



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE INGENIERÍA**

CARRERA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

**“Trabajo de grado previo a la obtención del Título de Ingeniero en  
Electrónica y Telecomunicaciones”**

TRABAJO DE GRADUACIÓN

**Título del proyecto**

DESARROLLO DE UN SISTEMA EMBEBIDO PARA AYUDA A LA  
NATACIÓN DE PERSONAS NO VIDENTES MEDIANTE EL  
RECONOCIMIENTO DE IMÁGENES

**Autor:**

Jaime Alejandro Medina Sánchez

**Director:**

Ing. Giovanni Cuzco

**Riobamba – Ecuador**

**AÑO 2016**

Los miembros de Tribunal de Graduación del proyecto de investigación con el título: **DESARROLLO DE UN SISTEMA EMBEBIDO PARA AYUDA A LA NATACIÓN DE PERSONAS NO VIDENTES MEDIANTE EL RECONOCIMIENTO DE IMÁGENES** presentado por: Jaime Alejandro Medina Sánchez y dirigido por: Ingeniero Giovanni Cuzco.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite el presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

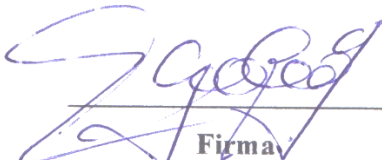
Para constancia de lo expuesto firman:

Ing. Paulina Vélez  
**Presidente del Tribunal**



Firma

Ing. Giovanni Cuzco  
**Director del Proyecto**



Firma

Ing. Fabián Gunsha  
**Miembro de Tribunal**

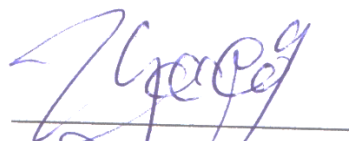


Firma

## CERTIFICACIÓN DEL TUTOR

Certifico que el presente trabajo de investigación previo a la obtención del grado de Ingeniero en ELECTRONICA Y TELECOMUNICACIONES. Con el tema: **“DESARROLLO DE UN SISTEMA EMBEBIDO PARA AYUDA A LA NATACIÓN DE PERSONAS NO VIDENTES MEDIANTE EL RECONOCIMIENTO DE IMÁGENES”** ha sido elaborado por el estudiante **Jaime Medina**, el mismo que ha sido revisado y analizado en un cien por ciento con el asesoramiento permanente de mi persona en calidad de Tutor por lo que se encuentran aptos para su presentación y defensa respectiva.

Es todo cuanto puedo informar en honor de la verdad



---

Ing. Giovanni Cuzco

C.I. 1802840718

## AUTORÍA DE INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a **Jaime Medina e Ingeniero Giovanni Cuzco**; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



---

Jaime Alejandro Medina Sánchez

C.I 060497023-6

## **AGRADECIMIENTO**

*A mi familia quienes a lo largo de toda mi vida me han apoyado mi formación académica, creyeron en mí en todo momento y no dudaron de mis habilidades. A mis profesores a quienes les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza. A mis amigos con quienes he compartido buenos y malos momentos*

***Jaime Medina***

## **DEDICATORIA**

*Dedico este proyecto de tesis a mi familia, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento. Depositando su entera confianza en cada reto que se me presentaba sin dudar ni un solo momento en mi inteligencia y capacidad. Es por ellos que soy lo que soy ahora.*

