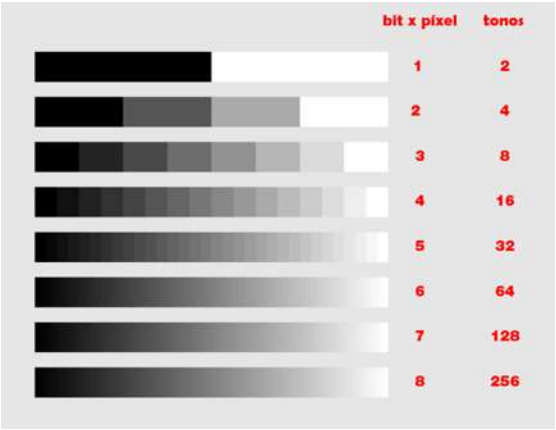


Figura 53 Cantidad de tonos posibles basada en la cantidad de pixeles

Como se muestra en **2.3.4** , se tienen pixeles de distintos colores, esto indica que si una imagen lo representa como un conjunto de vectores, al ser de color se requiere al menos 3 vectores superpuestos para poder alterar los valores, por lo que se opta por transferir la imagen a escala de grises, este es un procesado de imagen en el cual se toman los valores de los pixeles en cada vector y se promedian para que de su escala de grises y proceder a alterar la imagen de una manera más sencilla, posteriormente se puede regresar los colores.

2.8.9Histogramas

El histograma de una imagen es una gráfica que representa en número de pixeles por cada nivel de grises que aparece en la imagen a analizar como se muestra en la **Figura 54**.

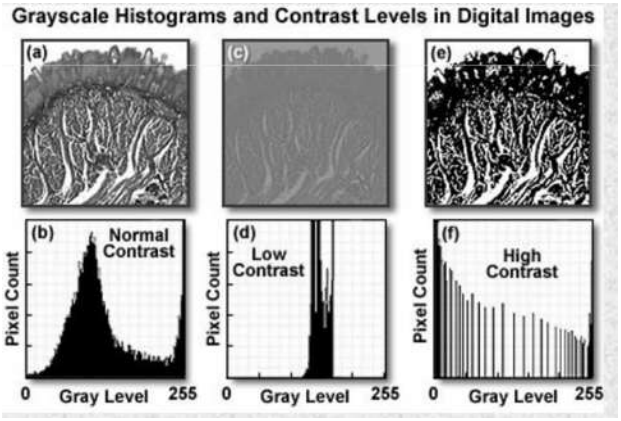


Figura 54Histograma de una imagen en escala de grises

Para usar un histograma, como se ha dicho antes, es necesario tener los tonos de grises, entonces, el histograma representa de manera horizontal los 256 valores de tonos y de manera vertical la cantidad de pixeles por cada tono.

Aparte de analizar, el histograma proporciona información sobre detalles de la foto como lo son resolución, nitidez y saturación, al mismo tiempo, permite saber que valores modificar para lograr alterar la imagen de una manera consciente para mejorar o empeorar según sea lo que se busca. El método más usado es la ecualización del histograma que busca repartir de manera "uniforme" los valores para tener resultados proporcionados.

2.8.10 Filtros

Un filtro es un elemento que discrimina o admite ciertos elementos que pueden o no ser requeridos para procesos específicos, en el caso particular del procesamiento de imágenes digitales, un filtro retiene elementos informáticos no deseados y deja pasar otros que sí lo son, a su vez proporciona corrección o alteración de la imagen.

Existen una gran cantidad de filtros según sea la finalidad de aplicarlos, para el procesamiento al que se hace alusión en el tema de este trabajo de tesis es necesario usar dos tipos de filtros, pasa bajas y pasa altas.

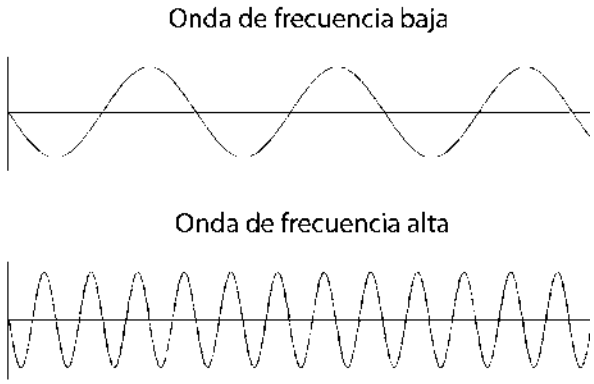


Figura 55Ejemplo de frecuencias, baja y alta respectivamente

En la **Figura 55** se aprecian una onda senoidal, una de frecuencia alta y otra baja, la frecuencia alta presenta una mayor velocidad para que ocurra un evento, es decir la cresta de la onda sucede en menor tiempo a comparación de la frecuencia baja, la cual presenta un tiempo mayor para que ocurra el mismo evento.

Filtro Pasa-bajas.

Este tipo de filtro, como su nombre lo indica, dentro del histograma, deja pasar solamente los valores que se encuentran más oscuros o de frecuencias más robustas, lo que hace que la imagen presente una difuminación, al no permitir pasar las altas, los bordes y relieves se ven atenuados, lo que permite atenuar el ruido.

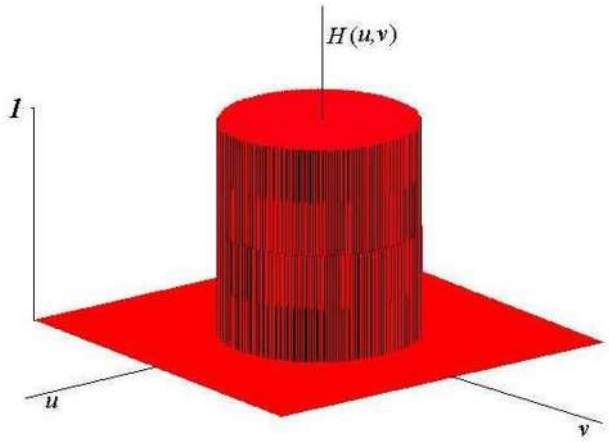


Figura 56Ejemplo de filtro pasa bajas

En la **Figura 56** se hace referencia a como se visualiza un filtro pasa bajas en una gráfica en tres dimensiones, donde el eje ***I*** representa la amplitud de la onda, el eje ***U*** representa el tiempo y el eje ***V*** representa la frecuencia.

El difuminado que otorga este tipo de filtro se le conoce como ***blur,*** también se les conoce como filtros promediadores, debido a que para eliminar ciertas características del ruido, completa pixeles vacíos con el promedio de los pixeles a su alrededor, elimina detalles definición pero proporciona un suavizado y corrección de imperfecciones.

Figura 57Imagen original sin procesamiento

Figura 58Imagen procesada con el filtro Blur

En la imagen se puede ver como se suaviza la imagen, la **Figura 57** es la original y la **Figura 58** es filtrada, se aprecia cómo se pierden detalles, pero, en ella se ve la imagen más constante, con bordes suavizados y sin tantos detalles.

Si en el ejemplo anterior se modificaran los valores del filtrado, a mayor cantidad la imagen empieza a convertirse en una figura sin formas, solo diferencia en los tonos de grises, como si estuviera desenfocada como se muestra en la imagen.

Figura 59Imagen con filtro Blur de menor orden

Por el contrario, si la imagen tuviera un filtro menor, se vería muy parecida a la original, con más nitidez como en la **Figura 59**.

Filtro pasa-altas.

De manera contraria, el filtro pasa-altas como su nombre lo indica, deja pasar aquellos elementos de altas frecuencias, lo que hace que se intensifiquen más los contornos de las imágenes, filtrando los valores bajos de frecuencia en las imágenes.

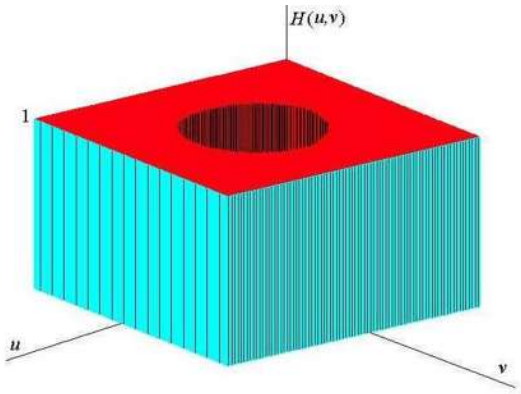


Figura 60Filtro pasa altas en tres dimensiones

En la **Figura 60** se hace referencia a como se visualiza un filtro pasa altas en una gráfica en tres dimensiones, donde el eje ***I*** representa la amplitud de la onda, el eje ***U*** representa el tiempo y el eje ***V*** representa la frecuencia.

Este tipo de filtrado es usado para detectar relieves, bordes y discontinuidades, acentúa estos rasgos y elimina el resto, es también conocido como filtro Prewitt, se puede decidir qué clase de bordes interesa, ya sean todos, o solo los que tengan cierta forma o cierta dirección, filtros horizontales, verticales y completos.

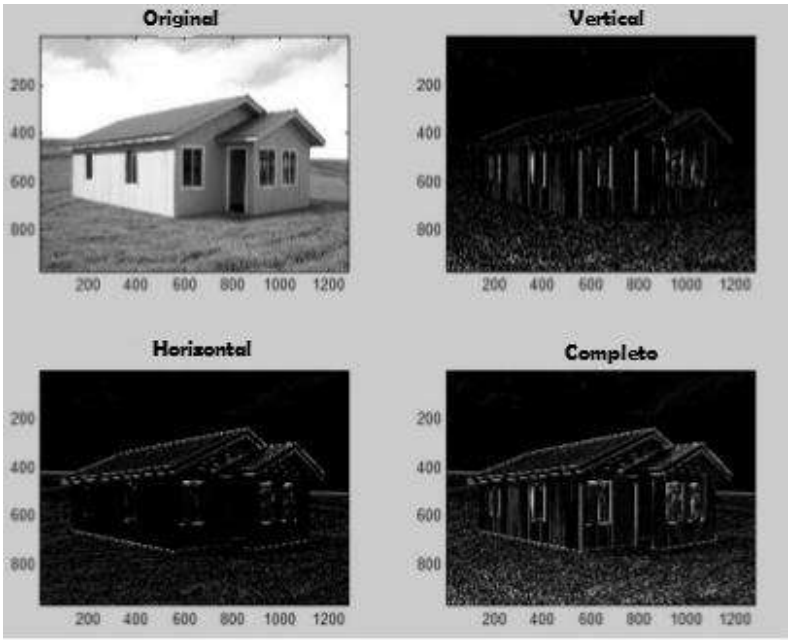


Figura 61 Filtro pasa altas aplicado a una imagen, aplicado en distintos vectores

En la **Figura 61** se aprecian algunos tipos de filtrado pasa-altas, donde se nota que según su orientación es la calidad de la imagen, en la última imagen, donde está completo el filtro que detecta los bordes más sobresalientes, los que están difuminados como las nubes no los toma en cuenta.

Este filtro funciona matemáticamente de manera sencilla, en cada vector de la imagen cuando detecta un cambio brusco en la numeración indica un borde y lo marca.

2.8.11 Detección de bordes.

Un borde caracteriza los límites de un objeto cualquiera, una discontinuidad intensa sobre los puntos de una función, matemáticamente un borde representa un cambio repentino o brusco en la matriz de representación de imagen.

En imágenes digitales los bordes son causados por [18]:

- Cambio brusco en la distancia entre el objeto y la lente de la cámara.
- Discontinuidad en la superficie de los objetos.
- Cambios de iluminación y sombras.

En la **Figura 61** se puede notar un cubo y dentro de él se han señalado dos secciones, una a la derecha y otra en la parte inferior, ambas representan esquinas o bordes, analizando los valores de la matriz de colores, se aprecia con claridad el cambio brusco en los valores.

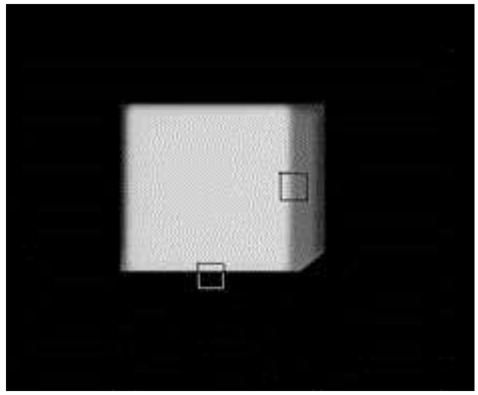


Figura 62Bordes en escala de grises para representación matricial

En la **Figura 63** se puede apreciar la matriz de referencia en la escala de grises del recuadro marcado al costado izquierdo, vemos que luego de la columna 5 de izquierda a derecha, se muestran los valores menores, ahí está el borde.

Sección del costado derecho

211	212	213	208	203	189	167	149	145	137	122	109	107	108	105	100
212	210	207	205	199	185	168	154	145	133	120	112	111	110	108	102
213	209	208	206	200	187	171	155	145	137	126	113	113	114	106	103
214	210	209	207	199	184	167	153	146	137	122	111	110	113	104	101
214	213	211	208	200	181	163	154	147	134	122	116	114	113	105	100
210	210	211	206	198	182	163	149	142	132	121	113	111	110	107	103
214	213	210	204	196	183	167	153	146	137	127	117	113	111	106	102
210	211	208	204	199	185	168	151	142	134	123	114	112	111	105	101
213	210	208	208	200	182	164	150	144	136	123	110	109	111	104	101
211	208	207	206	197	180	162	150	145	137	122	107	110	112	105	102
214	212	212	207	198	182	166	154	149	139	124	113	109	111	106	100
213	212	210	206	199	184	165	152	146	138	123	114	108	112	104	102
212	211	209	205	198	183	167	150	145	136	122	113	109	113	105	103
214	210	208	206	198	182	165	151	144	136	124	115	110	110	104	101
211	212	207	205	199	183	167	153	146	137	122	114	109	111	106	102
213	211	209	204	197	185	167	150	145	135	125	115	110	112	104	102

Figura 63 Representación numérica de la Figura 62 y como de representa un borde numéricamente

Sección inferior

210	212	215	214	213	214	212	211	215	215	214	213	214	213	213	212
208	211	212	210	209	211	211	210	210	210	210	211	209	208	211	210
211	213	213	209	210	214	214	210	208	210	210	209	210	213	214	212
211	213	211	210	210	211	211	209	211	208	209	210	209	209	210	211
211	212	210	210	210	211	210	209	209	211	211	211	210	208	209	210
207	208	208	208	209	207	205	205	207	208	208	207	207	207	208	206
207	206	208	208	207	207	206	206	208	211	209	209	208	207	208	206
205	206	205	205	205	205	205	204	205	207	206	205	206	205	205	204
182	180	181	181	181	181	183	184	183	181	181	183	183	183	182	181
093	093	093	093	093	094	093	093	093	093	093	093	093	092	092	093
004	001	003	006	004	007	005	000	002	001	004	003	005	002	004	003
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000	000

Figura 64De la Figura 62, un segmento del borde inferior representado en una matriz numérica

Eventualmente para detectar estos bordes por medio del procesamiento digital de una imagen, son necesarias algunas recomendaciones.

Para eliminar errores ocasionados por los colores, es necesario filtrar la imagen a escala de grises como se ve en **2.4.1 ,** sabiendo que la imagen es un conjunto de vectores, se promedian los valores de los vectores de color para que nos quede un solo vector en escala de grises.

En la **Figura 65** se presenta una imagen muestra de color, que será filtrada.

Figura 65 Imagen a color sin filtro

Una vez que se convierte a escala de grises, es necesario filtrar la imagen para eliminar ruidos naturales de la foto, o que, se hayan originado al momento de convertir a escala de grises, podemos usar un filtro pasa bajas, puede ser un filtro ***blur*** o un filtro ***gaussiano,*** los cuales atenúan los ruidos y procuran eliminar errores donde se encuentran pixeles vacíos o que no representan un contorno.

Figura 66 Imagen filtrada a escala de grises para eliminar colores

En la **Figura 66** se presenta la imagen de la Figura 65 promediada, logrando un filtro de escala de grises.

En este punto, es necesario usar un filtro pasa altas, este filtro detectará los bordes, eliminando todo lo que no es alta frecuencia en la imagen, por lo que la imagen quedará en negro con los contornos resaltando en blanco, es necesario definir cuál será el margen de tolerancia, en algunas ocasiones estos filtrados ocasionan algo de ruido, lo ideal es que ese ruido no interfiera con la imagen, y que sea tan preciso que no tome en cuenta los pixeles vacíos considerando solo aquellos que tengan una figura definida del contorno.

Figura 67 Imagen filtrada con Canny, filtro pasa altas para detección de bordes

En la **Figura 67** vemos la imagen de la Figura 66 luego de un par de filtros, logrando resaltar los bordes y contornos de la imagen.

Ahora que se tiene el contorno es necesario extraerlo y colocarlo sobre la imagen original, para esto es necesario guardar en una matriz separada los bordes que se han detectado, una vez hecho esto, se regresa a la imagen original y se sobre escriben los bordes de algún color para que resalten en la imagen original, como si fuera una capa de imagen, encimando ambas.

2.8.12 Instalación de OpenCV

Previamente en **2.6,** se mencionan las características de la biblioteca OpenCV, que es de código libre, su origen y los motivos por los cuales ha crecido tanto. A continuación se describe el uso de la biblioteca.

Se eligió está biblioteca por ser gratuita, extensa y tener una vasta comunidad de programadores, es útil y fácil de manejar, se puede instalar en distintos ambientes y lenguajes de programación como lo es Phyton, y permite procesar imágenes y video de

diversas formas, desde dibujar en un panel en blanco, hasta, editar videos en tiempo real, esta última es primordial para el desarrollo de este proyecto de tesis.

La instalación es bastante sencilla, basta con descargar la biblioteca de OpenCV para Android Studio de Internet, una vez descargado el archivo es necesario guardar esta carpeta que se nombrará "OpenCV librarie", es necesario ahora, buscar en nuestra computadora, la carpeta en la cual está instalado Android Studio, se abre la carpeta de instalación y se almacena el archivo que se acaba de descargar, se cierra el administrador para posteriormente abrir de manera normal nuestro ambiente de desarrollo de Android, ahí se tiene un asistente de instalación de bibliotecas, es posible localizarlo en la parte superior izquierda como una pestaña, al dar click en ese apartado correspondiente, abrirá una ventana emergente en la cual se debe buscar la carpeta que previamente descargamos de internet (OpenCV librarie) y que se guarda en la carpeta de instalación de Android Studio, aceptamos la selección de la carpeta donde se encuentra en la biblioteca de OpenCV y acto seguido, se iniciará la instalación, una vez que concluya la instalación, es necesario cerrar Android Studio, abrirlo, y dejar que se actualice, realizados los pasos anteriores, se podrá percatar de que el menú ha cambiado y ahora aparecen distintas opciones referentes a OpenCV.

2.9 Vehículos

2.9.1 Tracción por medio de motores.

Un motor, transforma algún tipo de energía en movimiento, existen distintos tipos de motores utilizados en la electrónica y robótica, como lo son los **BLCD**, **motor a pasos** y **servomotores**, solamente enfocaremos en dos tipos de motores eléctricos básicos, de **corriente alterna** y de **corriente directa.**

En orden, el primero de ellos, el motor de **corriente alterna** recibe su nombre dado que su funcionamiento se basa en transformar la energía alterna en movimiento, dicha corriente se da en dos sentidos, por lo que hay fluctuación de la energía. En un osciloscopio se puede observar que un ciclo completo tiene un semiciclo positivo y otro semiciclo negativo, como se ve en la **Figura 68**, a este sentido de cambio en el flujo de los electrones lo conocemos como frecuencia y se mide en **hercios (Hz),** esto permite que no tenga polaridad la conexión, se simboliza con las iniciales **AC,** y se puede encontrar en las líneas de transmisión y en los hogares**.** [21]

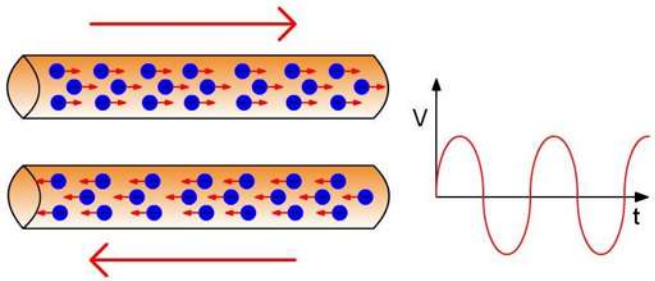


Figura 68Flujo de electrones y grafica de voltaje contra tiempo

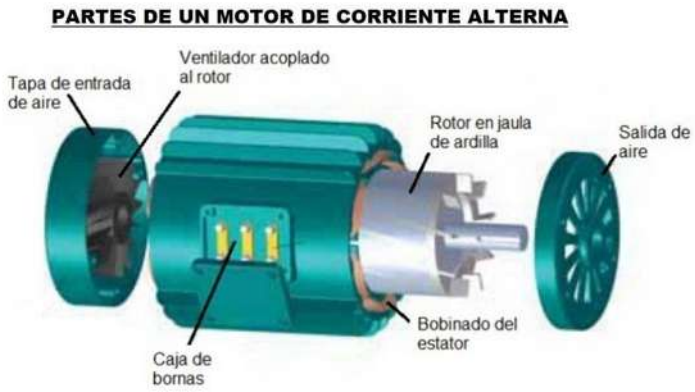


Figura 69 Partes de un motor de corriente alterna

En, el motor de **corriente directa,** de los electrones tiene un lugar de inicio y un lugar de llegada como se ve en la **Figura 70**, tiene polaridad, y como tal, va a tener una conexión específica, normalmente se distinguen por tener dos cables de distinto color entre sí, se simboliza con las iniciales **DC,** y se puede encontrar en las baterías de cualquier tipo y fuentes rectificadas, su ventaja es que no requiere una conexión a una toma de alimentación.

Su representación se encuentra en la **Figura 70.**

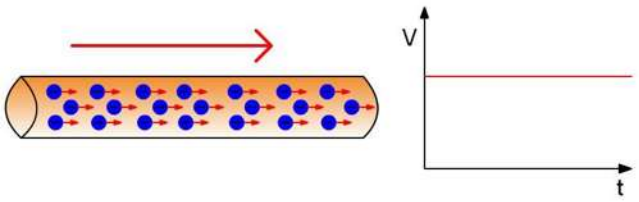


Figura 70Flujo de corriente directa

Para lograr transformar la corriente alterna en directa, es necesario, lograr que los electrones se separen y que en lugar de ir en dos sentidos solo vaya en uno, esto se puede lograr con un **puente rectificador** o un **puente de diodos** como en la **Figura 71** el cual muestra que del lado izquierdo tengamos la entrada de AC y al pasar por la rectificación de los diodos, obtener DC del lado derecho.

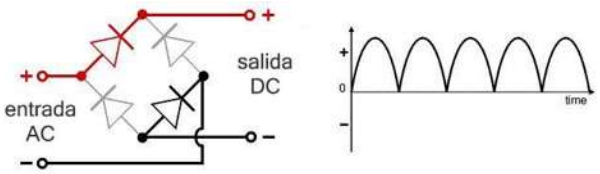


Figura 71Puente de diodos rectificador de onda

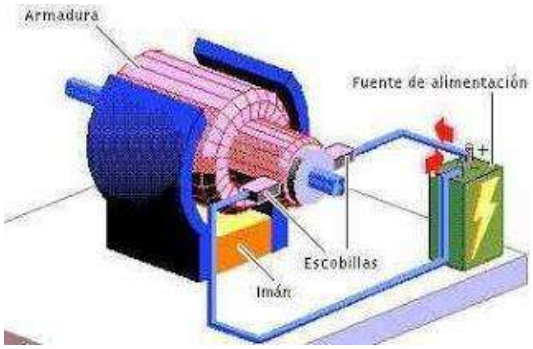


Figura 72 Partes de un motor de corriente directa

Los motores de **corriente alterna** presentan un torque mayor a comparación de los de **corriente directa**, sin embargo, para nuestro proyecto es necesario que tenga libertad de desplazamiento y que no esté conectado a un tomacorriente, para su funcionamiento se requieren baterías y, por lo tanto, **motores de corriente directa.**

Teniendo en el mercado una gran cantidad de motores de distintos tipos, tamaños y costos, se decidió acotar la búsqueda a aquellos motores que funcionaran con **corriente directa** y que no requiera un voltaje mayor a 6Volts, de los cuales se encontraron los siguientes modelos:

Faulhaber 1524E06:

Figura 73Motor Faulhaber de 6 V

El motor que se muestra en la **Figura 73** inicia su movimiento desde los 3V y hasta los 24V, teniendo una eficiencia máxima del 80% alcanzando una velocidad de hasta 9500RPM sin carga, cuenta con motoreductor y el precio en mercado ronda en los $500 pesos mexicanos.[21]

Nichibo Taiwan RF-500TB

Figura 74 Motor Nichibo RF-500 (imagen obtenida de mercado libre)

El motor de la **Figura 74** tiene voltaje nominal de 6V y un rango entre 1 y 26VDC, 2755RPM a máximo voltaje y una eficiencia neta de 55.16%, no tiene moto reductor y el precio en el mercado oscila alrededor de los $200 pesos mexicanos. [22]

NMB Technologies PPN7PA12C1

Figura 75 Motor NMB de 6V (Imagen obtenida de Amazon)

El motor de la **Figura 75** tiene un voltaje nominal de 5V y un rango entre 1V y 7V, 15000RPM a máximo voltaje y una eficiencia neta de 52.73%, no cuenta con motoreductor y el precio en el mercado oscila alrededor de los $230 pesos mexicanos. [23]

Moto reductor S330012

Figura 76Motoreductor S33 de 6V (Imagen obtenida de Mercado libre)

El motor reductor de la **Figura 76** tiene un voltaje nominal de 6V, 150RPM, máxima eficiencia de 73%, como su nombre lo indica tiene motoreductor, y su precio en el mercado ronda los $470 pesos mexicanos. [24]

Motoreductor amarillo.

Figura 77Motoreductor amaraillo de 6V.

El moto reductor de la **Figura 77** tiene un voltaje nominal de 5V, 200RPM a máxima eficiencia, como su nombre lo indica tiene motoreductor, y su precio en el mercado oscila los $50 pesos mexicanos. [25]

Con base en la investigación previamente mostrada, se llegó a la decisión de utilizar el motoreductor amarillo, el reductor ya incluido y el costo fueron los factores que nos llevaron a dicha conclusión, agregando su fácil adquisición.

2.10 El microcontrolador.

Un microcontrolador (**MCU**) es un circuito integrado programable, capaz de ejecutar algoritmos almacenados en su memoria interna.

En otras palabras, es una computadora de tamaño reducido, con características que permiten tener control de los puertos de entrada y salida del dispositivo, cuenta con memoria RAM, flash y procesador, es ideal para la automatización de procesos y el procesamiento de información.

Existen una gran cantidad de microcontroladores en el mercado, una de las diferencias entre estos consiste en el launchpad, o la tarjeta de desarrollo con los puertos integrados que facilitan la interacción entre el microcontrolador y el medio físico, estas representan en mayor medida, la diferencia entre herramientas de desarrollo

Otro factor importante en la capacidad de cómputo del microcontrolador, permite la administración de todos los puertos de entrada y salida, la frecuencia de reloj que es proporcional a la velocidad de su funcionamiento.

Los elementos básicos de un microcontrolador son:

- Microprocesador.
- Perifericos (Unidades de entrada y salida)
- Memoria.

Un **Microprocesador** incluye al menos tres elementos **ALU, Unidad control** y **Registro**. [27]

ALU que significa Unidad Aritmetica y Lógica,por sus siglas en inglés (Arithmetic Logic Unit), esta unidad está compuesta por circuitos electrónicos digitales del tipo combinatorios como lo son las compuertas, sumadores y multiplicadores, se encarga principalmente de las operaciones, estas operaciones se dividen en tres tipos:

- **Lógicas:** se encarga de las operaciones lógicas básicas, sumas, multiplicación lógica, diferencia y negación, esto lo logra por medio de las compuertas lógicas OR, AND, XOR y NOT. Cada operación lógica solo trabaja con 1 o 0.
- **Aritméticas:** Las operaciones aritméticas son la suma, resta, multiplicación y división. Dependiendo de la cantidad de bits del procesador que van de los 8 a 64bits será el tiempo de respuesta para realizar dichas operaciones.
- **Miscelaneas:** Se encarga de la transferencia de bits.

Unidad de control. Es el conjunto de sistemas digitales secuenciales, los cuales a diferencia de otros, dependen tanto de los valores actuales como el estado actual de las variables, de esta manera permite la correcta distribución lógica de las señales.

Registros. Son las memorias principales de los procesadores, funcionan a la misma velocidad del procesador y están construidos por FLIP-FLOPS que son elementos secuenciales en la electrónica digital.

Los **periféricos** son los circuitos digitales que permiten tener interacción con el mundo físico, aquel que es ajeno para el microcontrolador. Su función es habilitar y deshabilitar las salidas y entradas digitales, en caso de que el microprocesador cuente con convertidor analógico digital, los periféricos funcionan para elementos analógicos tanto entrada como salida.

- **Puertos de entrada/salida**. Los puertos están relacionados al tamaño del procesador, es decir que un puerto de 8 bits es porque el procesador es de 8 bits. Un procesador de 64 bits, tiene la capacidad de tener un puerto de 64 bits, como el nombre lo indica, permiten la entrada y salida de pulsos eléctricos.
- **Puertos Seriales.** Permiten transformar la información digital paralela (bytes de información) en señales que se pueden transferir por una línea de comunicación serie.
- **Periféricos analógicos.** Son aquellos que permiten la conversión entre señales analógicas a digitales o viceversa, esto sirve para una interacción con el mundo físico que es regido por elementos analógicos.

Por último, la **memoria,** está dividida en tres. La memoria para el programa (FLASH), la memoria para los datos o variables del programa (RAM) y la memoria para configuraciones o no volátil (EEPROM).

- **Memoria Flash.** Es un tipo de memoria que únicamente permite la lectura programable y borrable electrónicamente, lo que se conoce por sus siglas como EEPROM.
- **Memoria RAM.** Memoria de acceso aleatorio (Random Access Memory), su nombre proviene del hecho de que puede grabarse o recuperarse información de ella sin necesidad de un orden secuencial, demás es una forma de memoria temporal, que al apagar o reiniciar el sistema su contenido se pierde.
- **Memoria EEPROM.** son las siglas de Electrically Erasable Programmable Read-Only Memory (ROM) programable y borrable eléctricamente. Aunque puede ser leída un número ilimitado de veces, solo puede ser borrada entre 100,000 y 1 millón de veces. Es el antecesor de la memoria FLASH.

2.10.1 Tipos de microcontroladores.

Con base en estas características mínimas, en el mercado existen una gran cantidad de opciones que cumplen con este propósito, de las más conocidas son las **ARDUINO, PICS** y **Launchpad de Texas instruments**

Arduino es una marca de creación de tarjetas de desarrollo, es de software libre y cuenta con una gran comunidad de programadores que se dedican a mejorar las aplicaciones del mismo, el ambiente de programación es basado en C, pero adaptado a las necesidades

físicas de las tarjetas, cuenta con una gran cantidad de tarjetas que varían en costos y funciones.

Es una placa basada en un microcontrolador ATMEL. Los microcontroladores son circuitos integrados en los que pueden ser programados, por medio del entorno de programación de Arduino IDE. Estas instrucciones permiten crear programas que interactúan con los circuitos de la placa.

La placa a investigar es la de arduino UNO, como la que se muestra en la **Figura 78**

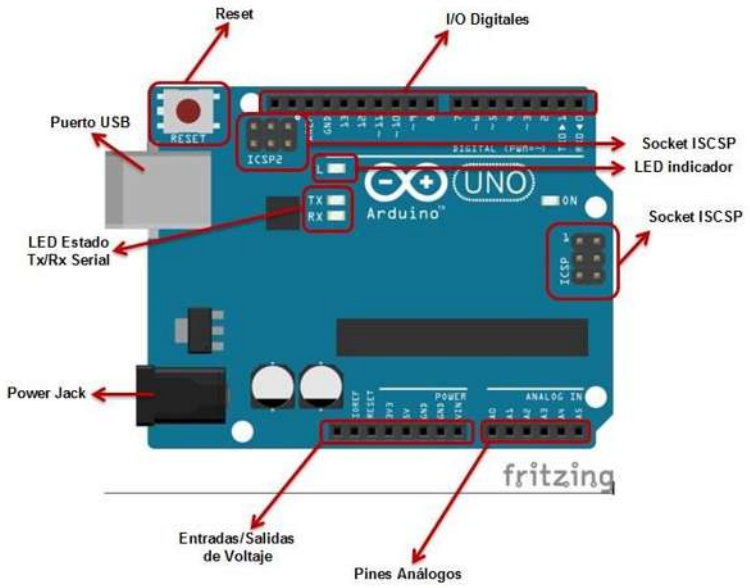


Figura 78Placa de Arduino UNO (Imagen obtenida de Xakata.com))

Algunas características son:

- Microcontrolador ATmega328
- Voltaje operativo de 5V.
- 14 entradas/salidas digitales
- 6 salidas PWM
- 6 pines de entrada análogas
- Memoria Flash: 32KB
- 0.5KB usado por el bootloader
- SRAM: 2KB.
- EEPROM: 1KB
- Velocidad de reloj: 16MHZ

Microcontrolador PIC, son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. El nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico).

Se programa usando lenguaje de ensamblador, que es programación en bajo nivel, nos permite controlar los procesos ejecutados por el chip. No se enfoca en auto eficiencia, sino que corre de manera secuencial.

No cuentan con una tarjeta de desarrollo exclusiva, por lo que Diferentes opciones que dependen de nuestras necesidades, una de las más comunes es la PIC-DIP40 la cual se muestra en la **Figura 79**

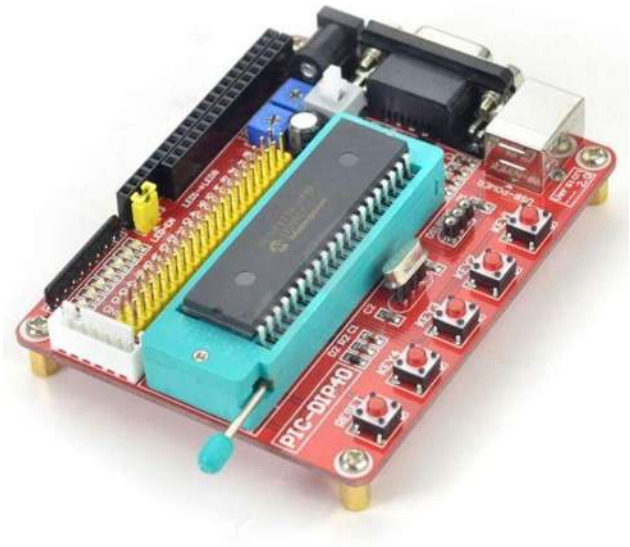


Figura 79PIC-DIP40

Algunas características del PIC16f887 son:

- El microcontrolador cuenta con 35 instrucciones distintas.
- 0-20MHZ de frecuencia de operación.
- Oscilador interno.
- Alimentación de 2 a 5.5V
- 35 pines de entrada/salida.
- ROM: 8KB con flash.
- EEPROM: 256bytes.
- RAM: 368 bytes.
- 14 pines de convertidos analógico digital.
- 1 módulo PWM.

De **Texas instruments** se investigó el **MSP430F5529,** un launchpad.

Un Launchpad es una plataforma de desarrollo para rápidas conexiones y resultados simples, es programable de manera sencilla y generalmente son de código libre, en el caso de Texas instruments se provee un ambiente de desarrollo sencillo, conocido como "**CodeComposerStudio",** el cual provee distintos lenguajes de programación como lo

pueden ser lenguaje C o lenguaje de ensamblador, proporciona a su vez la capacidad de una terminal para usar el puerto serie del microcontrolador.

Figura 80 Ambiente de desarrollo de Texas Instruments para los MSP

El **MSP430F5529** es un modelo específico dentro de la familia de los MSP430, la decisión tan específica se realiza por la facilidad de adquisición, esto no es limitante para el uso de otro microcontrolador para futuros proyectos.

Figura 81 Imagen del módulo MSP430F5529 a utilizar.

Las características principales del dispositivo son:

- 25MHz de CPU.
- 128kB memoria Flash.
- 8kB memoria RAM.
- Conversor Analógico Digital de 12bits.
- Timers, Comparadores
- Multiplicadores de 12bits.
- USB 2.0 de alta velocidad.
- 40 pines de acceso.

- Terminales de aislamiento con el MCU

Como se aprecia en la **Figura 81** se tiene una gran cantidad de puertos disponibles para su uso, cada uno identificado con un número, algunos puertos, cuentan con acciones especiales en el microcontrolador, como lo son el PWM o los puertos RX y TX de la comunicación serial, a su vez se tiene un par de botones disponibles y un par de leds de los cuales se puede hacer uso dentro de los proyectos.

2.11 Plataforma de un vehículo

La plataforma hace referencia al chasis del vehículo, el chasis, a diferencia de la carrocería es el soporte interno, el chasis es al vehículo lo que el esqueleto es al cuerpo humano. Como ejemplo se presenta un chasis de un vehículo en la **Figura 82**.

Figura 82 Chasis de un vehículo automotor. (Imagen obtenida de Wikipedia.com)

Representa la rigidez y soporte para el desplazamiento y el montaje de los elementos de transporte.

Figura 83Chasis vehículo para robótica (Imagen obtenida de markers.mx)

El material de que se realiza el chasis es sumamente importante, debe tener rigidez suficiente para transportar todo lo que lleva el vehículo y suficientemente ligero para no presentar una peso importante que reduzca su velocidad, para los carros eléctricos presentados en un kit de robótica básica, es necesario que sean de materiales económicos y sencillos de encontrar.

En la **Figura 83** se aprecia el chasis construido de plástico, ligero, fácil de cortar, no muy económico y tampoco muy sencillo de encontrar.

Figura 84Chasis de MDF(Imagen obtenida de Demoss.com)

En la **Figura 84** se muestra la construcción de un chasis para vehículo de electrónica básica realizado con MDF, un material sencillo, resistente, ligero, cortado con láser, al tener este método de corte, se puede realizar cualquier corte que se desee de manera sencilla.

Figura 85Chasis de acrílico (Imagen obtenida de googleimages)

En la **Figura 85** el chasis es construido de acrílico, es costoso, difícil de encontrar, no se puede cortar con láser, la ventaja que tiene sobre otros tipos, ya se venden de manera completa en kits de electrónica, como el ZK2 que se muestra en la **Figura 86** el cual incluye motores, portapilas, encoders, llantas y tornillería a precios bajos.

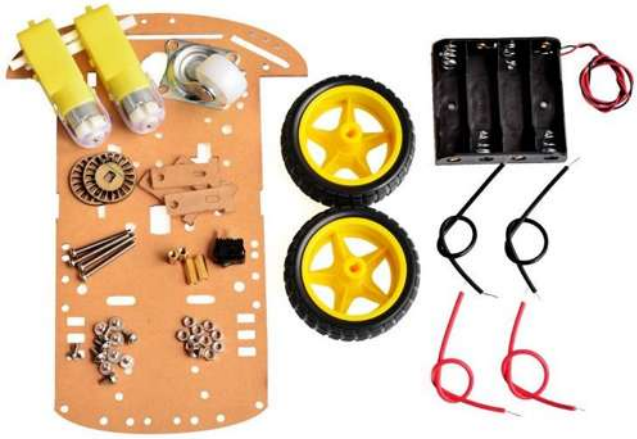


Figura 86 Chasis de acrílico ZK2 (Imagen obtenida de google images)

2.12 Comunicaciones inalámbricas

Las redes de comunicación evolucionan a la par de la industria y sus necesidades, los costos elevados y las dificultades naturales de las redes físicas imposibilitan la distribución correcta, a la par de la velocidad física que se puede conseguir.

En la actualidad, se han diseñado maneras de combatir las distancias y el costo físico que representa el cableado, las comunicaciones inalámbricas son aquellas en las que el transmisor no está conectado físicamente con el receptor.

Las redes inalámbricas presentan grandes ventajas como lo son:

- Grandes distancias
- Mayor velocidad de transmisión
- Instalación más económica que la cableada.
- Permite la conexión de gran cantidad de dispositivos móviles.
- Posibilidad de conectar nodos a grandes distancias sin cableado.
- Permite ampliar una red cableada en caso de redes mixtas.

A su vez, presenta una cantidad de desventajas, razones por las cuales no ha reemplazado a la comunicación cableada.

- Si hay objetos entre el emisor y el receptor, la comunicación puede fallar.
- Son más inseguras que las redes cableadas.
- No hay estudios sobre los posibles daños de las radiaciones constantes.
- Menor ancho de banda para la transmisión y recepción.
- Puede ser interrumpida por radiaciones electromagnéticas.

Lo que cambia en cada tipo de comunicación inalámbrica es el protocolo de comunicación usado, alguno de ellos son:

- Infrarrojo
- Radio comunicación
- Bluetooth
- WIFI
- Telefonía móvil

El **infrarrojo** ha ido en disminución a través del tiempo, representó una de las primeras opciones inalámbricas para la comunicación, utilizando haces de luz para su transmisión, lo cual requería que los dispositivos estuvieran a una distancia corta y colocados en cierta posición, de lo contrario la conexión era difícil o imposible de realizar.

2.12.1 Radio comunicación

Tecnología por la cual hoy se tienen tantos protocolos de comunicación, es la más antigua tecnología inalámbrica, teniendo dato de que se trabaja en esta tecnología desde 1886. [6]

Tiene un gran campo de aplicaciones, se maneja por todo el espectro electromagnético, no excluyendo campos fuera de él como lo son rayos X, infrarrojos y luz UV como se aprecia en la **Figura 87**.