***Jaime Medina***

## ÍNDICE GENERAL

<b>ÍNDICE DE FIGURAS</b> .....	<b>xi</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>xiii</b>
<b>RESUMEN</b> .....	<b>xiv</b>
<b>ABSTRACT</b> .....	¡Error! Marcador no definido.
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO I</b> .....	<b>3</b>
1. FUNDAMENTACIÓN TEÓRICA.....	3
1.1. Antecedentes .....	3
1.2. Sistemas embebidos .....	5
1.2.1. Componentes de un sistema embebido .....	5
1.3. Sistemas operativos embebidos .....	8
1.4. Algunos miniordenadores .....	10
1.5. Comparación entre Arduino, Raspberry Pi y Beaglebone Black.....	15
1.5.1. Arduino, ventajas y desventajas .....	15
1.5.2. Raspberry Pi, ventajas y desventajas.....	16
1.5.3. Beaglebone Black, ventajas y desventajas .....	16
1.5.4. Tabla resumen de las principales características de las tarjetas ..	17
1.6. Procesamiento de imágenes .....	17
1.6.1. Imagen digital.....	18
1.6.2. Aplicación de la Transformada de Fourier al procesamiento de imágenes.....	18
1.6.2.1. Transformada de Fourier continua .....	18
1.6.2.2. Transformada de Fourier Discreta.....	19
1.6.2.3. Filtros empleando Fourier .....	21

1.6.3.	Procesamiento de imágenes con OpenCV .....	23
1.6.3.1.	Filtrado de una imagen en OpenCV .....	23
1.7.	Comunicación inalámbrica .....	26
1.8.	Entorno de programación en Beaglebone Black.....	26
1.8.1.	Introducción a la beaglebone.....	26
1.8.2.	Actualización e instalación de librerías .....	30
1.9.	Creación de un proyecto en Python en la tarjeta Beaglebone Black ..	31
<b>CAPÍTULO II .....</b>		<b>33</b>
2.	METODOLOGÍA .....	33
2.1.	Tipo de estudio.....	33
2.1.1.	Descriptivo-Correlacional .....	33
2.2.	Métodos, Técnicas e Instrumentos.....	33
2.2.1.	Métodos .....	33
2.2.1.1.	Analítico .....	33
2.2.2.	Técnicas .....	33
2.2.2.1.	Observación.....	33
2.2.3.	Instrumentos .....	33
2.3.	Población y muestra.....	34
2.3.1.	Población .....	34
2.3.2.	Muestra .....	34
2.4.	Hipótesis .....	34
2.5.	Operacionalización de variables .....	34
2.6.	Procedimientos.....	34
2.7.	Procedimiento y análisis .....	35
2.7.1.	Inconvenientes que presenta un nadador no vidente .....	35



2.7.1.1. Diseño y selección de los componentes para el desarrollo del sistema de apoyo .....	35
2.7.2. Implementación del sistema embebido para ayuda a la natación	36
2.7.3. Desarrollo del software embebido para el reconocimiento de imágenes.....	37
<b>CAPÍTULO III.....</b>	<b>45</b>
3. RESULTADOS.....	45
3.1. Resultados de la prueba de los filtros a emplear para procesar las imágenes.....	45
3.2. Resultados de la prueba del sistema embebido para la ayuda a la natación .....	46
<b>CAPÍTULO IV .....</b>	<b>49</b>
4. DISCUSIÓN .....	49
<b>CAPÍTULO V.....</b>	<b>50</b>
5. Conclusiones y Recomendaciones .....	50
5.1. Conclusiones.....	50
5.2. Recomendaciones .....	51
<b>CAPÍTULO VI.....</b>	<b>52</b>
6. PROPUESTA.....	52
6.1. Título de la propuesta .....	52
6.2. Introducción .....	52
6.3. Objetivos.....	53
6.3.1. Objetivo general .....	53
6.3.2. Objetivos Específicos .....	53
6.4. Fundamentación Científico – Técnico .....	53
6.5. Descripción de la propuesta.....	53

6.6.	Diseño organizacional.....	54
6.7.	Monitoreo y Evaluación de la Propuesta .....	54
<b>7.</b>	<b>BIBLIOGRAFÍA .....</b>	<b>55</b>
<b>8.</b>	<b>ANEXOS Y APÉNDICES.....</b>	<b>57</b>

## ÍNDICE DE FIGURAS

<i>Figura 1.- Sistema Embebido Fuente.- J. Echavarría – Designing Eletronics</i>	5
<i>Figura 2.-Procesador Intel Core i7 Fuente.- Mishell Yagual – Conociendo tu Pc</i>	6
<i>Figura 3.- Memoria ROM</i>	6
<i>Figura 4.- Periféricos de entrada y salida de un pc.</i>	7
<i>Figura 5.- Sistemas operativos</i>	8
<i>Figura 6.- Diferentes modelos de arduino</i>	11
<i>Figura 7.- Placa de Raspberry PI</i>	12
<i>Figura 8.- Placa Beaglebone Black</i>	13
<i>Figura 9.- Placa de Nanode</i>	14
<i>Figura 10.- Placa Pandaboard</i>	14
<i>Figura 11.- Placa Waspnote</i>	15
<i>Figura 12.- a) Función impulso, b) Transformada de Fourier de una función impulso</i>	19
<i>Figura 13.-Discretizacion de la función <math>f(x)</math></i>	20
<i>Figura 14.- Aplicación de un filtro pasa bajos</i>	22
<i>Figura 15.- Aplicación de un filtro pasa alto</i>	22
<i>Figura 16.- Aplicación de un filtro pasa banda</i>	23
<i>Figura 17.- Suavizado de una imagen mediante Avering</i>	24
<i>Figura 18.- Imagen aplicado el Laplaciano para encontrar los bordes</i>	25
<i>Figura 19.- Obtención de bordes mediante canny</i>	26
<i>Figura 20.- Conexión de la beaglebone a la pc</i>	27
<i>Figura 21.- Instalación de drivers de la beaglebone</i>	27
<i>Figura 22.- Comprobación de los drivers instalados sea correctos</i>	28
<i>Figura 23.- Configuración del programa PuTTY</i>	29
<i>Figura 24.- Conexión de la beaglebone mediante SSH</i>	29
<i>Figura 25.-Configuración de VNC en la beaglebone</i>	30
<i>Figura 26.- Escritorio de la beaglebone con el sistema operativo Debían</i>	30
<i>Figura 27.- Instalación de librerías en beaglebone usando Debian</i>	31
<i>Figura 28.- Abrir LXTerminal</i>	31
<i>Figura 29.- Creación de un nuevo programa python en beaglebone</i>	32

<i>Figura 30.- Programa en python</i>	32
<i>Figura 31.- Diagrama de bloques del sistema a desarrollar</i>	36
<i>Figura 32.- Esquema del sistema de ayuda</i>	37
<i>Figura 33.- Diagrama del proceso ejecutado en la beaglebone</i>	38
<i>Figura 34.- Plantilla para dibujar la ruta</i>	41
<i>Figura 35.- Dibujo de la ruta</i>	41
<i>Figura 36.- Resultado de la configuración de la tarjeta Intel Edison</i>	42
<i>Figura 37.- Cálculo de la distancia entre dos puntos</i>	43
<i>Figura 38.- Cálculo del ángulo entre dos rectas</i>	44
<i>Figura 39.- Aplicación de un filtro pasa bajo empleando Fourier</i>	45
<i>Figura 40.- Aplicación de un filtro empleando las funciones de OpenCV</i>	45
<i>Figura 41.- Detección de la pelota</i>	46
<i>Figura 42.- Seguimiento de la pelota</i>	47
<i>Figura 43.- Seguimiento de la pelota</i>	47
<i>Figura 44.- Prueba del software con el objeto</i>	48
<i>Figura 45.- Prueba del sistema en el usuario</i>	48
<i>Figura 46.- Esquema Organizacional</i>	54
<i>Figura 47.- Pruebas con obstáculos</i>	57
<i>Figura 48.- Pruebas en piscina vacía</i>	57

## ÍNDICE DE TABLAS

<i>Tabla 1.- Comparación entre Arduino uno, Raspberry pi (Modelo B) y Beaglebone black.....</i>	<i>17</i>
<i>Tabla 2.- Operacionalización de las variables.....</i>	<i>34</i>
<i>Tabla 3.- Listado de materiales para el sistema.....</i>	<i>36</i>

## RESUMEN

Este proyecto de investigación presenta el desarrollo de un sistema embebido para la ayuda a la natación de personas no videntes mediante el reconocimiento de imágenes. El sistema está desarrollado en el lenguaje Python y emplea las librerías de visión por computadora OpenCV para el procesamiento de las imágenes. La implementación de este sistema se realizó sobre la mini pc Beaglebone Black con el sistema operativo Linux versión de kernel Debian.