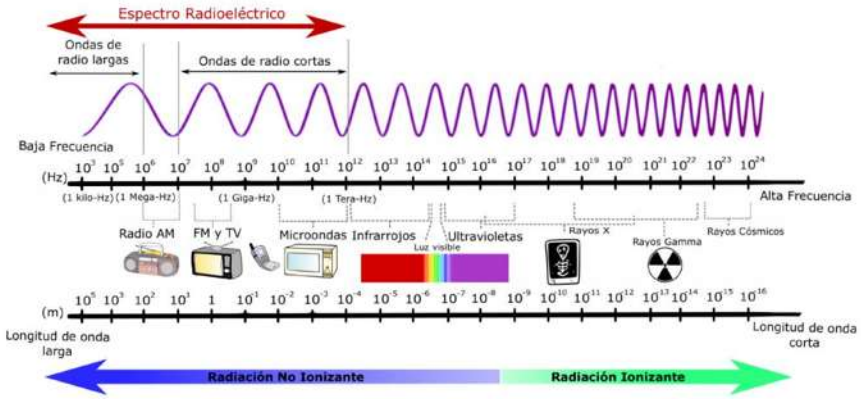


Figura 87Espectro electromagnetico

Una onda de radio se origina cuando una partícula cargada (por ejemplo, un electrón) se excita a una frecuencia situada en la zona de radiofrecuencia (RF) del espectro electromagnético.

Cuando la onda de radio actúa sobre un conductor eléctrico (la antena), induce en él un movimiento de la carga eléctrica (corriente eléctrica) que puede ser transformado en señales de audio u otro tipo de señales portadoras de información. El emisor tiene como función producir una onda portadora, cuyas características son modificadas en función de las señales (audio o vídeo) a transmitir. Propaga la onda portadora así modulada. El receptor capta la onda y la «demodula» para hacer llegar al espectador auditor tan solo la señal transmitida.

Existen tipos de modulación de onda, las más comunes son la AM y la FM, se recomienda que no existan señales a 100Khz de separación entre ellas para evitar la interferencia. La modulación AM significa **Amplitud Modulada** mientras que la FM representa **Frecuencia Modulada.** En la actualidad, la amplitud modulada se encuentra en decadencia debido a lo débil de su señal que ronda el orden de los KHz, mientras que la FM ronda los MHz, la amplitud modulada ocupa una onda portadora en la cual se modifique la original, se hace mayor su amplitud para que pueda llegar al receptor, un ejemplo como en la **Figura 88**.

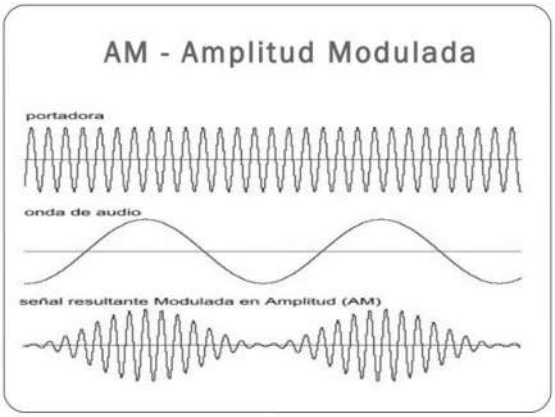


Figura 88 Amplitud modulada

Por otra parte, la frecuencia modulada modifica la señal original agregándole velocidad entre sus crestas, facilita la emisión, y por la velocidad, realiza que tenga una mayor potencia en el alcance final, un ejemplo de esta modulación se puede apreciar en la **Figura 89**.

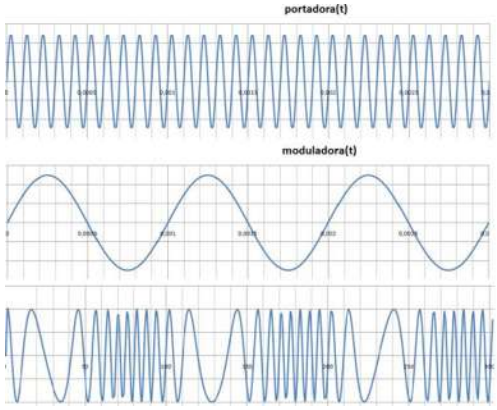


Figura 89 Frecuencia Modulada

2.12.2 Bluetooth

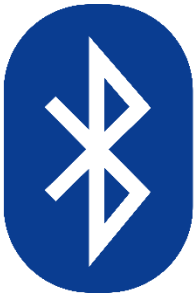


Figura 90Logotipo del protocolo Bluetooth

La comunicación Bluetooth surge de la necesidad de crear redes inalámbricas de mayor alcance y velocidad de las que se tenían en ese momento, alrededor de 1994, no es

sino, hasta principios del año 2000 que empieza a tomar fuerza en la industria de telefonía móvil. [19]

Llamado de esa manera por el rey danés y noruego, Harald Blatand, cuya traducción al idioma inglés, suena como bluetooth, de esa manera se quedó el nombre que conocemos, funciona con radiofrecuencia en el orden de los 2.4GHz, por lo que no interfiere con otros métodos de comunicación pero si puede afectarse por radiaciones superiores, la velocidad de transferencia de datos ha ido evolucionando junto con su capacidad, actualmente su velocidad teórica es de 32Mbits/S. [19]

El problema que tiene esta solución a las comunicaciones es la limitada distancia de transferencia, a pesar de ser muy rápida, tiene un alcance máximo de 10 metros a la redonda en terreno libre sin obstrucciones de por medio, sumado a esto, su otra limitante es el consumo de energía que requiere para su correcto funcionamiento, que oscila entre los 30mW y 1W [19] de potencia, haciendo que su uso desmedido acabe con fuentes de poder que usen baterías para su funcionamiento, es necesario que, cuando no esté en uso, se mantengan en modo de bajo consumo.

2.12.3 WIFI

Figura 91 Logo de comunicación WIFI

Wifi o Wi-Fi es originalmente una abreviación de la marca comercial ***Wireless Fidelity***, que en inglés significa “fidelidad sin cables”.

Una de las grandes ventajas del WiFi es la multiple conexión, a diferencia del Bluetooth o del infrarrojo, es posible que varios usuarios estén conectados a la misma red, para esto el ancho de banda se ve reducido a partes similares entre los dispositivos. Para su funcionamiento es necesario el uso de un ***modem,*** que significa “modulador y demodulador” esto es basado en el principio de comunicaciones, para que la información sea enviada a altas velocidades es necesario modular las ondas, montando unas en otras, para leer la información enviada es necesario demodular, es decir, traducir el mensaje enviado y eliminar la onda transportadora.

En la actualidad Wi-Fi utiliza los estándares 802.11a, 802.11b y 802.11g, siendo éste último compatible con el 802.11b; pero ahora, según las nuevas investigaciones, se puede ver en una próxima oportunidad la implementación del estándar 802.11n.

Cada estándar de tecnología representan una velocidad de transferencia, con las tecnologías 802.11a y 11g, que se utilizan hoy en día, las velocidades son de entre 20 y 24 Mbps, mientras que se espera que con la 802.11n se tengan velocidades de 300Mbps. [28]

Los tipos de wifi y algunas de sus características se presentan en la **Tabla 1.**

Tabla 1Características de algunos módulos wifi

Estandar	Frecuencia	Velocidad	Rango
wifi A (802.11a)	5 GHz	54 Mbits/s	10 m
wifi B (802.11b)	2.4 GHz	11 Mbits/s	100 m
wifi G (802.11b)	2.4 GHz	54 Mbits/s	100 m

2.12.4 Telefonía móvil

Figura 92 Telefonia Movil

La telefonía móvil o telefonía celular es un medio de comunicación inalámbrico a través de ondas electromagnéticas. Como cliente de este tipo de redes, se utiliza un dispositivo denominado teléfono móvil o teléfono celular.

Se tienen registros de finales de la década de los 50’s en el cual ya se experimentaba con este tipo de comunicación gracias al soviético Leonid Ivanovich Kupriyanovich, quien ya buscaba la manera de telefonía móvil. [6]

Al igual que otros protocolos de los que se habla en este apartado, requiere de un emisor y un receptor, la conexión no es directa como la de los dispositivos Bluetooth, sino que se conectan a la red por medio de antenas de telefonía celular, esta antena retransmite la señal haciéndola llegar al receptor, actualmente la telefonía de este tipo es digital, lo que indica que se reduce a paquetes de datos enviados de manera binaria, esto ayuda a la velocidad de transmisión, las antenas usadas para la retransmisión tienen un alcance de hasta 200Km cuadrados. [29]

Dentro de la evolución de esta tecnología, el tipo de conexión ha cambiado:

- **Primera Generación.** Utilizaba canales de comunicación analógicos y servía exclusivamente para transmitir mensajes de voz, con muy escasa seguridad en las comunicaciones y mala transferencia en el paso de una celda/antena a otra.
- **Segunda Generación.** Llegó al comienzo de la década de los 90 y fue el primero en utilizar protocolos de comunicación digitales, siendo el más famoso el conocido como GSM (Global System for Mobile communications). Fue el primer sistema en permitir la transmisión de datos, los populares SMS (Short Message Service), además de voz con un cierto nivel de encriptación.
- **Tercera Generación.** Surgió como respuesta a la necesidad de mayores velocidades de transmisión de datos y mayores capacidades, que permitieron ofrecer nuevos servicios a los usuarios: además de voz y datos, posibilitó el acceso de los terminales a Internet.
- **Cuarta Generación.** Comercializada por las operadoras de telefonía móvil desde 2010 ofrece, entre otras mejoras, mayor seguridad y calidad de servicio, junto a velocidades de acceso muy superiores a las anteriores. Este sistema se ha extendido gracias al uso masivo de los smartphones. La tecnología 4G ha permitido la generalización de aplicaciones tales como Whatsapp, videojuegos en el móvil, navegación en Internet, etc.
- **Quinta Generación.** Se espera que esté disponible en el 70% de los dispositivos en 2020. Su característica principal será la mayor velocidad de transferencia de datos e imágenes. [29]

Capítulo 3 Software desarrollado

3.1 Diseño de interfaz

"Una interfaz es dispositivo capaz de transformar las señales generadas por un aparato en señales comprensibles por otro". [19]

Con la definición anterior se puede entender que es el canal de comunicación entre dos sistemas que por sí solas serían incapaces de comunicarse e interpretarse, es como nuestros conversores analógicos-digitales, que convierten de un tipo de señal a otra, existe una gran cantidad de interfaces, la planteada para este proyecto, es una interfaz gráfica con el usuario, es decir, es un canal de comunicación entre el usuario y el programa que se está creando.

Cuando se crea un nuevo proyecto, se tiene una vista del dispositivo en blanco, sin nada más que la barra superior con la hora de nuestro dispositivo, y en ese lienzo, se pueden poner objetos (*widgets*), existen dos maneras de lograrlo, la primera es arrastrando del menú lateral al espacio en blanco como se aprecia en la **Figura 93**.

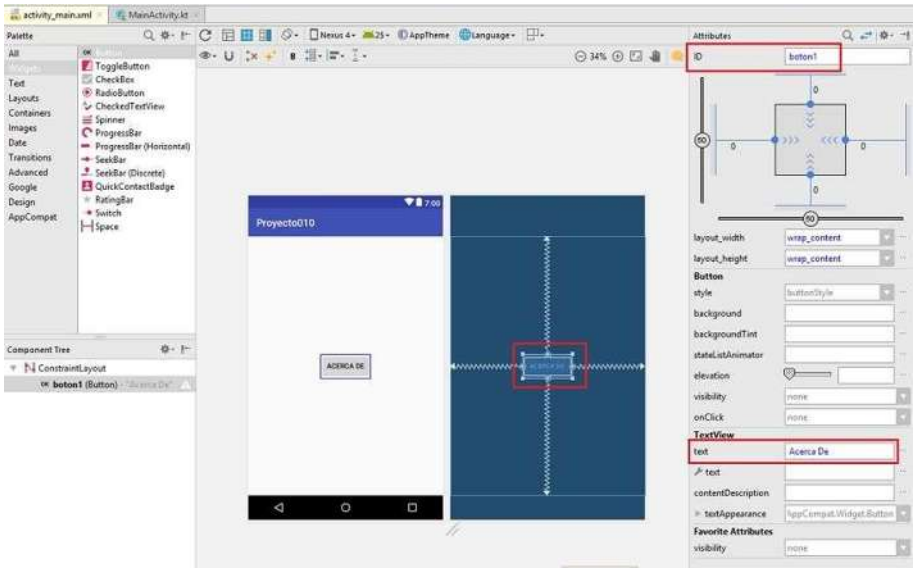


Figura 93Vista del dispositivo en Android Studio

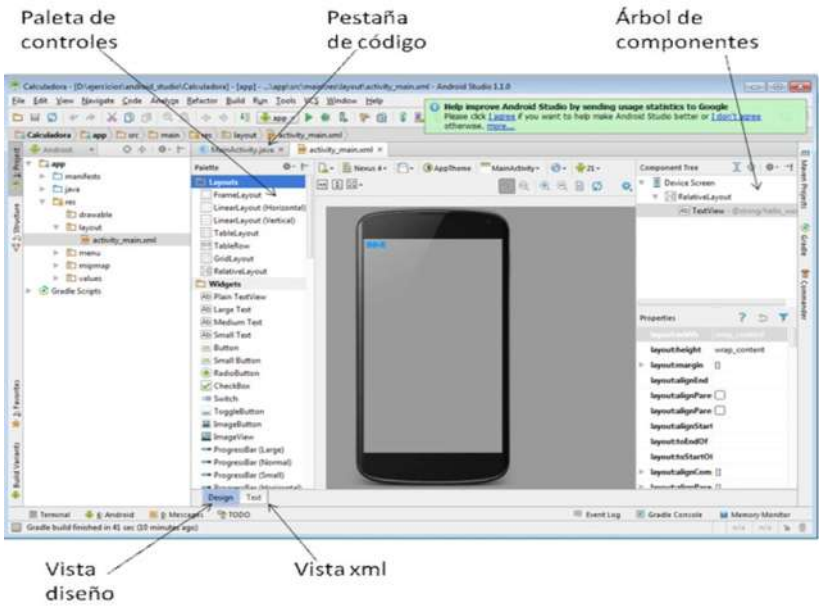


Figura 94Descripción de una ventana en Android Studio

Cada *widget* que se agregue a la vista del dispositivo tiene una repercusión en el la creación de la maqueta en una pestaña de Android Studio llamada XML que es un área de diseño donde indica el *widget* que se usó, en qué orden se colocó, en que capa y en que orientación, es una gran herramienta, arrastras *widgets* para iniciar el diseño, también, podemos programar los *widgets* desde el XML y cuando se vea la vista del dispositivo, se agregan con las características en el código como se ve en la **Figura 95.**

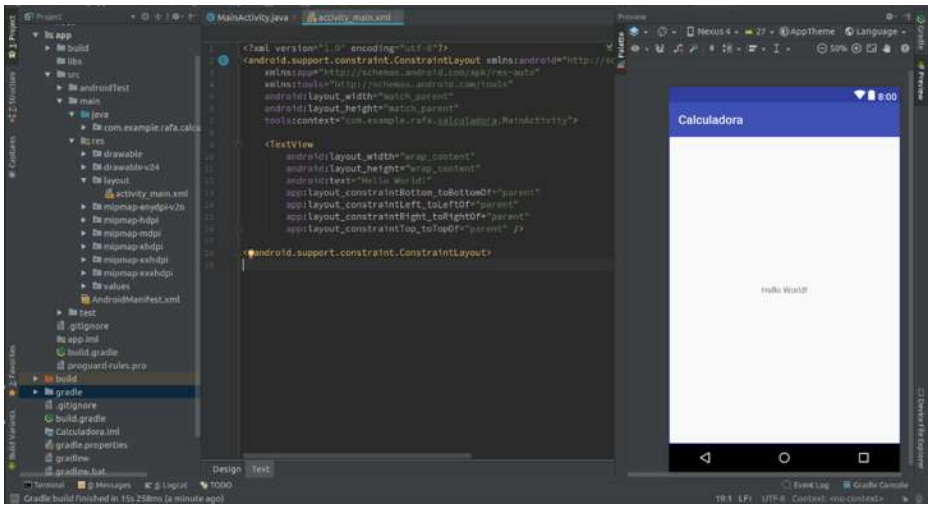


Figura 95Codigo XML de una vista en Android

3.1.1 Botones

Android Studio maneja distintos *widgets* para usar como se muestra en la **Figura 96.**

Figura 96Ejemplo de widges a disponibles

Son las opciones predeterminadas, pero, al ser de código abierto, da la posibilidad de agregar diseños a medida de nuestras necesidades, hay una comunidad muy grande en internet donde usuarios los comparten, o, se puede diseñar los propios.

En este proyecto, se utilizó el *widget* conocido como ***Text Button,*** el cual permite, usar un botón con alguna frase o palabra dentro, de forma rectangular, el tamaño dependerá del espacio que se le asigma en la vista y del tamaño de la palabra que se usa dentro del mismo, en la **Figura 97** se puede apreciar la imagen del programa en desarrollo, se nota el botón en la parte superior con la palabra ***"BLUETOOTH".***

Figura 97Visualización de la cámara dentro de la aplicación creada

Los botones gráficos, deben tener un indicador de que han sido seleccionados/presionados, de manera automática, Android da una función para realizarlo. En la **Figura 98** se puede apreciar los tres estados de un botón, en estado normal sin ser presionado y estando activo, cuenta con un color sólido. Si el botón está presente pero no

es posible de activar, está deshabilitado, por lo que tendrá transparencias y no tendrá colores sólidos. Por último, cuando el botón está seleccionado comienza a llenarse de otro color mientras esté seleccionado.

Figura 98 Ejemplos de Botones en Android Studio

Para indicar distintas acciones, también es posible que se modifique el texto del botón según las condiciones dadas por el programador, en este proyecto en particular, al presionar el botón, se conecta por BLUETOOTH al módulo receptor, por lo que, cambia el texto, indicando una nueva acción como lo es "***DESCONECTAR BT".***

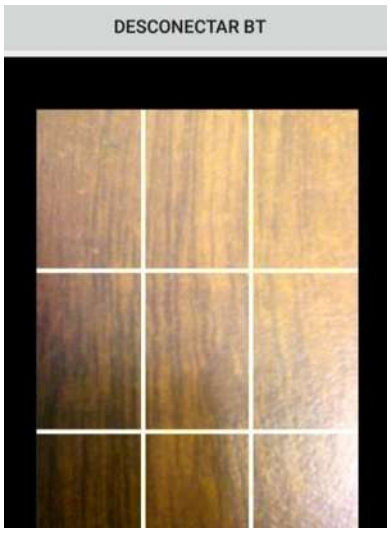


Figura 99 Visualización de la aplicación, luego de pulsar el botón

3.1.2 Fragments

Un *Fragment* representa una sección de la interfaz de usuario dentro de una actividad, estos fragments pueden ser usados en distintas actividades y distintas pantallas, por lo regular son creados para dar opciones, o como, mensajes emergentes que nos indiquen alguna circunstancia dentro de las aplicaciones [27], aunque estos fragments representan una vista gráfica, no son una actividad, ni una nueva pantalla en el dispositivo, y la pantalla que estábamos ejecutando antes de esta intervención, sigue en el fondo. Como mencionamos en el capítulo **3.3**, podemos ver en la mayoría de los casos la pantalla anterior en la parte de atrás del fragment, de un color más oscuro que nos indica que sigue existiendo, pero que, por el momento no es accesible como lo vemos en la **Figura 100** que es un ejemplo obtenido de la aplicación ***GMAIL®*** donde podemos apreciar de lado izquierdo el fragment de la aplicación, nos da opciones de algunas acciones mientras que la pantalla principal, la que vemos de lado derecho, sigue activa en el fondo del fragment de color más oscuro.

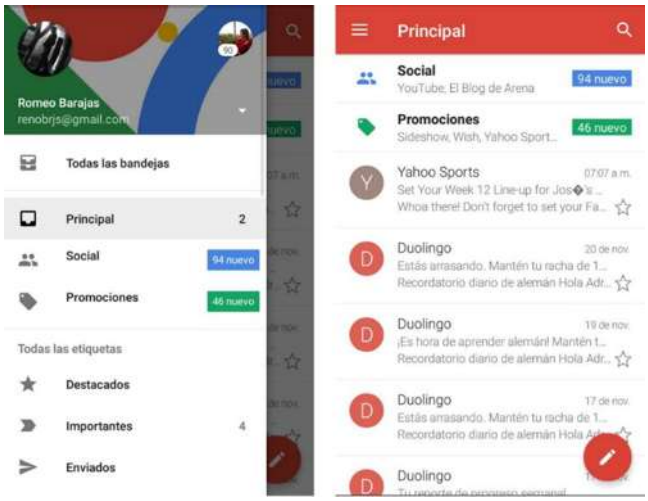


Figura 100Ejemplo de aplicación que usa Fragments

Para nuestro proyecto, usaremos un fragment a manera de pantalla emergente que nos solicita una acción, dicha acción se realiza únicamente cuando no se cumplan ciertas condiciones, en este caso, el fragment se mostrará al abrir la aplicación y no tengamos activada la función bluetooth del dispositivo, de lo contrario la aplicación seguirá con sus funciones programadas, en la **Figura 101** se aprecia el fragment.