El sistema embebido adquiere la información mediante una cámara web conectada a la beaglebone mediante el puerto USB. La beaglebone es la encargada de la parte computacional del sistema embebido, procesa la información empleando las librerías de OpenCV. Detecta objetos y proporciona las coordenadas para saber su ubicación, de esta forma puede guiar al usuario a través de la piscina mediante instrucciones de audio emitidos por el sistema de alertas desarrollado. Proporciona las indicaciones necesarias a través de un sistema de alertas con el fin de evitar que el nadador colisione con las demás personas u objetos que se encuentren en la piscina.

La información será transmitida al usuario empleando una comunicación inalámbrica. Esta información le llega a través de auriculares que están conectados a un dispositivo Android que ejecuta una aplicación para recibir los datos, la voz le guiará para realizar su respectivo ejercicio, se busca que el no vidente pueda orientarse en la piscina. Se espera que haya una familiarización con el medio más factible al recibir las instrucciones en tiempo real.



UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERIA  
CENTRO DE IDIOMAS INSTITUCIONAL



Lic. Geovanny Armas

11 de Agosto del 2016

### ABSTRACT

This research project shows the development of an embedded system for helping blind people to swim through the recognition of images. The system is developed in the *Python* programming language and uses the OpenCV computer vision libraries for the processing of images. The implementation of this system was carried out on the Beaglebone Black mini PC with the Linux operating system, kernel Debian version. The embedded system obtains the information through a web camera connected to the Beaglebone via USB port. The Beaglebone is in charge of the computing part of the embedded system, it processes the information using the OpenCV libraries. It detects objects and provides coordinates in order to know its location, in this way it can guide the user through the swimming pool by using audio commands emitted by the alert system developed. It provides the necessary instructions through audio commands with the purpose of avoiding the swimmer to collide against other people or objects in the swimming pool. The information will be transmitted to the user by using a wireless communication, this information comes through headphones, the voice will guide him/her to perform his/her corresponding exercise, the objective is that the blind person can guide himself/herself inside the swimming pool. A familiarization with the most feasible way to receive instructions in real time is expected.



## INTRODUCCIÓN

La actividad física para las personas es muy importante ya que ayuda a aliviar el estrés, es una forma natural de combatir y reducir los síntomas de la ansiedad y la depresión, reduce la presión arterial y el riesgo de padecer enfermedades cardiovasculares, en cambio para las personas con alguna discapacidad es vital, esto es debido a que necesitan potenciar al máximo sus habilidades para así compensar sus deficiencias y poder desenvolverse de una manera óptima. Sin embargo algunos individuos no realizan actividades recreativas debido a su condición física, como es el caso de las personas no videntes que a pesar de contar con sus extremidades intactas se niegan hacer actividades deportivas por temor que el desarrollo de las actividades los pueda afectar.

La natación es un deporte que aporta grandes beneficios en cualquier edad y condición física. Para el nadador con deficiencia visual le aportará grandes beneficios como: aprovechar el tiempo libre de una forma sana y divertida, evitar el sedentarismo, entre otras. Por lo tanto se busca la manera de incentivar a estas personas a realizar este deporte mediante el empleo de un sistema de ayuda de natación para personas no videntes mediante el reconocimiento de imágenes desarrollado en un sistema embebido que le proporcionara al usuario la capacidad de orientarse en el agua para que pueda nadar de forma libre evitando colisionarse con las demás personas u objetos que se encuentren en la misma.

El sistema trabaja con el procesamiento de la imagen para capturar ciertas características del entorno, para ello se colocará una cámara en un punto estratégico de la piscina, con el fin de tener una vista completa de la misma. La información recibida por medio de la cámara irá conectada al sistema embebido desarrollado, este sistema estará conformado por una SBC<sup>1</sup>, esta tarjeta es la encargada del sistema computacional del sistema embebido. La SBC procesa las imágenes

---

<sup>1</sup> Single Board Computer es una computadora completa en un solo circuito



recibidas a través del software embebido implementado, y proporciona las instrucciones necesarias a través de un sistema de alertas diseñado con el fin de evitar que el nadador colisione con las demás personas u objetos que se encuentren en la piscina. La información será transmitida al usuario empleando una comunicación inalámbrica. El sistema de alertas conformado por instrucciones de voz, otorgara indicaciones al usuario. Esta información le llega a través de auriculares, la voz le guía para realizar su respectivo ejercicio. Se espera que haya una familiarización con el medio más factible al recibir las instrucciones en tiempo real.

Con la implementación de este proyecto se busca incentivar a las personas con discapacidad visual a realizar la natación, proporcionando una manera de reinsertarse en la sociedad, haciéndolo sentir mejor consigo mismo, elevando su autoestima y mejorando su estilo de vida.

# CAPÍTULO I

## 1. FUNDAMENTACIÓN TEÓRICA

### 1.1. Antecedentes

El reconocimiento de imágenes es el eje fundamental para el desarrollo de varios sistemas como son: detección de patrones, reconocimiento de objetos, visión artificial entre otros. En el presente proyecto de investigación se empleara el reconocimiento de imágenes como eje central, por tal motivo se vio conveniente revisar estudios anteriores con el fin de obtener fundamentos previos, algunos de estos estudios se detallan a continuación:

En su proyecto los ingenieros Efraín Arévalo, Arturo Zúñiga, Juan Villegas y Carlos Avilés presentan la Implementación de reconocimiento de objetos por color y forma en un robot móvil. Su trabajo está enfocado en otorgar al robot la capacidad de analizar un objeto con base a su color y forma y desplazarse para buscar otro objeto idéntico mediante el procesamiento de imágenes en tiempo real. Para la parte del hardware se emplea una tarjeta Raspberry Pi<sup>2</sup>, una cámara web, una tarjeta de red inalámbrica y el sistema de movimiento mientras que para la parte del software la integran el sistema operativo instalado en la tarjeta SD del Raspberry Pi, la librería de visión artificial por computadora OpenCV<sup>3</sup>, servidor Web Lighttpd<sup>4</sup> el cual es parte del sistema de monitoreo y el código fuente del programa que representa el sistema de procesamiento de imágenes. Con esto se obtiene un sistema de operación autónomo que toma decisiones del movimiento del robot con base al análisis y procesamiento de las imágenes capturadas, con un procesamiento fluido

---

<sup>2</sup> Ordenador de placa simple (SBC)

<sup>3</sup> Biblioteca open source para C/C++ para procesamiento de imágenes y visión computarizada.

<sup>4</sup> Servidor web agil y liviano

de 5 cuadros de imagen por segundo, logrando una detección en tiempo real. (Arevalo Vazquez, Zuñiga Lopez, Villegas Cortez, & Aviles Cruz, 2015)

Los ingenieros José Salgado, Luis Vásquez y Mauricio Vidal presentan en su proyecto Diseño e implementación de algoritmos para el procesamiento de imágenes en sistemas embebidos. Es un sistema que permite el reconocimiento de colores, detección de bordes y seguimiento de color por medio de un algoritmo de visión artificial, para lograr este objetivo emplean un sistema embebido de bajo costo y fácil adquisición en específico emplean la FPGA SPARTAN 3AN<sup>5</sup> y la RASBERRY PI, para la adquisición de imágenes utilizan dos cámaras que serán las encargadas de capturar las imágenes luego se aplican técnicas de umbralización y segmentación, se toma en cuenta el gradiente y el operador sobel. Para el desarrollo de los algoritmos de procesamiento de imágenes se utiliza el lenguaje VHDL<sup>6</sup> para la FPGA y PYTHON<sup>7</sup> para la RASBERRY PI. (Salgado Patron, Vasquez Diaz, & Vidal Solano, 2013)