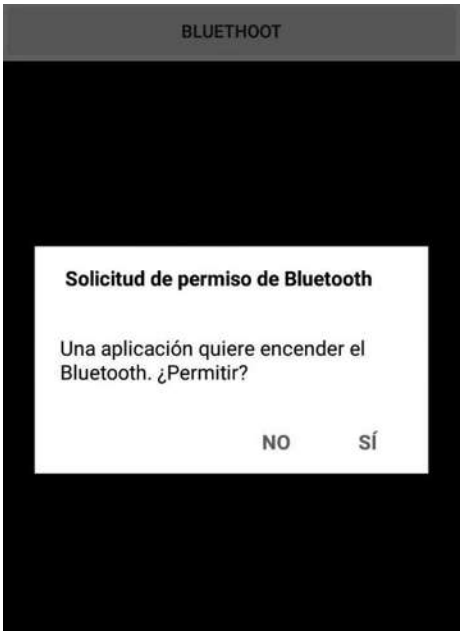


Figura 101 Solicitud de permiso de un recurso en Android

El fragment, indica que la aplicación necesita el modulo Bluetooth para funcionar y permite elegir si se quiere habilitarlo al acceder a la conexión, se abrirá el gestor de recursos de comunicación bluetooth del dispositivo, quedando en segundo plano la aplicación. Una vez que se habilite el protocolo de comunicación, bastará con salir del gestor para reanudar la actividad pasada.

3.1.3 Imágenes

Agregar imágenes a Android Studio requiere un formato de imagen .PNG, y guardarla dentro de los archivos del programa, y, arrastrarla hasta la pantalla donde se requiere usar, ya sea como una imagen estática o como parte de algún *widget* como lo puede ser un botón.

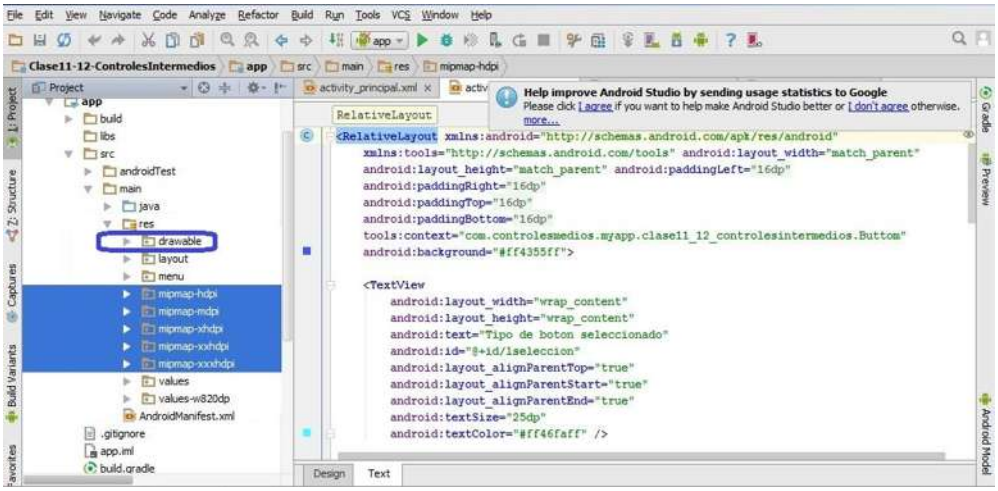


Figura 102Pantalla de Android Studio, señalando la carpeta "drawable"

Se debe acceder a la carpeta *drawable,* dentro del árbol de carpetas del proyecto como se muestra en la **Figura 102** una vez en esta carpeta, se debe guardar la nueva imagen que se tenia previamente, ahí aparecerán opciones pare renombrarla, una vez teniendo la imagen ya guardada dentro de nuestro proyecto de Android Studio,es posible disponer de ella, según las necesidades, bastará con colocar el contenedor de imágenes que necesitemos en la vista del teléfono y, acto seguido, colocar la imagen que se desea.

3.1.4 Cámara

Antes de utilizar la aplicación es necesario solicitar los permisos correspondientes para hacer uso de la cámara del dispositivo como se vio en **2.4,** una vez que concluido esto, se debe de agregar un *widget* en la vista del teléfono que divida la pantalla, también son conocidos como *layouts*, esto es para reservar un espacio en blanco donde se colocará la imagen proveniente de la cámara.

En la **Figura 103** se aprecia un ejemplo de cómo los *layouts* dividen la pantalla y reservan ciertos espacios para utilizar con diferentes *widgets.*

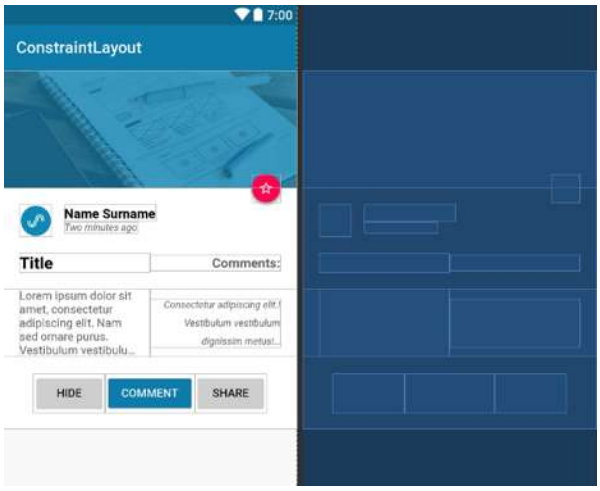


Figura 103Imagen de muestra representando Layouts

El asistente de opencv proporciona una manera rápida de visualizar la camara en el *layout* que dejamos en blanco, a través de un *canvas.*

<org.opencv.android.JavaCameraView dentro del layout que destinamos para la imagen de la cámara, por último en nuestro *mainactivity*, se procede a solicitar el uso de la cámara con la función: ***mOpenCvCameraView.enableView();***esta función solicita el uso de la cámara por medio de OpenCV, estos permisos autorizan el uso de los datos enviados por la cámara hacia el almacenamiento, si se desea modificar dichos datos, se debe programar el *Canvas* que nos proporciona la biblioteca usada. [28]

3.2 Captura de imagen

Como se comentó en **2.4.2.1.** las imágenes digitales son un conjunto de pixeles que van ordenados a manera de una matriz, está matriz tiene distintos tamaños dependiendo del dispositivo con el que se obtengan, una mayor matriz permite tener una mejor resolución de imagen pero se requiere de más recursos para el análisis de la imagen debido al tamaño.

La captura de imagen es innecesaria para este proyecto de tesis, ya que no requiere una captura de imagen, sino, un análisis rápido de ella, la imagen se almacena en el buffer de recepción de datos hasta que otra imagen llegue a reemplazarla, como se realizara con video en tiempo real, quiere decir que, se efectuará está actividad alrededor de 24 veces por segundo. El proceso de analizar una gran cantidad de imágenes en tan poco tiempo tiene como consecuencia una velocidad de cálculo atenuada. Esta cuestión depende del hardware y software de cada dispositivo en el que se quiera implementar la aplicación.

3.2.1 Conversión a escala de grises

Una imagen grande con colores es difícil de manejar, recordemos que cada color primario (RGB) tiene su propia matriz en la imagen de colores que al combinarse crean los colores que se aprecia, en la **Figura 104** se aprecia un ejemplo de cómo están colocadas las capas de matrices de una imagen a color. Cada matriz es de dos dimensiones.

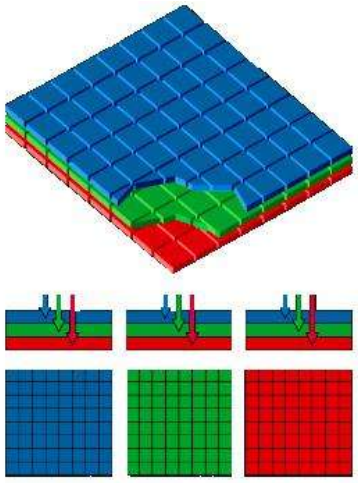


Figura 104 Tres Matrices de dos dimensiones representando colores primarios

Existen diversos métodos para realizar la conversión a escala de grises, unos con mejor resolución que otros, pero, más sencillos de aplicar, se tienen tres métodos frecuentes, los cuales son:

- Método de brillo.
- Método de promedio.
- Método de luminosidad.

El **método del brillo** hace referencia directa en la imagen, el color negro representa cero de brillo, mientras que, el blanco tiene 255 de brillo como ya se vio en **2.4.1,** si una imagen tiene un color muy saturado de brillo, con este método lo convertiría en un tono blanco, entre más opaca sea la imagen, más oscuro será el pixel usado en la nueva imagen, como referencia usamos la imagen de la **Figura 105** y en la **Figura 106** vemos la conversión usando este método.

Figura 105 Imagen original a color

Figura 106 Imagen procesada por el método del brillo

El **metodo del promedio,** es el más conocido y facil de usar, como su nombre lo indica, basta con hacer un promedio de las tres capas, pixel por pixel, para lograr una imagen en la escala de grises, suponiendo que en la **Figura107** tiene dimensiones ***M*** y ***N,*** donde M es el ancho y N el largo, y cada pixel tiene una posición ***(i, j),*** se calcula con la **ecuación 3.** [29]

$$Gris_{i,j} = \frac{R_{I,J1}+V_{I,J\,2}+A_{I,J\,3}}{3} \quad (3)$$

Ecuación 3

Donde **Gris** es la imagen ya convertida en escala de grises de las mismas dimensiones ***M*** y ***N,*** y donde $R_{I,J1}$ es cada pixel de la capa del color rojo, $V_{I,J2}$ es cada pixel de la capa del color verde y $A_{I,J1\,3}$ es cada pixel de la capa del color azul, la sumatoria para cada pixel de ***M*** hasta ***N*** y luego se promedia, de esta manera obtenemos la escala de grises como se ve en la **Figura 108** usando como referencia la misma imagen de la **Figura107.**

Figura107 Imagen original a color

Figura 108 Imagen procesada por el método del promedio

Por último se tiene, el **método de luminosidad,** es un método similar al promedio pero, más complejo. Para el ojo humano es difícil apreciar muchas tonalidades de colores, esto dependerá del color que se observa y su luminosidad, los seres humanos son más susceptibles a ver colores verdes que otros por estar prácticamente en el centro del espectro

de luz visible, le sigue el rojo que es más complicado de ver y por último el azul, color que es más alejado al centro del espectro.

Para realizar este método, es necesario no solo promediar, sino tomar en cuenta a que tonalidades somos más sensibles y en qué medida para poder hacer un escalado de grises mejor, por lo que se toman en cuenta los valores 0.21 R + 0.71 V + 0.07 A, para promediar los valores, es decir, será como en la **ecuación 4.** [30]

$$Gris_{i,j} = \frac{(0.21)*(R_{I,J1})+(0.71)*(V_{I,J\,2})+(0.07)*(A_{I,J\,3})}{3} \quad (4)$$

Ecuación 4

Al resultado de dicha ecuación, podemos ver la **Figura 110** producto de la misma imagen de la **Figura 110.**

Figura 109 Imagen original a color

Figura 110Imagen procesada por el método de luminosidad

Para hacer una comparación más directa, en la **Figura 111** se muestra una pequeña paleta de colores, a su derecha esa misma paleta en escala de grises usando el método de brillo, seguido por el método del promedio y por último usando luminosidad.

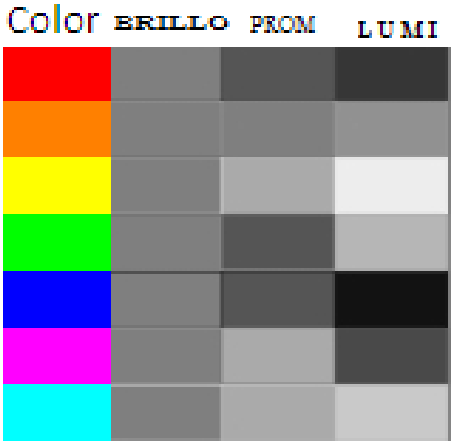


Figura 111Paleta de colores y distintos métodos de procesamiento para blanco y negro

Se aprecian de manera más clara las diferencias entre los distintos métodos, esto no indica cual es mejor o peor, solo que existen maneras diferentes de hacer el procesamiento de la imagen y que dependiendo de la imagen, es conveniente usar uno u otro método.

OpenCV otorga herramientas para realizar esta conversión de manera sencilla, el método que utiliza es el de luminosidad, y es sumamente sencillo de realizar, solo es necesario llamar la función con la variable que contiene la imagen original, una variable con la nueva imagen en escala de grises y por último la instrucción para convertir en escala como se muestra en la siguiente línea.

Improc.cvtColor(imOriginal, imGris, Improc.COLOR_BRG2GRAY);

La salida de la función es una matriz con la imagen en escala de grises, y a continuación solo es necesario presentarla en la pantalla del dispositivo móvil.

3.2.2 Rotar imagen

Al obtener la imagen por medio de la cámara, usando OpenCv, se presenta la problemática de la rotación de la imagen, cuando el dispositivo está en vertical la imagen la captura en horizontal y viceversa, causando problemas de visión, este problema tiene una solución algorítmica y algebraica, se debe encontrar en donde se almacena la imagen, esta es en un buffer que se encuentra en el canvas, como las imágenes son una matriz de pixeles, debemos girar el orden de los pixeles para poder rotar la imagen, se realiza de la siguiente manera, en el canvas, ir a la sección donde se encuentra el mapa de bits (***bitmap***) y se colocan las siguientes líneas de código.

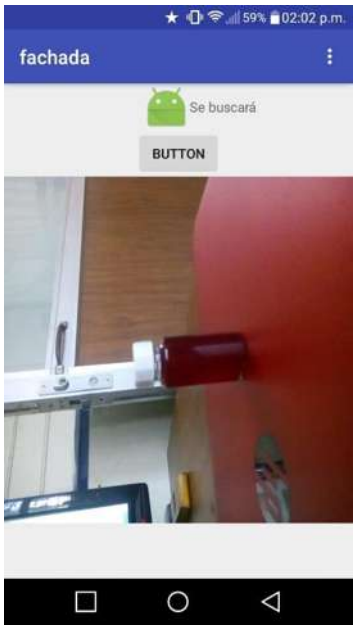


Figura 112 Fachada de la aplicación con la cámara a 90°

```
 canvas.save();
canvas.rotate(90, (canvas.getWidth()/2), (canvas.getHeight()/2));
```

En el código, se guarda el canvas, acto seguido, se modifican, con dicha el metodo *rotate*, requiere tres parámetros, el primero representa los grados en que la imagen va a rotar en sentido horario, el segundo parámetro es sobre el ancho de la imagen y el tercero, el alto de la imagen, se aprecia que lo dividen sobre dos para recudir el tamaño de la imagen a la mitad, como se aprecia en la **Figura 113**.

Figura 113 Vista de la aplicación con la cámara en orientación correcta

3.2.3 Filtrado previo

Como se mencionó en **2.4.2,** las imágenes se les pueden aplicar distintos tipos de filtros, cada filtro interactúa directamente con cada pixel y la forma en la que se relaciona con los pixeles aledaños, para este proyecto en particular, es necesario que la imagen visualizada tenga la menor cantidad de modificaciones, entre ellas, el color de la imagen, es necesario hacer el filtrado y detección de patrones en un matriz paralela a la original, en la original se montará el resultado de la modificación.

Para realizar el filtrado de la imagen, es necesario obtener la transpuesta de la matriz, acto seguido se debe modificar su tamaño, esto se realiza con los métodos:

Core.transpose(mRgba, mRgbaT); para la transpuesta y,
Improc.resize(mRgbaT,mRgbaF,mRgbaF.size(),0,0,0); para modificar el formato de la imagen y, ajustarla a la pantalla del equipo móvil o celular en el que se visualiza.

En este paso, es necesaria la modificación del color de nuestra imagen, no se debe olvidar que, debemos hacer esto en una matriz distinta.

Improc.cvtColor(mRgba, mRgbareal,Improc.COLOR_BGR2GRAY); el cual usa el método de luminosidad para convertir la imagen en escala de grises.

Por último, es necesario que la imagen tenga un filtro en el cual para corregir errores de la imagen inherentes a la toma de la misma o al filtrado de color, para esto es necesario aplicar un filtro Gausiano y un filtro de Canny. La mezcla de ambos genera una imagen que permite apreciar los bordes de la imagen original en su totalidad, de manera similar a como se obtiene previamente, se usaron las líneas de código con bibliotecas que tienen incluidos los siguientes métodos.

Improc.GaussianBlur (mRgba, mRgbareal, size, 0,0, BORDER_DEFAULT);

Improc.Canny(mRgba, mRgbareal, 80, 100);

Después del filtrado, se obtiene una imagen adecuada para realizar la detección de patrones en la matriz.

3.3 Detección de círculos.

Para la detección de círculos se aplica la **Transformada de Hough,** es un método de extracción de características. Su creador es **Paul Hough,** esta transformada se utiliza en la detección de bordes en imágenes digitales.

Originalmente nace de la idea de detectar líneas rectas, en la actualidad se utiliza para la detección de diversas formas geométricas, para la función se usará la detección de círculos.

La transformada de Hough para círculos funciona estableciendo la ecuación de la forma básica de un círculo, la cual es:

$$(x + a)^2 + (y - b)^2 = r^2 \quad (5)$$

Ecuación 5

Donde (a,b) es el centro del círculo y (x,y) es un punto en el perímetro del círculo, dando como resultado ***r*** que es el radio. Para la transformada es necesario definir los intervalos de ángulos entre ***0°*** y ***360°*** donde se establecen el valor de ***r*** si lo conocemos. Posteriormente, se crea una matriz que tenga valores en cero donde se introducirán los valores de (a,b), los cuales deberán ser, la solución a la ecuación del círculo.

Se evalúa cada punto del borde *(x, y)* en la **Ecuación 5** de cada pixel de la matriz para ver cuántos puntos son concordantes con (*a*, *b*) propuestos. Para finalizar, se hace una relación de los valores de la matriz para los puntos con más ocurrencia para sobre ellos dibujar el círculo con el radio correspondiente a los parámetros en la imagen original.

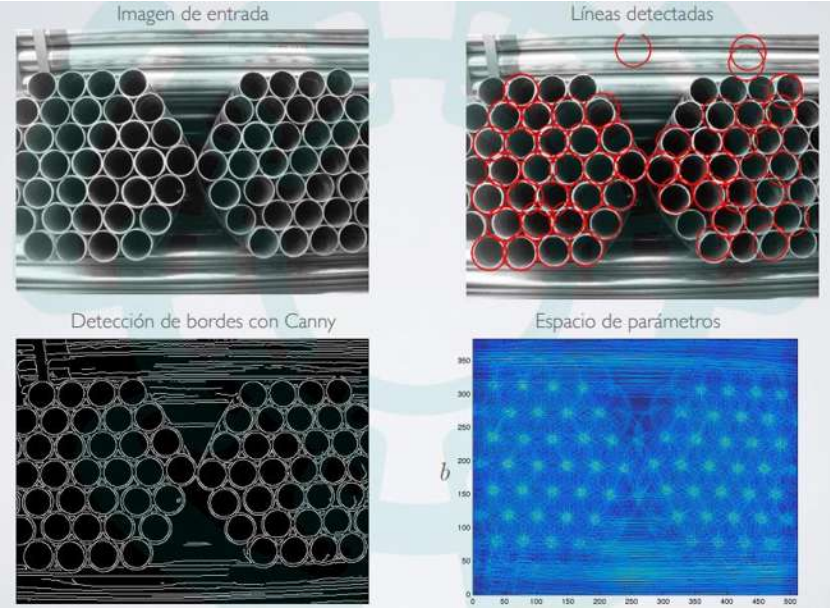


Figura 114 Procesamientos de una imagen utilizando distintos filtros

En la **Figura 114** se aprecia la imagen original, luego el filtro para la detección de bordes donde se utiliza la transformada de Hough para su obtención, la siguiente imagen del espacio de parámetros son todas las posibilidades de círculos que puede encontrar, muchos de las detecciones corresponden a sombras y elementos engañosos en la imagen, se implementa la discriminación específica de círculos para detectar solo aquellos que tengan mayor repetición en sus puntos *(a, b)* y, el resultado de esa selección nos devuelve los círculos detectados en la imagen original.

En el método que se encuentra en la biblioteca OpenCV, solo se colocan de manera correcta los parámetros más adecuados.

[[Improc.HoughCircles(input, circles, Improc.CV_HOUGH_GRADIENT, 1, 100, 100, 90, 10, 130);]]

Se aprecia una serie de valores que de primera vista no son entendibles. Cada valor es referente a parámetros específicos del método usado. El primer dígito, corresponde a dp que es la resolución del inverso del radio del círculo que buscaremos. Por esa razón, el valor es 1. El siguiente dígito representa la mínima distancia entre círculos que detecte, min_dist, ésta distancia la pondremos de 100 puntos procurando que detecte solamente un círculo a la vez. Los dígitos siguientes son parámetros para detectar los límites externos e internos del círculo respectivamente. El primer valor debe ser mayor que el segundo. 100 puntos y 90 puntos son suficientes para realizar la buena detección.

Por último, los dos dígitos restantes representan el radio mínimo, ***min_radius*** y el radio máximo, ***max_radius***. Con esto, podemos filtrar los resultados de la búsqueda para tener círculos de tamaños predeterminados y que el programa no los confunda con aquellos que son distintos. Los valores para éste proyecto son de 10 puntos y de 130 puntos. Si se desconoce el tamaño promedio que se busca, bastará con colocar 0 puntos y 0 puntos, para que de esa manera, el método aplicado detecte todos los tamaños que pueda.

El filtrado y detección de círculos se realiza en una imagen modificada. Posteriormente se coloca la información obtenida sobre la imagen real, donde se podrá apreciar los contornos de los círculos en la imagen a color, que tomamos al inicio. Para ello es necesario tener una matriz con los círculos detectados y plasmarla sobre la imagen real. Una vez realizado, es necesario colocar el color a esta detección, lo cual se realiza con el siguiente método.

Improc.circle(input, center, radius, new Scalar(255,0,0), 2);

En Input se inserta la imagen original, un centro y el radio obtenido en la detección de círculos. Por medio de un escalar plasmamos los colores deseados, el primer valor es la concentración del color rojo, el segundo es el verde y el último es el azul, el dígito fuera del escalar es el grosor de la línea dibujada sobre el círculo como se muestra en la **Figura 115**.

Figura 115Detección de círculos usando procesamiento digital de imágenes

3.4 Configuración del bluetooth.

Dadas las necesidades de este proyecto, se utiliza el modulo bluetooth HC-05 como el que se muestra en la **Figura 116** el cual contiene 4 pines de conexión y funciona en un rango de voltaje de 3.6V a 6V, características que lo hacen conveniente para este tipo de circuitos pequeños. Tiene además, la característica de que es rápido de usar al contener, también, un pin **RX** para recibir información y un **TX** para enviar información de manera serial con una computadora. En éste caso, un microcontrolador.

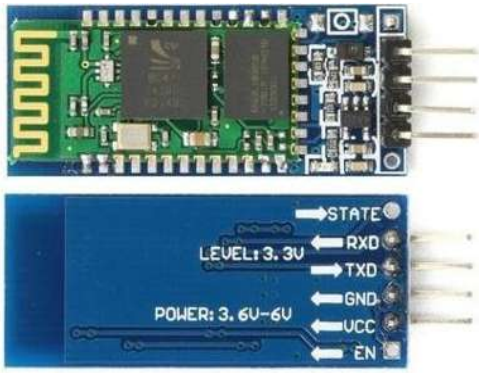


Figura 116Modulo Bluetooth con comunicación serial

Otra ventaja del módulo Bluetooth, es su capacidad de trabajar tanto en modo "maestro" como en modo "esclavo" para las conexiones. En éste caso, se configura en modo "esclavo", las configuraciones se realizan por medio de comandos AT, dichos comandos nos permiten interactuar con el módulo de manera sencilla. Estos comandos permiten modificar el nombre, la clave PIN, rango y hasta su rol como maestro o esclavo, para conectarlo al microcontrolador, basta conectar la terminal RX del Bluetooth al TX del microcontrolador y viceversa, este módulo utiliza la **comunicación serial,** la velocidad de comunicación dependerá de la velocidad de recepción y transmisión de los dispositivos involucrados.

Dado que no se modificarán los valores del módulo, basta con abrir el programa del microcontrolador e iniciar una comunicación por medio de UART del microcontrolador. A continuación, se muestran los pasos necesarios para la configuración del puerto serie:

1.- Activar el puerto serie.

2.- Inicio de la comunicación entre el microcontrolador y el modulo Bluetooth.

3.- Mientras no esté activo, mantener en modo de bajo consumo.

4.- Al recibir información, activar el protocolo de control de los motores.

5.- Controlar el movimiento de los motores, modificarlo o detenerlos.

Estos son los comandos más importantes:

```
1. - __interrupt void USCI0RX_ISR(void){

2. - while (!(IFG2&UCA0RXIFG));
```

El primero habilita las interrupciones del puerto serie, el segundo, espera que se active la interrupción para atenderla.

3.4.1 Bluetooth en Android.

Como se vio en **3.4,** se requieren permisos especiales que permitan hacer uso de los recursos con los que cuentan los dispositivos móviles. En la configuración actual, lo primero que se hace es habilitar el bluetooth, se solicita el permiso al usuario, una vez que se permite, es necesario crear un socket virtual en la aplicación que nos permite realizar transferencias de información entre el móvil y el modulo. Se configura para que únicamente sea posible conectarse con el módulo, se guarda el nombre y el pin de emparejamiento.

Dentro de la programación del proyecto, existen diversos casos como:

1.-Iniciar la actividad de bluetooth.

2.-Buscar el módulo bluetooth en la conexión presente.

3.-En caso de no detectar el módulo bluetooth, envía mensaje de alerta al usuario.

4.- Si detecta el módulo, intenta la conexión entre el dispositivo móvil y el bluetooth.

5.-En caso de no lograrse la conexión, envía un mensaje de error al usuario.

6.-Cierra la aplicación.

7.- Si la conexión se logra correctamente, el módulo espera recibir información.

8.- Realizar el intercambio de información entre el módulo y el dispositivo móvil.

9.- Si no hay información que recibir, el módulo bluetooth entra en modo de bajo consumo en la espera de información.

10.- Al realizar la desconexión, envía alerta al usuario de la desconexión.

11.-Si no realiza la desconexión de manera correcta, manda un error al usuario.

12.-Cierra la aplicación.

Para este proyecto, al intentar conectar el dispositivo, aparece el mensaje de conexión exitosa o, en caso contrario, un error de conexión, el cual cierra la aplicación para reiniciarla e intenta nuevamente la conexión, cuando la conexión es exitosa, el mensaje de nuestro botón cambia de "*BLUETOOTH*", a "*DESCONECTAR BT*", esto con la idea de que el usuario tenga como referente visual el momento en que se realiza la conexión, tal como se muestra en la **Figura 117**

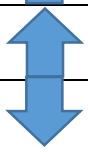
Figura 117Ejemplo del cambio de texto en los botones al ser activados

A la par, es necesario establecer mensajes para desconectar el dispositivo, sino puede realizar dicha acción, muestre un mensaje de error, el cual impedirá la desconexión hasta que las fallas terminen. Cuando no existe un módulo al cual conectarse, el dispositivo mostrará un dialogo donde indica que no se detecta un lugar físico al cual poder conectarse de manera en que fue programado.

Una vez realizada la conexión y habilitar el socket, es necesario establecer un protocolo, en este proyecto no es necesario que el módulo envié información al dispositivo, y, solamente el dispositivo usará la transferencia de información, para esto es necesario crear una lista de distintos casos, cuando el círculo salga del centro de la pantalla, el dispositivo debe informar al módulo para que a su vez, este ejecute alguna acción de las establecidas previamente, dentro del socket se habilita la parte de la transferencia de datos, la cual se envía en base hexadecimal para que pueda ser interpretada de manera correcta por el microcontrolador. Se establecen cuatro casos, el primero es que al detectar el círculo dentro del costado derecho de la pantalla, envié ***0x04,q***ue significa que el vehículo girará a la izquierda, por el contrario si es del lado opuesto, enviará ***0x03*** girando a la derecha, cuando el objeto se encuentre centrado en la imagen, el dispositivo enviará ***0x05,*** que detendrá los motores y**,** si no es cualquiera de estos casos, el dispositivo transferirá la información de ***0x02*** que indica que avanzarán ambos motores.

En la **Tabla 2** se aprecia el sentido de giro de cada motor de acuerdo al código que llega al módulo bluetooth, para avanzar es necesario que los motores giren en el mismo sentido, mientras que para girar en cualquier sentido es necesario que los motores giren en sentido contrario, siendo el motor que gire hacia adelante el que dirija el movimiento.

Tabla 2 Interpretación del sentido de giro de los motores para cada código posible

Código hexadecimal	Dirección motor derecho	Dirección motor izquierdo
0x02		
0x03		
0x04		
0x05		

El código se envía mediante el bufferTx, mientras que la recepción es en el bufferRx, los datos que lleguen a este buffer son eliminados si no se almacenan, por ellos se establece una variable para guardar estos datos, una vez leídos, son purgados del programa en la espera de nuevos datos.

3.4.2 Conexión

La conexión física de Bluetooth trabaja a través de un estándar de radio frecuencia de baja potencia en la banda **ISM** de los 2,4 GHz, lo que la convierte en una comunicación inalámbrica, que logra mantener una velocidad de transmisión permanente consiguiendo no sólo una velocidad notable en la transferencia de datos, sino también un ahorro de las baterías de los equipos.

Figura 118 Conexión Bluetooth

A través de Bluetooth se envían señales de baja potencia que no llegan a superar 1 milivatio, limitando el rango de acción de los equipos a un máximo de 10 metros de distancia, con el objeto de que la comunicación entre los dispositivos no pueda ser interferida por otras señales. De esta manera, los datos viajan de manera eficaz y segura.

La conexión bluetooth varia de dispositivos a dispositivos, es complicado que un dispositivo pueda realizar múltiples conexiones simultaneas, es decir, como se aprecia en la **Figura 118,** un dispositivo de audio no puede establecer conexión con más de un dispositivo emisor a la vez, en caso de entrar en conflicto, el bluetooth se conectara con el último dispositivo que ha mandado la solicitud o que tenga preferencia.

3.4.3 Emparejamiento

El emparejamiento es similar a intercambiarse el número de teléfono. Del mismo modo que intercambias el número de teléfono con una persona a la que quieres llamar, para conectar dispositivos Bluetooth, primero es necesario emparejarlos, a fin de registrar la información de conexión de cada uno de ellos, para realizar esto, es necesario que intercambien un código de seguridad, el cual permite que no cualquier dispositivo pueda conectarse sin previa autorización. Después de vincular los dispositivos por primera vez, ya no es necesario repetir este proceso de emparejamiento. Ello se debe a que cada dispositivo ha guardado las credenciales necesarias y, por lo tanto, puede conectarse fácilmente. [34]

Figura 119 Emparejamiento Bluetooth

En la **Figura 119** se aprecia el equivalente de la conexión bluetooth haciendo referencia a una presentación entre dos personas que antes no se conocían, luego del primer encuentro, no es necesario volverse a presentar, ya serán conocidos.

Este código de emparejamiento es una contraseña que el usuario puede modificar en la mayoría de los dispositivos, por defecto la contraseña suele ser "*0000"* o *"1111",* este emparejamiento es parte del protocolo de seguridad, para encargarse de que ningún dispositivo que no conozcas trate de enviar información.

Al emparejarse se guarda el número de conexión, el código de seguridad y el nombre del dispositivo, esto como parte de las credenciales de autenticación, es decir, para que dos personas puedan comunicarse, deben conocerse si es que quieren repetir la conexión.

3.4.4 Protocolos

Los protocolos describen cómo se realizan las tareas básicas como señalización telefónica, gestión de enlace y lo que se conoce como Service Discovery, es decir, determinación de qué servicios están disponibles desde o través de otros productos Bluetooth.

Bluetooth se rige bajo un estándar denominado **IEEE 802.15.1** el cual permite la conexión entre dispositivos, este estándar cuenta con una pila de protocolos como se muestra en la **Figura 120**.

Figura 120 Pila de protocolos bluetooth

Bluetooth está definido como un protocolo de arquitectura de capa que está formado por unos protocolos centrales, protocolos de reemplazo de cable, protocolos de control de telefonía, y protocolos adoptados. Como mínimo, toda pila de protocolos de Bluetooth debe tener los siguientes protocolos: LMP, L2CAP y SDP. Además, los dispositivos que se comunican por Bluetooth pueden usar casi siempre los protocolos HCI y RFCOMM.

- **LMP.** El protocolo de control de enlace (Link Management Protocol, LMP) se usa para el establecimiento y control del enlace de radio entre dos dispositivos. Está implementado en el controlador.

- **L2CAP.** El protocolo de control y adaptación del enlace lógico (Logical Link Control and Adaptation Protocol, L2CAP) es usado para multiplexar conexiones lógicas entre dos dispositivos que usan diferentes protocolos de nivel superior. Proporciona segmentación y reemsamblado de los paquetes.

- **SDP.** El protocolo de descubrimiento de servicio (Service Discovery Protocol, SDP) permite a un dispositivo descubrir servicios que ofrecen otros dispositivos y sus parámetros asociados. Por ejemplo, cuando usas un teléfono móvil con unos auriculares Bluetooth, el teléfono usa SDP para determinar qué perfil de Bluetooth pueden usar los auriculares y los ajustes del protocolo de multiplexor necesarios para que el teléfono pueda conectarse con los auriculares. Cada servicio está identificado por un **UUID** (Universally Unique Identifier).

- **RFCOMM.** (Radio Frequency Communications) es un protocolo de reemplazo de cable usado para generar un flujo de datos virtual en serie. RFCOMM ofrece transporte de datos binarios y emula las señales de

control a través de la capa de banda base del Bluetooth. Ofrecen un flujo de datos confiable y sencillo para el usuario, similar a **TCP.**

- **BNEP**. Protocolo de encapsulación de red de bluetooth (Bluetooth Network Encapsulation Protocol) transmite paquetes IP de perfiles de red de área personal, similar al **SNAP** pero de manera local.

- **AVCTP**. Protocolo para el transporte de audio y video (Audio/video Control Transport Protocol) es usado por el perfil de control remoto para transferir órdenes de control de audio/vídeo a través de un canal L2CAP. Los botones de control en un auricular estéreo usa este protocolo para controlar el reproductor de música.

- **TCS**. El protocolo de telefonía binario (Telephony Control Protocol - Binanry) es el protocolo orientado a bits que define la señalización de control de llamadas para establecer llamadas de voz y datos.

- **WAE.** Especifica un marco de aplicación para los dispositivos inalámbricos y WAP es un estándar abierto que permite a los usuarios móviles acceder a los servicios de información y telefonía.

3.5 Software de microcontrolador.

En **2.10.1** se justifica el uso del microcontrolador de Texas instruments por su grandes ventajas y su precio económico, de esa misma línea de producción es elegido el microchip **msp430g2553**.

La posibilidad de elegir el lenguaje de programación que se emplea en el microcontrolador gracias a ***CODE COMPOSER STUDIO,*** de los mencionados se ha optado por la implementación de lenguaje C, esto debido a la facilidad de programación.

3.5.1 Función del microcontrolador.

Es necesario describir la función del programa antes de realizarlo, con el objeto de tener clara la finalidad y el mejor método para atacar la necesidad.

Se busca que A interactúe con B, donde B realizará las acciones motoras o de movimiento, es decir, A analiza y ordena mientras que B obedece, A es el Smartphone y B el vehículo que transporta a A.

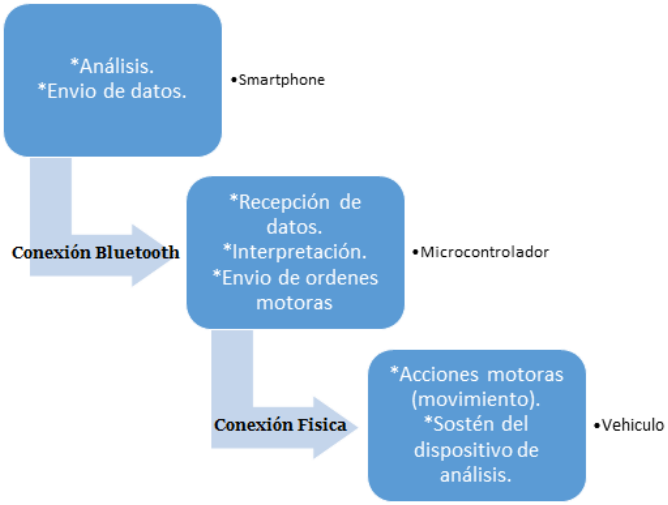


Diagrama 1 Diagrama de interacción

La comunicación entre el smartphone y el vehículo, como se ha mencionado en capítulos anteriores, se realizará por medio de protocolo de Bluetooth, por lo que el Smartphone debe tener enviar información por esta vía, mientras que el microcontrolador debe recibir la información enviada, interpretarla y realizar una acción conferida a los datos recibidos.

1. Inicio del programa, se detienen los temporizadores del guardián.
2. Analizar si el oscilador del microcontrolador está oscilando a frecuencia estable.
3. Si el oscilador está estable, continúa la inicialización del programa, caso contrario, espera a que esté estable.
4. Se declaran cuatro pines como salidas, las cuales son las que controlan los motores, dos por motor, de ambos pines, el activo indica el sentido de giro.
5. Se configura el puerto UART que funciona como interfaz de conexión y comunicación con el dispositivo Bluetooth.
6. Dentro del UART se configuran dos pines, uno para recepción (RX) y otro para transmisión (TX).
7. Configurar velocidad de comunicación de 9600 Baudios.
8. Se habilitan las interrupciones para el buffer de recepción.
9. Se mantiene abierto el buffer de recepción.
10. Cuando llega un dato al buffer, se inicia una máquina de estados.
11. Se establecen los algoritmos de interacción con los motores.
12. Caso 1: Ambos motores van hacia adelante.
13. Caso 2: Ambos motores van hacia atrás.

14. Caso 3: Motor derecho hacia adelante, motor izquierdo hacia atrás para hacer un giro.
15. Caso 4: Motor derecho hacia atrás, motor izquierdo hacia adelante para hacer un giro contrario al anterior.
16. Caso 5: Se detienen los motores.
17. Dependiendo del dato recibido, se activa un caso dentro de la máquina de estados.
18. Mientras no se reciba otro dato, continúa con el estado anterior.
19. Fin del programa.

3.5.2 Puertos

En **2.10.1,** se muestran los diversos periféricos del mirocontrolador, para el correcto desarrollo es necesaria la correcta activación de cada uno de los pines, considerando la utilización de dos motorreductores, es necesario tener 4 salidas, dos por cada motor, de esas dos, una por cada sentido de giro.

Para el motor A se usan los pines 2.3 y 2.4, uno para cada sentido de giro, por su parte, los pines 2.5 y 1.6 corresponderán al control del motor B.

Para la comunicación entre el módulo bluetooth y el microcontrolador, se dispone del puerto serie, para ello se requiere de los pines 1.1 y 1.2 que representan el RX y TX del microcontrolador, respectivamente.

En la **Figura 121** se puede apreciar cuales son los pines que están siendo utilizados, de color rojo aquellos para los motores y de color azul aquellos para la comunicación serial.

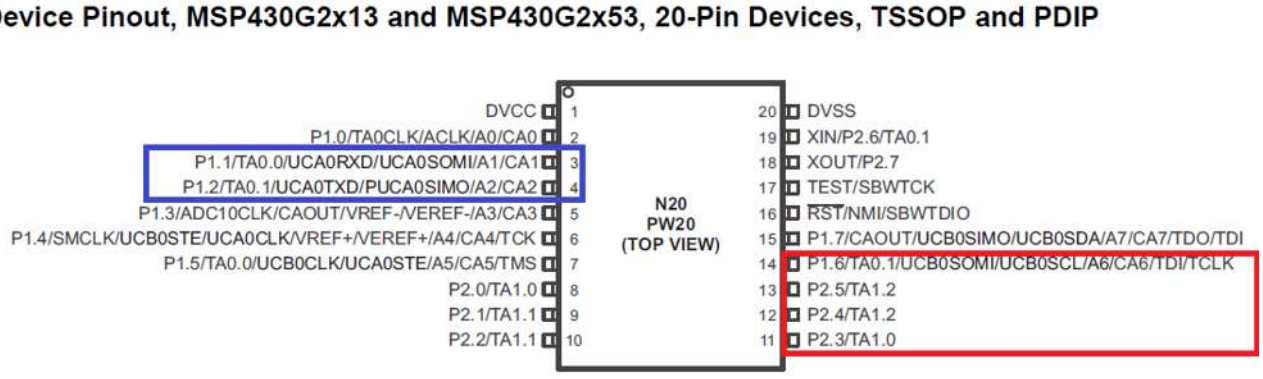


Figura 121Pines utilizados del mspg2553

Se consideró la ubicación de los pines para el desarrollo de un **PCB** .