Los ingenieros Marcelo Raponi y Rodolfo Bonnin presentan un Dispositivo de rehabilitación visual basado en sistemas embebidos del tipo ARM<sup>8</sup>, este prototipo permite la adquisición de señales de video en tiempo real, mediante una interfaz diseñada específicamente para este caso, es posible seleccionar diferentes procesamientos digitales que se aplicara a la señal entrante como es el empleo de filtros espaciales, modificar el brillo y contraste, hacer zoom, invertir colores entre otras. Este dispositivo está diseñado en la placa Beagleboard<sup>9</sup>, para el desarrollo de la plataforma se emplea el IDE Qt-creator y librerías open-source (Qt, Highgui y OpenCV). El patrón visual final es enviado a un módulo de visualización montado en las lentes de unos anteojos, con esto se busca mejorar diversas tareas de la vida cotidiana que realizan los pacientes con baja visión. (Raponi & Bonnin, 2011)

---

<sup>5</sup> Dispositivo programable que contiene bloques de lógica

<sup>6</sup> Lenguaje de descripción de hardware de circuitos integrados de muy alta velocidad

<sup>7</sup> Lenguaje de programación de alto nivel

<sup>8</sup> Microprocesadores de pequeño tamaño

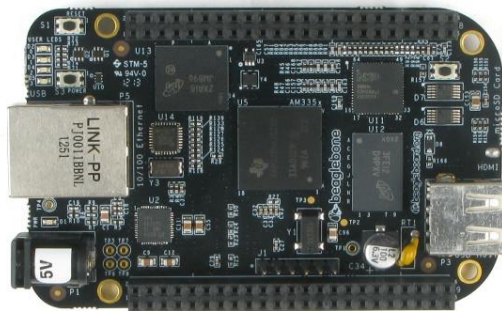
<sup>9</sup> Placa de computadora de hardware libre

## 1.2. Sistemas embebidos

Los sistemas embebidos son dispositivos electrónicos diseñados para cumplir con una función específica, es decir son sistemas dedicados hechos para resolver un problema específico, a diferencia de las computadoras que son diseñadas para cumplir con diferentes funciones.

Los sistemas embebidos o también conocidos como sistemas empotrados, tienen todos sus componentes en la placa base (tarjeta de video, audio, modem, etc.). Los sistemas embebidos se pueden programar mediante lenguaje ensamblador haciéndolo directamente sobre el microprocesador o micro controlador que tiene incorporado, o a través de compiladores específicos se puede usar los lenguajes C, C++, java. (Solis Gonzalez, 2016)

Los sistemas embebidos están presentes en nuestra vida cotidiana, en automóviles, celulares, electrodomésticos de uso diario como son refrigeradores, hornos microondas, etc.



*Figura 1.- Sistema Embebido*

*Fuente.- J. Echavarría – Designing Eletronics*

### 1.2.1. Componentes de un sistema embebido

Los sistemas embebidos están compuestos por:

**Procesador.-** Es un circuito integrado encargado del proceso lógico de los datos, el rendimiento del sistema dependerá de su velocidad.



*Figura 2.-Procesador Intel Core i7*

*Fuente.- Mishell Yagual – Conociendo tu Pc*

**Memoria.-** Dispositivo electrónico encargado de almacenar la información necesaria para el correcto funcionamiento del sistema. Existen dos tipos de memorias:

**RAM,** el almacenamiento es temporal debido a que necesitan corriente eléctrica para mantener la información.

**ROM,** no necesita de corriente eléctrica para almacenar la información, contiene la información necesaria para iniciar el computador.



*Figura 3.- Memoria ROM*

*Fuente.- Mishell Yagual – Conociendo tu Pc*

**Periféricos.-** Son dispositivos que sirven de enlace a la computadora con el exterior. Existen periféricos de entrada, de salida, o periféricos de entrada/salida.

**Periféricos de entrada.-** Son mediante los cuales introducimos información al pc. Ejemplo: Tenemos teclado, mouse, micrófono, entre otros.