3.5.3 Interrupciones

En la programación secuencial, una vez que se pasa una instrucción, es necesario que se repita el programa ejecutar la misma instrucción, en muchas ocasiones los

micrcontroladores deben responder a eventos asíncronos en tiempo real. En los programas secuenciales es probable que sea ignorado el evento.

Las interrupciones, deben realizar las siguientes tareas:

- Habilitar las interrupciones
- Atender la interrupción
- Finalizar la interrupción

Si el problema ocurre durante la producción de un programa, el mismo programa debe estar constantemente revisando las interrupciones en caso de encontrar alguna, salta a esa instrucción, atiende la interrupción, manda una señal de que ha resuelto el problema y regresa a la parte del programa donde se encontraba antes del salto.

Supongamos un salón de clases, mientras el docente explica la materia, un alumno levanta la mano para hacer una pregunta, esta sería la interrupción, el docente al percatarse de la mano levantada, detiene la materia mientras atienda la duda del alumno, una vez atendida, pide que el alumno baje la mano y no la levante, sino, hasta que tenga otra interrupción, de esa manera finaliza la acción y el docente continúa donde se quedó.

En el programa desarrollado, las interrupciones están en espera de que el módulo bluetooth reciba alguna señal, mientras esto no ocurra, el vehículo se encuentra realizando la última acción, en cuanto el módulo recibe una señal, la envía por puerto serie al microcontrolador, el microcontrolador detiene la acción previa y atiende la interrupción, cambiando el estado de la trayectoria anterior.

Para hacer uso de las interrupciones en **code composer,** es necesaria una función dentro del programa, especificando que se trata de una interrupción.

3.5.4 Algoritmo de seguimiento

Para el algoritmo es necesario implementar en el microcontrolador, una máquina de estados, como se vio en **3.4.3**, se tienen las siguientes funciones:

1. Adelante
2. Atrás
3. Vuelta a la derecha
4. Vuelta a la izquierda
5. Alto

Como se aprecia en la **Figura 122** la imagen se ha segmentado, el punto de referencia es el cuadrado del centro, y, el vehículo tiene como objetivo centrar la imagen.

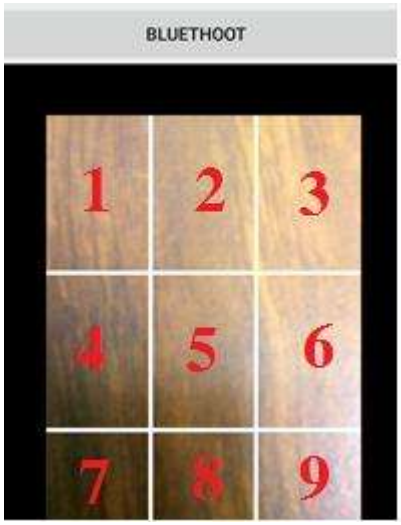


Figura 122 Cuadricula de la aplicación como referencia

En caso que el círculo es detectado en el costado derecho de la imagen (cuadrantes 3, 6 o 9), el vehículo aplicará un giro a la derecha, es decir activará el motor izquierdo en sentido de las manecillas del reloj y el motor derecho en sentido contrario a las manecillas, mientras que, el caso contrario, si el círculo se encuentra del costado izquierdo (cuadrantes 1, 4 o 7), el motor izquierdo gira en sentido contrario a las manecillas mientras que el derecho gira en sentido hacia las manecillas del reloj.

Cuando el círculo se encuentra en la parte superior de la cámara (1,2 o 3), ambos motores irán en reversa intentando centrar la imagen, en caso contrario, si se encuentra en la parte inferior (cuadrantes 7,8 y 9), ambos motores giraran hacia adelante.

Para que esto suceda, es necesario que el dispositivo Android, envié por bluetooth un código para cada caso, el código enviado son los números “1”, “2”, “3”, “4” y “5”, correspondientemente en hexadecimal, el módulo bluetooth recibe estos códigos, los compara con la máquina de estados y realiza la acción necesaria según los casos presentados.

Capítulo 4. Hardware Implementado

Para un correcto uso del proyecto, es necesario, un vehículo a pequeña escala, capaz de controlarse por medio de la cámara del Smartphone. Buscando reducir costos y tamaño, del hardware, se considera en una plataforma que no tenga dimensiones mayores a los 25 cm de largo.

4.1 Dimensiones y características de la plataforma.

Las medidas seleccionadas son 25cm de largo y 10cm de ancho, suficiente para colocar en su superficie el circuito impreso, las baterías y el smarthphone, se piensa en un material ligero. La primera elección es MDF, por su bajo costo en relación a otros materiales derivados de la madera, es ligero y, en caso de ser necesario, es posible cortarlos con láser para crear figuras específicas.

Durante el proceso de selección del material y la compra, se seleccionó un kit de inicio para estudiantes, el cual contiene un plataforma de acrílico transparente, soportes para los motores, un socket para baterías doble A y, los motores que en un inicio se decidió usar, como se aprecia en la **Figura 123**.

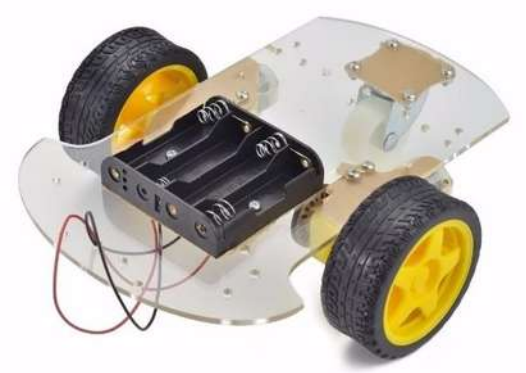


Figura 123 Plataforma que conforma el cuerpo del vehículo a controlar

El kit mostrado es más económico que comprar las piezas por separado y construir un diseño en MDF, se optó por adquirirlo y montarlo a nuestro gusto moviendo algunas piezas de lugar para lograr la distribución adecuada de los componentes adicionales necesarios para el proyecto.

Dentro del diseño se tiene una tercera rueda, conocida como "rueda loca" la cual funciona para estabilizar el vehículo.

En la **Figura 124** se pueden apreciar las medidas de la plataforma, 147mm de ancho sin contar el grosor de las llantas y 208mm de largo, la altura dependerá del diámetro de las llantas

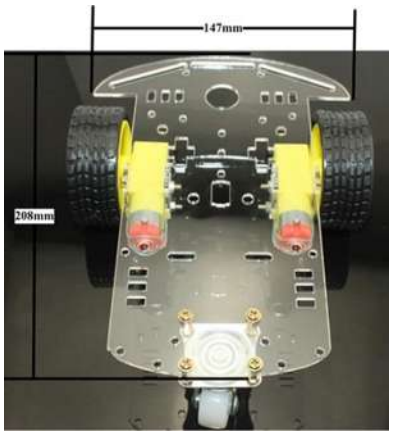


Figura 124Dimensiones del chasis

4.2 Ruedas y llantas utilizadas

Para tener movilidad es necesario tener objetos circulares que faciliten el desplazamiento, para ello usa dos llantas y una rueda loca, todas las piezas vienen incluidas en el kit ZK2 del carro robótico, en la **Figura 125** se aprecian las dimensiones de las ruedas principales, van unidas al motorreductor, el diámetro interno del eje de la llanta es de 2mm encaja con el eje externo del motor de la misma dimensión, tiene forma de cuña para evitar giros libres de movimiento. El diámetro la llanta es de 66mm, por lo que ¾ partes de esa distancia se añaden de altura a la plataforma, el ancho de las ruedas de goma es de 28mm.

Figura 125 Dimensión de la rueda a utilizar

Las ruedas trasladan el movimiento gracias a los motores, la "rueda loca" se decide posicionarla en la parte trasera para cumplir el propósito de estabilidad, la rueda loca mide las ¾ partes de la altura de las llantas, de la base a la superficie de la llanta loca, como se ve en la **Figura 126**.

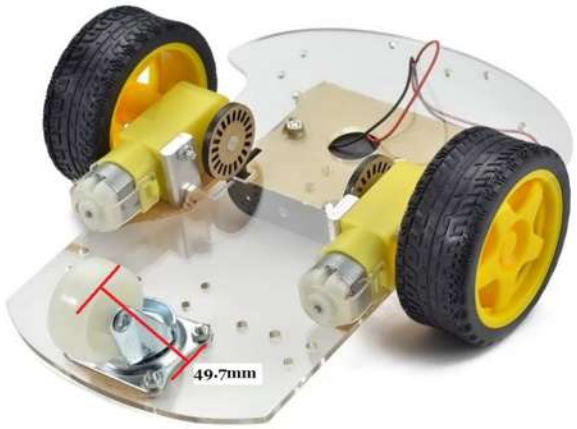


Figura 126 Altura de la "rueda loca"

4.3 Motorreductores utilizados

Como ya se vio en **2.5.1,** existe una gran cantidad de motores en el mercado que satisfacen nuestras necesidades, la elección fue basada en el costo, el kit un par de motorreductores que, convenientemente, son los seleccionados.

Sumado a las ventajas de dicho motorreductor, se contempla la facilidad de adquisición de componentes necesarios para el proyecto en particular, donde emplearemos ruedas para el movimiento lineal del vehículo, dichas ruedas pueden conseguirse en tiendas locales de electrónica, a bajo costo y tamaño proporcional al torque dado por el motorreductor, como se aprecia en la **Figura 127** vemos la comparativa del tamaño y se presenta el producto adquirido, cabe resaltar que se ocupan dos motores con dos llantas, para ventaja del consumidor, dichos elementos, vienen en la compra del kit al que se hace referencia en **4.1.**

Figura 127 Motorreductor y la llanta a utilizar

Las especificaciones del motorreductor son las siguientes [31]:

- Velocidad sin carga (3V): 120rpm.
- Velocidad sin carga (6V): 240rpm.
- Corriente sin carga (3V): 40mA.

- Corriente sin carga (6V): 70mA.
- Fuerza de torque (3V): 3200mg*cm.
- Fuerza de torque (6V): 5500mg*cm.
- Peso: 40g.
- Dimensiones: 65mm largo * 25mm alto *20mm.ancho

Figura 128 Dimensiones motorreductor

4.4 Controlador de los motores.

Para controlar ambos motorreductores es necesario usar un componente de potencia, ya que el microcontrolador es incapaz de suministrar la corriente necesaria para un buen funcionamiento. Para ello usaremos un "***Puente H"*** este dispositivo lleva el nombre derivado de su representación gráfica como si fuera un H, tal como se muestra en la **Figura 129**.

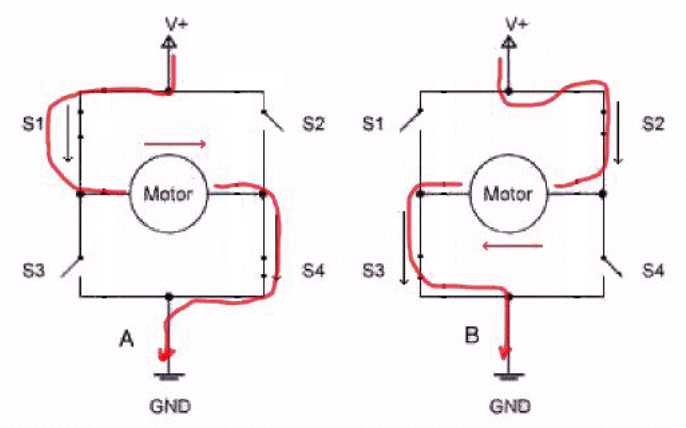


Figura 129 Diagrama de flujo eléctrico en un puente H

En la **Figura 129** podemos observar no solo la forma de H que tiene, sino, también la circulación de la corriente, este dispositivo permite modificar la dirección de giro de los motores, con esto podemos realizar el movimiento en sentido de giro de las manecillas del reloj o en su contra.

Con base al diagrama presentado, es necesario que nunca se activen los interruptores S1 y S3 al mismo tiempo, tampoco S2 y S4 de la **Figura 130**, esto ocasionaría un cortocircuito en nuestra fuente de alimentación.

Existen distintos tipos puentes, con combinaciones de transistores y hasta con compuertas lógicas, para el proyecto se decide utilizar el **L293D,** un puente H de fácil adquisición, precio accesible y fácil de utilizar, de la hoja de datos extraemos el diagrama de uso que vemos en la **Figura 130**.

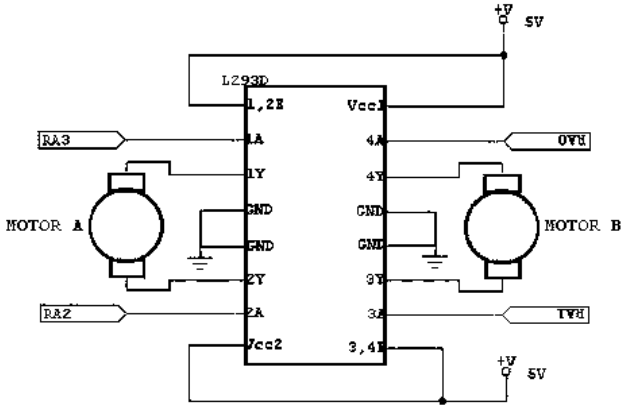


Figura 130 Conexión de un puente H a los motores y entradas del microcontrolador.

En la **Figura 130** donde se indican n las conexiones a realizar con el dispositivo, la parte inferior es alimentada por una fuente de poder externa a nuestro microcontrolador como se comenta al inicio de este tema, donde vemos las etiquetas de ***RA0, 1, 2, 3*** indica los pines donde se conecta el microcontrolador para indicar la dirección de cada uno de los motores, a diferencia de como vimos anteriormente, aquí se pueden activar las dos entradas del microcontrolador de cada rama, lo cual ocasionaría un paro en la dirección del motor, el motor girara en un sentido según como se active desde el microcontrolador, si activamos ***RA3*** solamente el motor izquierdo tendrá movilidad, si por el contrario, activamos ***RA2***, la movilidad del motor será inversa a la anterior, lo mismo ocurre con los otros puertos de entrada, pero ocurriendo, en el motor derecho.

En la **Figura 131** podemos ver cómo funciona el **L293D** usando comparadores lógicos.

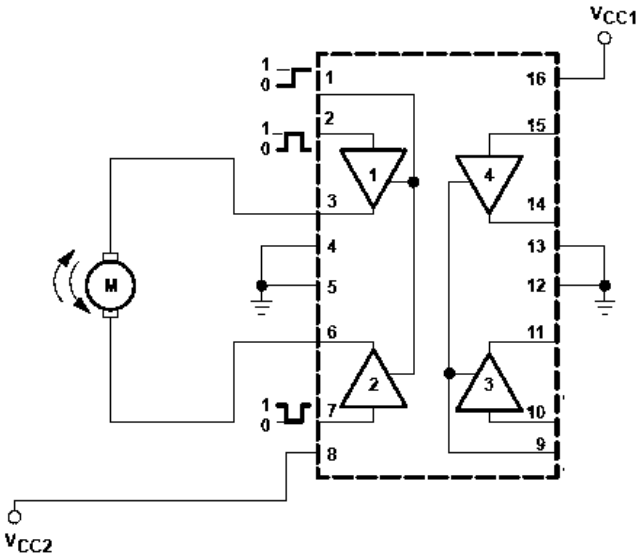


Figura 131Esquema del puente H con comparadores lógicos para su entendimiento

4.5 Base del teléfono

Como se mencionó en **4.1**, se adquirió la plataforma con medidas específicas, esto atendiendo la necesidad de espacio para componentes sobre el vehículo, sobre la plataforma se coloca un porta baterías AA, un PCB diseñado para el control de los motores, el microcontrolador y el modulo bluetooth.

Agregado a esto, se necesita una plataforma que sostenga el dispositivo Android, la base debe cumplir requisitos mínimos:

- Rigidez y al mismo tiempo evitar transferir las vibraciones.
- Capacidad de usar diversos dispositivos con la misma base.
- El área máxima debe ser 5cm cuadrados.
- Flexibilidad para orientar el dispositivo manualmente.

Las restricciones de espacio sobre la plataforma, evitan el desbalanceo y provocar un accidente.

La **Figura 132** representa un ejemplo de la base adquirida, cumple con los requisitos y es de fácil adquisición en tiendas comerciales.

Figura 132Base para teléfono celular

4.6 Baterías

Para seleccionar el nivel de voltaje necesario, hemos considerado que el microcontrolador funciona con 3.3V, el puente H requiere una alimentación externa de al menos 5V para controlar los motores, y el modulo bluetooth requiere 5V para el correcto funcionamiento. De manera que es necesario considerar una fuente de alimentación de al menos 5V.

Haciendo uso del socket que contiene el kit mencionado en **4.1**, podremos realizar la conexión de 4 baterías doble A, cada batería tiene un voltaje de 1.5V, al ser 4, tendremos 6V de salida, suficientes para alimentar el circuito aun, considerando una ligera caída de tensión.

La gran mayoría de las baterías actuales son recargables, lo cual reduce costos y ayuda al medio ambiente, pero, solamente proporcionan 1.2V, y fueron descartadas para ser utilizadas en el proyecto, el voltaje que proporciona viene en relación a los materiales con las que estas se fabrican, como podemos apreciar en la Tabla 3 Características de distintos tipos de baterías.

Tabla 3 Características de distintos tipos de baterías

Química	Nombre IEC	Nombre ANSI / NEDA	Voltaje nominal	Capacidad típica (mAh)	Capacidad típica (Wh)	Recargable
Zinc–carbono	R6	15D	1.50 V	1100	1.6	No
Alcalina	LR6	15A	1.50 V	1700-3000	2.6-3.6	Algunas
Li-FeS_2	FR6	15LF	1.55 V	3000	3.6	No
Li-ion		14500	3.6-3.7 V	700	2.6	Sí
NiCd	KR6	1.2K2	1.25 V	500-1100	0.63-1.2	Sí
NiMH	HR6	1.2H2	1.25 V	1300 -3000	1.5-3.2	Sí

Para los componentes que requieran un menor voltaje, se utilizan reguladores de tensión buscando el correcto funcionamiento del proyecto.

Capítulo 5. Pruebas y resultados

5.1 Detección de círculos

Como se menciona en **3.3** se emplea el método de Hough Circles para la detección de círculos, este método ha probado ser eficaz en el desarrollo de esta aplicación, con amplia gama de parámetros que cuenta.

Al ser un método probabilístico de detección, el primer problema es la discriminación de los elementos, de no hacerse de manera correcta, el rendimiento del dispositivo se verá afectado. El poder de procesamiento es menor con forme se tienen más elementos para analizar.

La manera de discriminación para emplearse es la corrección de tamaño concéntrico, es decir, limitar el radio de los círculos que vaya a detectar, si son más grandes o más pequeños que el tamaño deseado, ignorarlos, esto es posible gracias a los parámetros definidos en el capítulo **3.3.**

Para realizar las pruebas, es necesario un ambiente neutro, sin elementos que puedan perturbar el procesamiento digital, un fondo de pared sin elementos visibles lo hace ideal, a ese fondo, agregar elementos circulares de distintos tamaños, para comprobar la discriminación producida con base en el tamaño.

En la **Figura 133** se aprecian los elementos usados en las pruebas de discriminación de tamaños.

Figura 133 Elementos para la discriminación

De izquierda a derecha encontramos un bloque de cemento con forma cilíndrica imperfecta, una taza de aluminio con interior manchado, un cilindro de estaño, un cilindro de cable y por último un encoder de color oscuro. Cada uno con diferente tamaño, algunos con círculos intermedios dentro de ellos.

La selección de estos elementos fue realizada en base a sus tamaños, formas y colores, las imperfecciones son necesarias para poner a prueba la mayor cantidad de posibilidades en las detecciones.

Se someten los objetos a un análisis en la app creada, colocando los objetos de manera desordenada, se espera que se detecten los círculos de tamaño medio, con un radio de alrededor de **100 puntos**, como se define en el **capítulo 3.3**.

En la **Figura 134** se aprecia desde la aplicación desarrollada, como la discriminación de tamaño se realiza de una manera satisfactoria, de los cinco elementos presentes, identifica aquellos que tienen un radio dentro de los puntos establecidos, se aprecia el contorno de un color blanco y el centro de color blanco o negro según sea la posición respecto a la cuadricula central.

Figura 134 Primera detección de círculos

Continuando con las pruebas, se decide eliminar del cuadro los elementos de la taza y el encoder, dejando el bloque de cemento y los dos rollos de distinto tamaño, sumado a eso, se aleja la cámara 4cm del objetivo, logrando que los círculos disminuyan los tamaños presentados en la imagen anterior.

En la **Figura 135** se aprecia la discriminación por tamaño, nuevamente, tomando elemento de tamaño intermedio de los tres que restan, demostrando, una vez más, el correcto funcionamiento de la discriminación por tamaño.

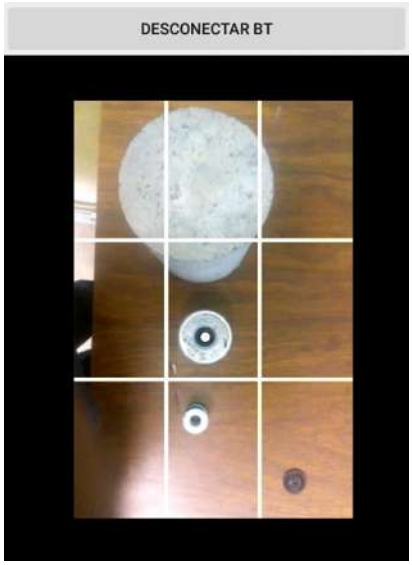


Figura 135Segunda detección, menor cantidad de elementos

De la misma manera, se toman los tres elementos más pequeños, dos rollos de distinto tamaño y un encoder, se acerca la cámara 8cm a comparación de la imagen anterior, con esto se logra aumentar la perspectiva de los círculos en la aplicación.

En la **Figura 136** se aprecia la discriminación realizada, la detección con base al tamaño logra hacer exclusión de aquellos círculos que no se encuentran dentro del rango de detección detectado, solo detecta el carrete de tamaño mayor.

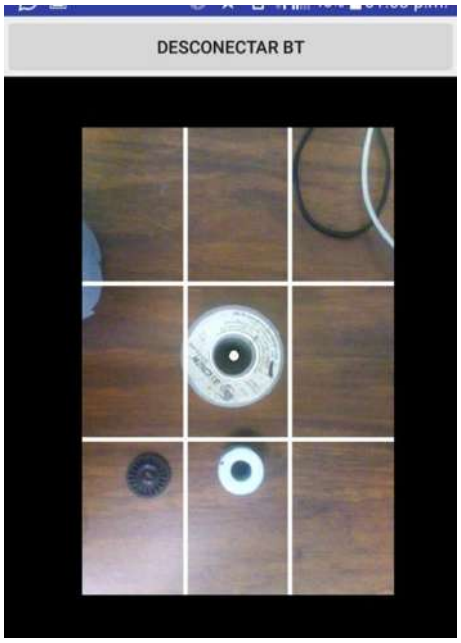


Figura 136 Detección en los elementos menores

Realizando nuevamente la prueba, se decide seleccionar los tres elementos centrales, la taza y ambos carretes.

Como se puede apreciar en la **Figura 137** la discriminación es realizada de la manera deseada, rechazando los elementos de mayor y menor tamaño, tanto la taza como el carrete mayor, se encuentran dentro del rango del tamaño permitido, teniendo éxito en la prueba.

Figura 137Detección de los círculos medianos, detecta más de uno.

De la prueba discriminatoria con base en el tamaño de los circulos, se concluye un correcto proceso de calibración y detección, en el método de detección, es necesario especificar que en caso de obtener más de un resultado, como se aprecia en la **Figura 137,** el dispositivo es incapaz de reaccionar de manera correcta, por lo que es primordial el uso correcto de objetivos, es posible ajustar los parámetros para disminuir el rango de detección del radio de los círculos, opción que puede ser perjudicial en el desarrollo del proyecto actual, el tamaño de los círculos en la percepción del dispositivo móvil depende de dos factores:

- Tamaño del objeto.
- Distancia del dispositivo al objeto.

El **tamaño del objeto** se puede especificar, como ya se dijo, en la sección de detección en la creación de la aplicación móvil, o bien, consiguiendo elementos de un tamaño determinado, por otro lado**, la percepción del dispositivo con base a la distancia** se puede ver afectada con forme el dispositivo realice sus funciones de seguimiento, es decir, al realizar un movimiento lineal se alterará la distancia entre el foco de atención y el lente del dispositivo, ocasionando una percepción errónea en la medición del radio del circulo detectado.

5.2 Niveles de iluminación

En sistemas ópticos o gráficos para realizar reconocimiento y/o control, dentro de los factores considerados como necesarios, son los niveles de iluminación, es decir, que tanto brillo y nitidez se puede tener en el sistema a analizar.

En fotografía, los niveles de iluminación son conjunto de técnicas, materiales y efectos utilizados para iluminar una escena o sujeto. La luz es el factor más importante en el arte fotográfico, ya que se vale de ella para ser creada,y su correcto control es determinante para el resultado final. [6]

Así como hay niveles de iluminación, gracias a la luz que rebota en los objetos en cuestión, existe la sombra, la contra parte de la luz, es generada en aquellas zonas donde la luz no llega o llega de manera parcial, se tiene cuando la luz rebota en un objeto y este proyecta, en sentido contrario al origen de la luminosidad, su forma proyectada a manera de sombra.

Para el proyecto desarrollado, es necesario crear un correcto contraste entre luz y sombre de los objetos, que la nitidez sea suficiente para lograr una percepción correcta, para distinguir de manera sencilla los círculos a detectar.

La siguiente prueba a la que se somete la aplicación es la detección de figuras con sobreexposición a la luz, es decir, donde la nitidez no permite distinguir las formas de manera correcta.

En la **Figura 138** se aprecian tres elementos sujetos a la prueba, la taza, la carreta mayor y el carrete pequeño, de **5.1** sabemos que solo se detectan los círculos medianos, es decir, la taza y el carrete mayor, para esta prueba es necesario sobreexponer uno de los elementos a la luz, aprovechando la luz natural que entra al lugar de experimentación.

Figura 138 Detección con poca exposición de luz

En la **Figura 138** se demuestra la identificación con base al tamaño, y se detectan los elementos que se esperaban, continuando con la experimentación, es necesario permitir una saturación luminosa.

En la **Figura 139** es apreciable que al permitir la saturación de luz de la parte inferior de la imagen, el carrete mayor es menos apreciable de lo que era en la **Figura 138,** dando como resultado, una discriminación, es decir, la luz interfiere en la percepción del elementos, este aparenta perder las características permisibles en la discriminación, resumiendo esto, el elemento no es detectable y, lo que en otras condiciones es detectado como un círculo, ahora no es reconocible como tal.

Figura 139 Sobre exposición a la luz

Comprobando lo anterior, se intenta disminuir la luminosidad presente en la fotografía, basta con hacer un movimiento en el espacio de los objetos, de esa manera la luz entrante no afecta directamente el entorno del carrete mayor, logrando que la percepción cambie.

En la **Figura 140** se aprecia el movimiento de los objetos y la clara disminución de la intensidad de la luz, aunque sigue existiendo una exposición mayor, es posible detectar figuras gracias a las zonas de penumbra que permiten tener mayor nitidez en los contornos.

Figura 140 Menor exposición a la luz, mayor apreciación de los contornos

Realizando más pruebas el resultado es muy similar, se decide usar un elemento que contiene dos semicírculos en su interior, el carrete pequeño y un elemento cuadrangular de superficie reflectante para alterar la canalización de la luz en los objetos y el lente de la cámara.

Partiendo de los resultados anteriores, se hace un acercamiento del dispositivo, logrando que los objetos aparenten un mayor tamaño ante la perspectiva de la cámara, de esa manera los elementos presentes tienen los puntos de radio suficientes para ser detectados por la aplicación.

En la **Figura 141** se aprecian varios aspectos del funcionamiento de la aplicación creada, en primera instancia es de percatarse que la luminosidad es baja, suficiente para delinear los contornos de los objetos y sus interiores, acto seguido apreciamos que detecta el carrete pequeño, el tamaño a esa cercanía de la cámara logra que este dentro del espectro de detección programado, por último, se aprecia que las figuras semicirculares son ignoradas por la aplicación, debido a que no son círculos perfectos, es decir, hace una correcta discriminación de los elementos.

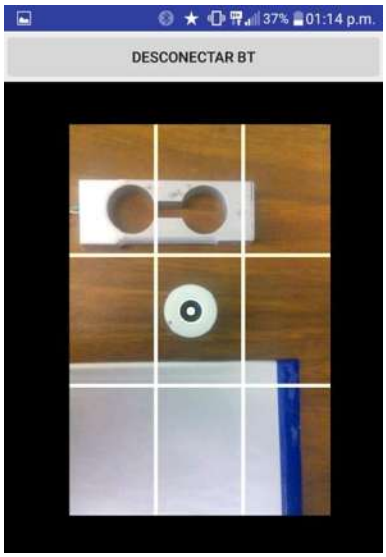


Figura 141Baja luminosidad, cercanía de objetos y discriminación de formas imperfectas

Tomando como referencia la detección realizada del elemento circular, se altera la saturación luminosa entrante en la imagen, esto con motivo de identificar la cantidad en la cual ya no es permisible la correcta diferenciación de los contornos presentes, de esa manera el carrete pequeño no es detectable.

En la **Figura 142** se aprecia el resultado del supuesto planteado en el párrafo anterior, se logra que un objeto bien definido sea no detectable al alterar la saturación de la luz, el lente pierde detalle de los objetos definidos, de esa manera las sombras y contornos son no detectables.

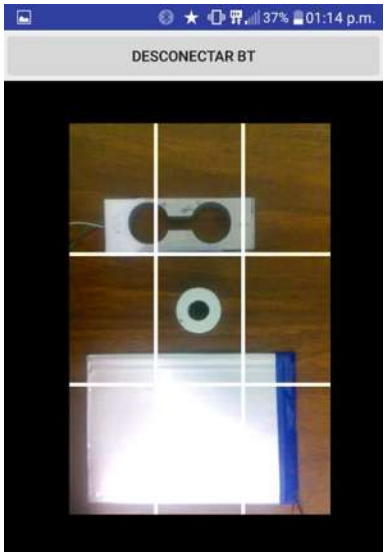


Figura 142 Saturación de luz a un objeto que previamente era definido

A manera de conclusión, se demuestra la importancia de la luz que incide en nuestro proyecto, es necesario tener una luminosidad controlada, la discriminación de tamaño y forma se realiza de manera correcta, sin embargo, si la iluminación no es la adecuada, podemos tener un resultado erróneo.

Haciendo la medición de la luz entrante en la cámara del dispositivo, por medio de sensores de luz y microcontroladores, se estima que entre **500 y 650 lumens** es la cantidad ideal para la visualización de los contornos, la sobre exposición de luz es medida en valores superiores a los **700 lumens.**

Existen distintos métodos para el control de la iluminación, como lo son fuentes artificiales de emisión, de esta manera controlamos la iluminación brindada por lámparas en espacios oscuros o donde la luz natural sea despreciable, otra manera es controlar el lugar y horario de realizar los experimentos, de esa manera la iluminación será la misma en un mismo punto con condiciones muy similares, lo cual es impreciso y poco práctico, a su vez, se agrega al software desarrollado, la capacidad de autoenfoque, con la cual puede alterar la manera en que el diafragma de la cámara reacciona a distintas cantidades de luz, esta solución es practica pero no suficiente debido al tiempo que toma en realizar el autoenfoque.

Se determina el uso de ambientes controlados para la correcta función.

5.3 Discriminación del entorno

Sumado a **5.2,** es necesario el uso de un óptimo entorno, en **3.3**, se hace mención de los algoritmos necesarios para detectar círculos, en el cual se especifica la manera de detección, analiza el entorno en búsqueda de la mayor cantidad de círculos con características específicas.

El costo computacional para la detección se ve incrementado por la cantidad de objetos a analizar, es decir, si tenemos un entorno con menor cantidad de objetos el procesamiento se realizará a mayor velocidad, traduciéndose en mayores respuestas el vehículo.

La discriminación del entorno se realiza en dos ejes:

- Método de Hough Circles.
- Entornos físicos.

La **discriminación con el método de Hough Circles** se realiza estableciendo valores en el procesamiento de imagen, dichos valores indican la cantidad de círculos que serán detectados a la par, para este proyecto, la limitación de la detección se reducirá a dos elementos circulares al mismo tiempo.

La **discriminación con entornos sólidos,** hacemos referencia al entorno plano que se ha comentado previamente en este capítulo, al no tener más objetos que analizar los elementos que se desean introducir en el ambiente controlado será los únicos factores que influyan.

Se recomienda el uso de colores neutros para el entorno y que hagan contraste con el elemento circular, esto ayudando a la mejor detección, como se muestra en la **Figura** 143 se presentan dos elementos de igual tamaño pero distinto color, el carrete pequeño de color blanco y el encoder de color negro, es de notar como solo se detecta el carrete, debido a su color que contraste con el entorno oscuro que los rodea, mientras que el encoder es no detectable ante la lente de la cámara.

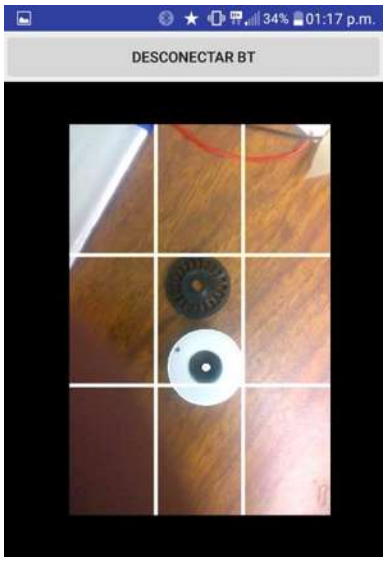


Figura 143 Objeto no detectable por el entorno de color oscuro

Con esta prueba se demuestra la importancia del contraste para facilitar la detección del contorno de los objetos, es necesario cambiar el entorno.

En la **Figura 144** se muestra el cambio de entorno, sigue siendo un entorno sólido, de color claro, permitiendo que el contorno del elemento encoder de color oscuro, sea visible, la luz y la nitidez permiten la correcta detección del objeto, sin embargo, no es definitivo, puesto que podría verse afectado este resultado por la iluminación del entorno.

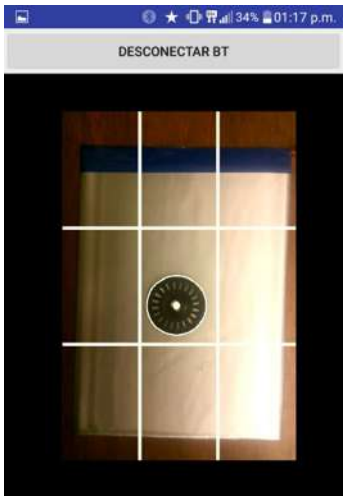


Figura 144 Objeto en entorno solido que genera contraste

Pensando en limitar la correcta identificación de los objetos, se toma una muestra que se presenta en la **Figura 145** donde se aprecia dos objetos visiblemente idénticos, pero en entornos distintos, uno en un entorno con contraste y el otro en un entorno oscuro, demostrando que solo con el entorno que representa un contraste es que funciona la detección de manera adecuada.

Figura 145 Objetos idénticos en entornos distintos

Se comprueba que en el entorno correcto que presente un alto contraste, la detección es realizada de manera correcta y estable, logrando, cumplir el objetivo.

En la **Figura 146** Se aprecian dos encoders circulares de color negro, en un entorno liso contrastando el color que originalmente tienen, de esa manera se hace la detección de ambos elementos de manera correcta, comprobando el buen funcionamiento de la aplicación.

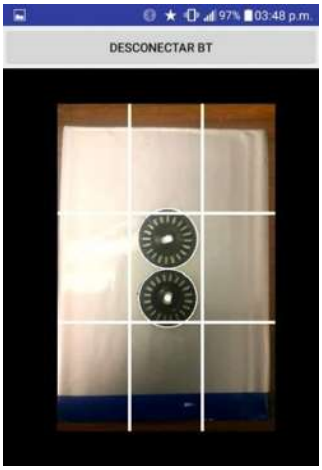


Figura 146 Detección doble en un entorno contrastante

Con las pruebas anteriores, se concluye la necesidad de un entorno neutro, es decir, sin elementos indeseables que puedan causar conflicto en el procesamiento de imágenes de la aplicación desarrollada, se recomienda el uso de un cercado con paredes sólidas y lisas en el cual el vehículo pueda desplazarse libremente sin usar procesos computacionales mayores a los necesarios.

Una vez contemplados los factores de discriminación del entorno, se crea una cerca de metal con paredes lisas de tonos claros, con finalidad de algo contraste, las dimensiones suficientes para la libre conducción del vehículo son de: 34.5cm de largo por 28cm de alto en cara pared.

La construcción cuenta con tres paredes como se muestra en la **Figura 147**.

Figura 147Construcción de cerca para contención del vehículo

En la **Figura 148** se aprecia el prototipo de la cerca construida para la contención vehículo y discriminación del entorno, en ella se aprecia el correcto espacio entre el vehículo y las paredes, que le permiten una movilidad completa.

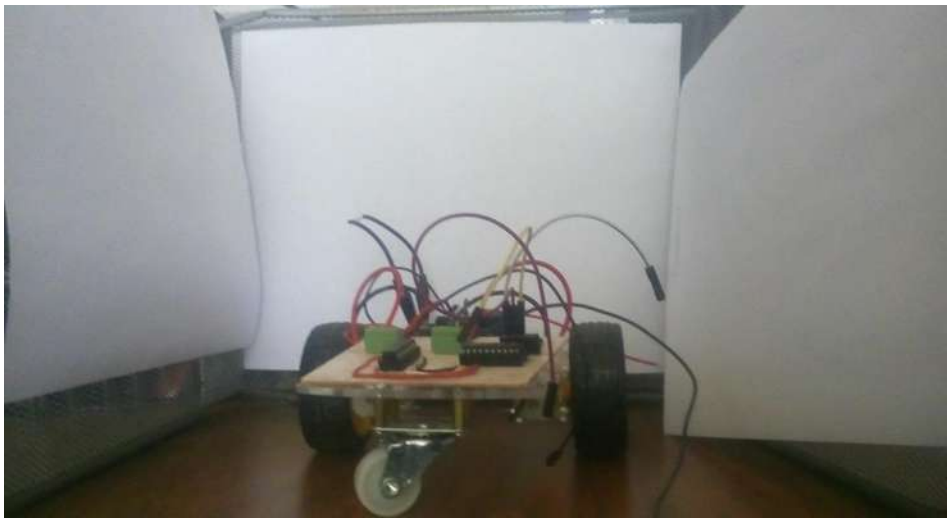


Figura 148Primera presentación del vehículo ante la barrera, comprobando las correctas dimensiones.

La decisión de tres paredes únicamente, obedece a las necesidades de manejo del vehículo y movilidad del mismo.

5.4 Movimiento de la plataforma.

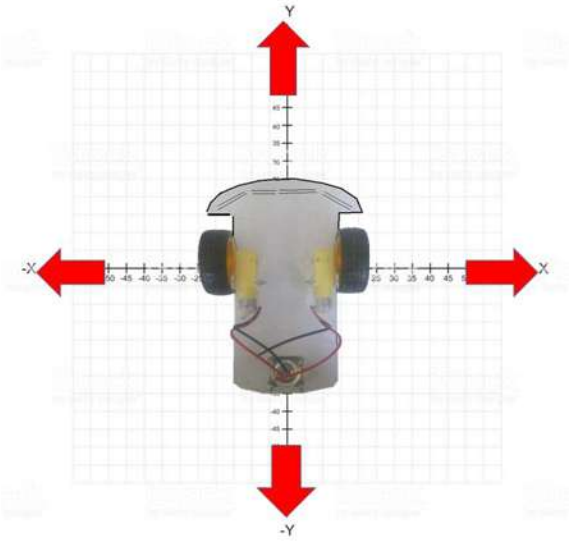


Figura 149 Ejes de movimiento

Como se vio en **3.4.2** la **Tabla 2** Indica la movilidad del vehículo, la cual se complementa con la **Figura 149,** el vehículo es capaz de tener movimiento libre en dos ejes.

El movimiento de la plataforma va en función de la detección de los patrones, los motores que se encuentran en las laterales del vehículo permiten el movimiento en dos ejes, la velocidad no es cuestión de control en este proyecto, por lo tanto, la velocidad es regulada por el voltaje de alimentación de los motores, el cual es de **4.5Volts** obtenido de tres baterías doble AA.

El movimiento en dos ejes cubre de manera superficial las 9 divisiones de la pantalla que se tienen, la llanta loca cumple plenamente su función de estabilizar la plataforma.

El libre movimiento de la plataforma atiende a la capacidad de la aplicación creada de poder centrar el círculo detectado.

5.5 Seguimiento del patrón.

El seguimiento del patrón se realiza con éxito a lo esperado en **3.5.3,** el dispositivo Android localiza el círculo detectado en cualquiera de los 9 sectores en los que fue segmentada la pantalla.

Se configura la aplicación para comenzar a realizar la detección de círculos una vez que se haya establecido con éxito la conexión con el modulo Bluetooth, por lo que al llegar a esta etapa, se concluye que se realizó la exitosa conexión.

Una vez iniciada la detección de círculos, envía de manera correcta los códigos descritos en el **3.5.3,** iniciando el movimiento correcto, tomando como referencia la parte trasera del vehículo, la cual es donde se encuentra la llanta loca.

El vehículo se encuentra inmóvil si el círculo no aparece en la pantalla o si se encuentra en el segmento número 5.

La detección en los sectores superiores 1,2 o 3 se realiza de manera exitosa, sin embargo, se presentan problemas con el control, si el vehículo se mueve en reversa, no garantiza que el círculo se centre, 9 de 10 pruebas realizadas con el circulo en esta posición, la detección era cancelada, el círculo salía del rango de la pantalla.

El defecto anterior es reproducido de igual manera si el círculo es detectado en la parte inferior, en los segmentos 7, 8 o 9, si el vehículo avanza o retrocede para continuar con la detección la detección se ve interrumpida y desconoce el objeto como parte de un círculo o, sale por completo de la imagen.

Con base a la documentación y experimentación del proyecto, se llega a la conclusión de que el defecto es producido por la falta del dominio en un tercer eje que proporcione la facilidad de tener control sobre la altura a la cual la cámara apunta.

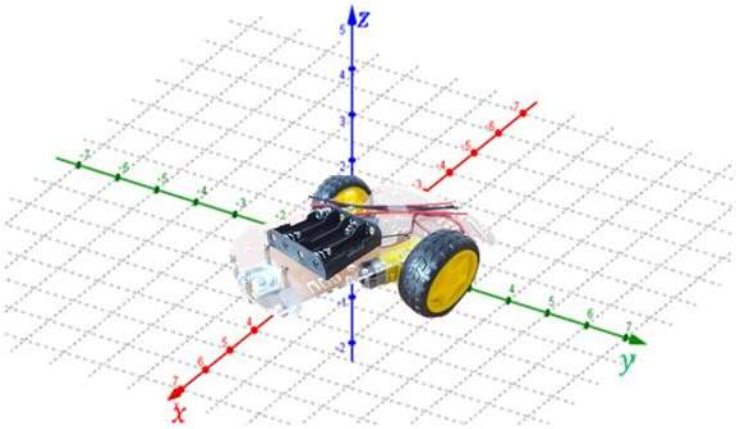


Figura 1500 Vehículo móvil en un plano de 3 ejes

Como se aprecia en la **Figura 150** el vehículo solo es capaz de ir en dos ejes, el tercer eje (el eje z) es alcanzable solo para movimientos que eleven el vehículo.

Dado el acotamiento de este proyecto de tesis, se decide no hacer uso del tercer eje de elevación, se decide reducir las opciones de movimiento del vehículo, los movimientos serán giros en sentido horario o anti horario.

Los nuevos códigos de envió de datos de la aplicación creada serán:

1.- Vuelta a la derecha.
2.- Alto.
3.- Vuelta a la izquierda.

Por lo que se decide disminuir los segmentos de detección como se aprecia en la **Figura 1511.**

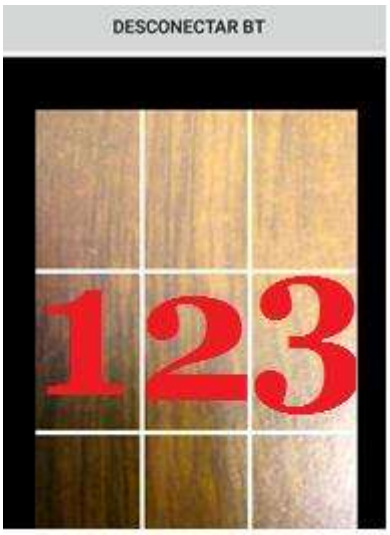


Figura 151 Nuevos segmentos de detección

Si el circulo se encuentra en cualquiera parte del segmento 1, el vehículo girara a la izquierda, si por el contrario se encuentra en cualquiera parte del segmento 3 el vehículo girará a la derecha, si se encuentra en el segmento 2, el vehículo se detendrá.

El círculo puede ser detectado en cualquier parte de la pantalla pero solo tiene tres opciones de movimiento, si se detecta en los segmentos antes mencionados.

5.6 Velocidad de seguimiento

Como se vio en **4.3** la velocidad de los motores sin carga es de aproximadamente 240PM a 6V, con el peso de los componentes se ve reducido a aproximadamente 235RPM, suficiente para lograr una gran velocidad, sumado a la acción de dar los giros provocando que las llantas giren en sentidos opuestos, el giro es más cerrado y logra una mayor velocidad, dado que no tiene que arrastrar una llanta inmóvil.

Debido a las restricciones físicas en el cálculo de procesos del dispositivo Android, la velocidad de procesamiento digital de video se ve reducido a 6 **FPS (**Frames Per Second/ Cuadros por segundo**),** velocidad considerable en ambientes controlados, cuanto menor sea el número de objetos a analizar,mayor será la velocidad de procesamiento, logrando procesar hasta **12 FPS** en condiciones óptimas, motivo por el cual en **5.1** se redujo la detección a un máximo de dos objetos de manera simultánea.

Al realizar las pruebas de manera separada, el vehículo tiene una gran velocidad de movimiento y giro, mientras que el dispositivo móvil tiene una gran oportunidad de detección en ambientes controlados, la reacción entre la detección y el movimiento del vehículo es casi de manera inmediata, limitado por la velocidad de transferencia y recepción del Bluetooth.

Cuando se monta el dispositivo móvil en el vehículo y comienza la detección, la velocidad de los motores supera la velocidad de procesamiento, por lo que si el dispositivo detecta un objeto de lado izquierdo, comienza el giro en sentido anti horario, y la gran velocidad hace que el dispositivo deje de detectar el dispositivo y el giro es lo suficiente grande para perder el objeto a detectar lejos del foco.

Por el motivo anterior, se toma la decisión de disminuir el voltaje de alimentación de los motores, el voltaje mínimo de funcionamiento es de 3V, el cual logra una velocidad de 120RPM sin carga. Al colocarle la carga, los motores son incapaces de iniciar su movimiento, por lo que se toma la decisión de incrementar el voltaje a **4V**, capaz de tener una velocidad relativa de **160RPM** sin carga.

Con el voltaje de 4V con los componentes puestos, los motores son capaces de iniciar su movimiento y girar con mayor lentitud, sin embargo, aún es demasiado rápido comparado con la velocidad de detección y procesamiento, por lo que se opta por cambiar la estructura del movimiento de los motores, haciendo que al dar los giros solo un motor realice el movimiento y el otro sea una carga inmóvil que ralentice un tanto la velocidad.

Tabla 4 Nuevo sentido de giro de los motores

Sentido del movimiento	Dirección motor izquierdo	Dirección motor derecho
Giro a la derecha		
Giro a la izquierda		
Alto		

Posterior a las modificaciones, la velocidad de giro es aún mayor que la velocidad de procesamiento de video del dispositivo, razón por la cual, el dispositivo no detecta con precisión los círculos, hasta que el vehículo este detenido, el vehículo completo el fin del movimiento cuando el objeto a detectar esta fuera del foco de dispositivo.

Las limitaciones físicas del dispositivo para realizar las pruebas, es apenas suficiente para utilizar la aplicación creada, se espera que al usar un dispositivo distinto y dedicado a este experimento, los resultados mejorarán en gran medida.

Capítulo 6. Conclusiones

Se han cumplido apenas de manera satisfactoria los objetivos planteados al inicio de este documento en **1.2,** en el cual se establece como objetivo general la creación de una aplicación para dispositivos móviles que cuenten con sistema operativo Android, dicha aplicación debía ser capaz de detectar figuras geométricas especificas por medio del procesamiento digital de señales, utilizando los recursos del dispositivo tales como la cámara, acceso a la memoria y conexión por medio de bluetooth.

De manera general se demuestra como en nuestros dispositivos de uso cotidiano tenemos poderosos procesadores capaces de realizar complejos procesos de cálculo, y con el paso del tiempo los dispositivos son aún más potentes para realizar operaciones cada vez más complejas.

El dispositivo usado es el LG-Q10, con características:

-Procesador 1.3GHz.
-1GB de RAM.
-Cámara de 8MP.
-Android 5.0.

El uso de la biblioteca de **OpenCV** fue la piedra angular para lograr el procesamiento de imágenes, su integración en el entorno de Android Studio fue una labor que permitió el éxito del proyecto, con dicha biblioteca se hizo posible además, lograr las especificaciones de los objetos a detectar, tanto el uso de bibliotecas internas para la detección de círculos, como la discriminación de tamaños, separación y cantidades de detección simultaneas.

Satisfactoriamente se concluye que es posible realizar todas las operaciones previstas, que es posible lograr una conexión de bluetooth estable con el módulo HC-05 y a su vez comunicarlo con un microcontrolador de Texas Instruments **msp430g2553.**

La creación de PCB de forma artesanal y no industrial es suficiente para el desarrollo de proyectos complejos, que en este proyecto de tesis es enfocado en la creación de la aplicación, no sería posible su desarrollo sin la capacidad de probarlo con componentes físicos.

A su vez, la mejora del proyecto es viable con la adaptación de un componente que permita el manejo del dispositivo móvil en un tercer eje, logrando que enfoque objetos localizados en la parte inferior y superior desde su punto de vista.

Sumado a lo anterior cabe destacar que fue necesario el uso de una batería extra para el uso del módulo bluetooth y el microcontrolador, al usar la misma fuente de poder para todo el vehículo (parte de potencia y parte de control) ocasionaba fallas, debido a la energía requerida en el momento de arranque de los motores, causaba un pico de bajada de tensión,

la cual no permitía el correcto uso del módulo bluetooth, y se optó por utilizar fuentes separadas.

El uso de un dispositivo Android dedicado para este proyecto es recomendable, con la finalidad de disponer de todos sus recursos y características de construcción.

Capítulo 7. Trabajos futuros

Considerando la dificultad de elevar el vehículo y que continúe con el movimiento, se propone la elevación del dispositivo Android, la elevación sobre el eje z seria realizada sobre el vehículo y no complicaría el movimiento en los otros dos ejes.

A manera de propuesta, se menciona el uso de un servomotor **SG99** representado en la **Figura 152** el cual tiene una fuerza de hasta 1Kg [26] suficiente para elevar un dispositivo Android de tamaño medio,

Figura 152 Servomotor SG99

De esa manera sería posible alcanzar los segmentos superiores e inferiores de la pantalla de detección.

Las características ideales para un mejor procesamiento de imágenes dentro de un dispositivo Android, son:

-Dispositivo dedicado (recomendado pero no necesario).
-Procesador 2GHz.
-2GB de RAM.
-Cámara de 8MP.
-Android 7.0 (recomendado, solo es necesario que sea superior a 4.0.2),

A manera de mejora, se plantea la implementación de un contenedor anti vibraciones que soporte al dispositivo Android, el cual elimine el ruido causado por las vibraciones del movimiento del vehículo, dichas vibraciones provocan dificultades en la detección de los objetos.

A manera más importante se espera como trabajo futuro, se está trabajando en la mejora de la aplicación, la cual permita localizar otras figuras geométricas, algunos colores, en la **Figura 153** se muestra el prototipo del trabajo realizado para el futuro.

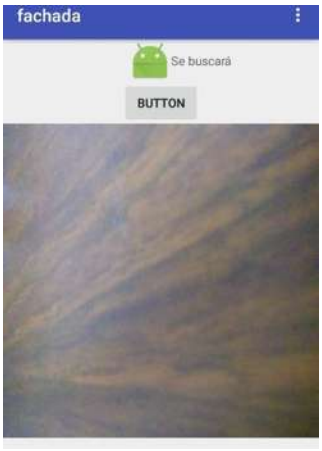


Figura 153 Fachada de la nueva aplicación

En esta mejora de la aplicación, el usuario tiene un menú despegable en el cual seleccione un color en la sección de "Paleta", si buscar una figura geométrica en la sección de "*Figura"* o si cambiar las opciones de conexión bluetooth en la opción de "*Settings".*

En la **Figura 154** se aprecia el menú despegable del usuario, localizado en la parte superior derecha de la pantalla.

Figura 154 Menu despegable

En la **Figura 155** se nota que al seleccionar la opción de "*Paleta"* aparece un recuadro con una paleta de colores.

Figura 155 Paleta de colores

Por último, en la **Figura 156** se demuestra la lista despegable que aparece al seleccionar la opción "*Figura"* en la que se observan las 4 opciones de figuras geométricas que se espera se puedan detectar en trabajos futuros.

Figura 156 Lista de figuras geométricas disponibles

GLOSARIO

A

AgfaColor
Procedimiento alemán para reproducir la imagen en colores naturales. Consta de un solo negativo con tres emulsiones, cada una sensible a uno de los tres colores fundamentales. 47

Algoritmos
Secuencia de pasos a seguir. 101

Android
es un sistema operativo móvil desarrollado por Google, basado en el Kernel de Linux y otros software de código abierto. VI

B

Baudios
Unidad de medida de frecuencia. 101

C

Callback
Metodo de devolución, en el cual se tienen dos funciones, la función al ejecutarse la función "B" hace un llamado a la función "A". 28

Canvas
Del inglés, que significa "Lienzo" es un método en Android en el cual permite tener herramientas de dibujo y creación. 85

Celuloide
Material plástico, muy flexible empleado en la industria fotográfica y cinematográfica para la fabricación de películas 47

Ch

Chasis
Soporte que sostiene los componentes de un vehiculo y ayuda a que mantenga su forma. 72

C

Comunicación serial
Protocolo de comunicación de dos vías, una de transferencia y otra de recepción. 94

D

Daguerrotipo
fue el primer procedimiento fotográfico anunciado y difundido oficialmente en el año 1839. Fue desarrollado y perfeccionado por Louis Daguerre 46

E

Encoders
Dispositivos para medir la cantidad de vueltas que da un motor. 72

F

Fotografía
Representación visual de alguna cosa o situación, puede ser imaginaria (visiones o fantasias) o física 46

Fragment
Representa un comportamiento o una parte de la interfaz de usuario en una Activity,parte con la que interactua el usuario. 82

G

GMAIL
Servicio de correo electronico proporcionado por Google. 82

I

Imagen
Del latín imago, es una representación visual, que manifiesta la apariencia visual de un objeto real o imaginario. 8

ISM
Rango de frecuencia destinado al uso Cientifico, Medico e Industrial, usos sin licencia no deben interferir con los especificos. 97

K

Kodachrome
Kodak introdujo la película analógica Kodachrome, la cual permitió que por primera vez se pudiese realizar fotografía en color usando un carrete. 47

L

Layouts
Hace referencia al esquema que será utilizado y cómo estan distribuidos los elementos y formas dentro de un diseño 84

lumens
(Símbolo
lm) es la unidad del Sistema Internacional de Medidas para medir el flujo luminoso, una medida de la potencia luminosa emitida por la fuente. 120

M

Maketado
Previa creación, sujeta a errores, simil a bosquejo. 79

MDF
Por sus siglas en inglés Medium Density Fibreboard, que significa que es un panel de fibra de densidad media, material de construcción. 71

Modulo
Dispositivo Electrónico con entradas y salidas capaces de adaptarse a las plataformas de desarrollo. 94

Motorreductor
Motor con engranes que dan fuerza a la velocidad del motor. 107

O

onClick
En Android, función que permite que los objetos seleccionados realicen alguna acción que se les haya programado. 33

P

PCB
Del inglés "Printed Circuit Board", que es una placa de circuito impreso, generalmente con pistas de cobre, y de diversos materiales aislantes como cerámica o plástico. 102

PNG
Formato de imagen. 84

PWM
Del inglés "Pulse Weidth Modulator" que significa modulador de ancho de pulsos, el cual modifica el ciclo tr trabajo de una señal. 67

R

RAM
Del Inglés Random Access Memory, que significa una memoria de acceso aleatorio. 64

RPM
Revoluciones Por Minuto. 128

S

Simultánea
Al mismo tiempo. 128

Smartphone
Telefono movil inteligente, 33

SNAP
Es un protocolo recogido por la norma IEEE 802 que permite direccionar diferentes protocolos utilizando usando un servicio de puntos de acceso público. 100

T

TCP
Protocolo de control de transmisión. Este protocolo se encarga de crear “conexiones” entre sí para que se cree un flujo de datos. Este proceso garantiza que los datos sean entregados en destino sin errores y en el mismo orden en el que salieron. 100

Timers
Del inglés, que significa temporizador. 101

Tracking
Del inglés, que significa seguimiento. 35

U

UART
Protocolo de interpretación de datos para la comunicación serial. 94

V

View
Del inglés, que significa "vista", representa la pantalla que es visible para la interacción con el usuario. 30

W

Watchdog
Del inglés, que significa "Perro Guardian" el cual hace referencia a un algoritmo para mantener funcionando el dispositivo de fábrica 101

Whatsapp

Widget

microaplicaciones que se despliegan en la pantalla de tu computador, en una página web o en tu dispositivo móvil 13

Referencias

[1] DEPARTAMENTO DE ELECTRÓNICA, AUTOMÁTICA , Introducción al Procesamiento digital de Señales, Madrid, España.

[2] Google Inc., "Android Developer," Android, 04 10 2017. [Online]. Available: https://developer.android.com/about/index.html. [Accessed 04 10 2017].

[3] L. Contreras, «Blog de Historia de la Informatica,» Universidad Politecnica de Valencia, 14 Diciembre 2012. [En línea]. Available: http://histinf.blogs.upv.es/2012/12/14/android/. [Último acceso: 8 Octubre 2017].

[4] Ontecnia Media Networks, «Malavida,» Ontecnia Media Networks, S.L., [En línea]. Available: http://www.malavida.com/es/analisis/la-historia-de-android. [Último acceso: 9 Octubre 2017].

[5] Google Inc., «Android,» Google, 2014. [En línea]. Available: https://www.android.com/intl/es-419_mx/history. [Último acceso: 14 Octubre 2017].

[6] P. E. Tippens, «Luz e Iluminación,» de *Fisica, Conceptos y Aplicaciones*, China, McGraw Hill, 2007, pp. 642-657.

[7] J. M. López Sancho, E. Moreno Gómez y M. J. Gómez Díaz, «Museo Virtual de la Ciencia del CSIC,» 2005. [En línea]. Available: http://museovirtual.csic.es/salas/luz/luz34.htm. [Último acceso: 09 Noviembre 2017].

[8] E. Hernadez, «Facultad de Arte y Diseño de la UNAM,» 2007. [En línea]. Available: http://blogs.fad.unam.mx/asignatura/elva_hernandez/wp-content/uploads/2013/08/teoria-de-los-colores-luz-y-tipos-de-luz-e.pdf. [Último acceso: 10 Noviembre 2017].

[9] A. H. Navarro, «ahenav,» 2015. [En línea]. Available: https://ahenav.com/2014/03/28/que-es-la-luz-que-es-el-color-el-espectro-visible/. [Último acceso: 10 Noviembre 2017].

[10] PCE Ibérica S.L., «PCE Iberica,» 30 Septiembre 2014. [En línea]. Available: http://www.pce-iberica.es/instrumentos-de-medida/sistemas/sensores.htm. [Último acceso: 11 Noviembre 2017].

[11] Red Gráfica Latinoamérica, «Red Grafica,» 21 Enero 2010. [En línea]. Available: http://redgrafica.com/Que-es-un-pixel. [Último acceso: 20 Noviembre 2017].

[12] Informatica, «Informatica Hoy,» Abril 2007. [En línea]. Available: https://www.informatica-hoy.com.ar/aprender-informatica/Megapixel-Texel-y-Voxel.php. [Último acceso: 16 Noviembre 2017].

[13] FotoNostra, «FotoNostra,» junio 2014. [En línea]. Available: http://www.fotonostra.com/grafico/teoriacolor.htm. [Último acceso: 22 enero 2018].

[14] Definista, «Concepto Definicion,» junio 2011. [En línea]. Available: http://conceptodefinicion.de/imagen/. [Último acceso: 18 febrero 2018].

[15] Maison Nicéphore Niépce, «Museo Maison Nicéphore Niépce,» 2003. [En línea]. Available: http://www.photo-museum.org/es/historia-fotografia/. [Último acceso: 21 febrero 2018].

[16] R. Asomoza y Palacio , P. Cabrera Muñoz, M. Cruz Vázquez, I. Hernández Pérez, B. G. Nuñez Fernández y L. Velazquez Orozco, «Biblioteca digital,» Instituto Latinoamericano de la Comunicación Educativa ILCE, 2010. [En línea]. Available: http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec_9.htm. [Último acceso: 28 febrero 2018].

[17] R. Wainscheker, J. M. Massa y P. Tristan, *Procesamiento digital de imagenes,* Buenos Aires: Universidad Nacional del Centro de la Prov. de Bs. As., 2010.

[18] Universitas Miguel Hernández, «INGENIERÍA DE SISTEMAS Y AUTOMÁTICA - UMH,» 30 mayo 2017. [En línea]. Available: http://isa.umh.es/asignaturas/crss/temasvision/t06.pdf. [Último acceso: 28 febrero 2018].

[19] J. Wales, «Wikipedia,» wikimedia, 15 enero 2001. [En línea]. Available: https://en.wikipedia.org/wiki/OpenCV. [Último acceso: 10 abril 2018].

[20] Intel, «OpenCV,» Intel, marzo 1999. [En línea]. Available: https://opencv.org/. [Último acceso: 10 abril 2018].

Lista de tablas

Lista de ecuaciones

Lista de Figuras