**Periféricos de salida.-** Nos permiten obtener información proveniente del pc. Ejemplo: Tenemos monitor, altavoces, impresora, entre otros

**Periféricos de entrada/salida.-** Realizan ambas acciones.



*Figura 4.- Periféricos de entrada y salida de un pc.*

*Fuente.- R. Castañeda – Software y Hardware*

**Sistema operativo.-** Es el software necesario para el funcionamiento del ordenador, maneja los periféricos y sirve de intermediario entre el ordenador y el usuario. Los sistemas operativos más empleados son: Windows CE, Linux, Android.



*Figura 5.- Sistemas operativos*

*Fuente.- Compulab*

### **1.3. Sistemas operativos embebidos**

En un sistema embebido es necesario la instalación de un sistema operativo el cual será el encargado de administrar los recursos de manera eficiente, es el responsable de la comunicación hardware-software, para la ejecución de las tareas para las que fue programado, y así maximizar su desempeño. (Caballero Paz, 2013)

Entre los sistemas operativos embebidos tenemos:

- ✓ Android
- ✓ Linux
- ✓ Windows CE
- ✓ Windows Embedded Automotive
- ✓ OSEK
- ✓ OS/2 (eComStation)
- ✓ QNX
- ✓ VxWorks
- ✓ ThreadX
- ✓ FreeBSD

**Android.-** Este sistema operativo está diseñado para ser usado en teléfonos inteligentes, tabletas, televisores, dispositivos de vehículos, etc.

**Linux.-** Es un sistema multiplataforma desarrollado para las arquitecturas x86, x86-64 y ARM.

**Windows CE.-** Sistema operativo en desuso, era usado en cajeros automáticos y sistemas de navegación para los vehículos. Fue reemplazado por Windows XP Embedded y Windows Embedded Standard.

**Windows Embedded Automotive.-** Este sistema operativo lo podemos encontrar en vehículos de marcas como FIAT, Nissan y Ford.

**OSEK.-** Este sistema es usado en vehículos de marcas como BMW, Chrysler, Opel y Renault.

**OS/2 (eComStation).-** Este Sistema operativo se encuentra en desuso, era usado en cajeros automáticos, fue muy popular en los 90.

**QNX.-** Se usó en videoconsolas como Dreamcast, está disponible para diferentes arquitecturas, presenta estabilidad ante fallo de periféricos.

**VxWorks.-** Es diferente a los demás sistemas operativos, pertenece a la rama RTOS<sup>10</sup>, fue usado para operar vehículos espaciales. Sistema operativo en tiempo real

**ThreadX.-** Este sistema operativo se usa en impresoras, cámaras digitales, módems y sondas espaciales. Sistema operativo en tiempo real.

---

<sup>10</sup> Real time operating systems



**FreeBSD.-** Es un sistema operativo completo, lo encontramos en televisores, routers y sistemas de seguridad. Es la base sobre la que se apoya CellOS, el sistema que rige PlayStation 3 y su Cross Media Bar (XMB).

#### 1.4. Algunos miniordenadores

Miniordenadores, son computadores de placa única (SBC<sup>11</sup>), que tienen integrados un microprocesador, una memoria y periféricos de entrada y salida. Son computadores completos.

Entre estos miniordenadores tenemos los siguientes:

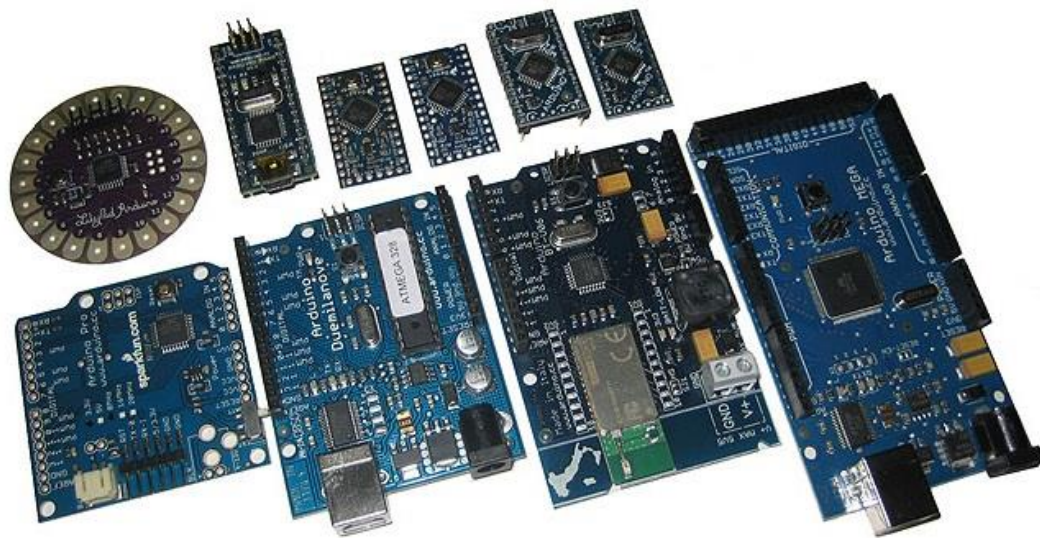
**Arduino.-** Es una plataforma electrónica de fácil empleo, orientada a público no experto como son artistas, estudiantes de robótica o computación. Es una placa electrónica no solo de creación sino también empleada para el aprendizaje en el diseño de sistemas automáticos. Arduino es una placa pre ensamblada, que tiene varios elementos incluidos que facilitan su utilización. Contiene varias entradas y salidas tanto analógicas como digitales.

Arduino tiene un micro controlador de la familia ATMEL, el cual es el encargado de realizar los procesos lógicos y matemáticos, además de controlar a los dispositivos conectados a él. El lenguaje empleado en la programación de Arduino se llama Wiring, está basado en la plataforma Processing, principalmente en el lenguaje de programación C/C++, lenguaje que se ha vuelto muy popular a tal grado de ser el más preferido para enseñar programación a alumnos de nivel superior que estudian computación y robótica, gracias que es muy fácil de aprender y brinda soporte para cualquier necesidad de computación. (arduinodhtics, 2016)

Arduino tiene una gran variedad de modelos, como podemos ver en la siguiente figura.

---

<sup>11</sup> Single Board Computer, en ingles



*Figura 6.- Diferentes modelos de arduino*

*Fuente.- arduinodhtics*

El más reciente y novedoso modelo de arduino es el modelo Intel de Edison, incluye un procesador Intel® Atom™ de 500MHz dual-core dual-threaded CPU and an Intel® Quark™ 100MHz microcontroller. Posee WiFi y Bluetooth 4.0 LE integrado, tiene soporte para yocto, Python, nodejs y wólfram.

**Raspberry pi.-** Es un miniordenador que puede ser utilizado como un ordenador de escritorio, se puede hacer hojas de cálculo, procesar texto, juegos, reproducir video. Se creó con el objetivo de fomentar la enseñanza de las ciencias de computación en los niños. Esta placa es del tamaño de una tarjeta de crédito, tiene entradas USB, puertos de Ethernet y salida HDMI, para conectar al miniordenador con otros dispositivos como teclados, mouses, pantallas.

Tiene todos los componentes esenciales requeridos para correr un sistema operativo. Usa el controlador Broadcom, que es un Soc<sup>12</sup>. Este Soc tiene un poderoso procesador ARM11 que corre a 700 MHz. Esta minicomputadora no trae display, pero puede ser usado con un display HDTV o los estándares de TV NTSC

---

<sup>12</sup> System on Chip

o PAL. Tiene un puerto Ethernet que permite conectarlo a una red. Se pueden cargar sistemas operativos desde Mac, Windows y Linux. Su capacidad para correr Linux y la accesibilidad a la plataforma mediante una red LAN lo hace una opción perfecta para pequeños servidores web dedicados. (Casco, 2014)

Con Raspberry se puede desarrollar proyectos más complejos que con arduino, emplea lenguajes de programación de alto nivel como Python, C++ y java.

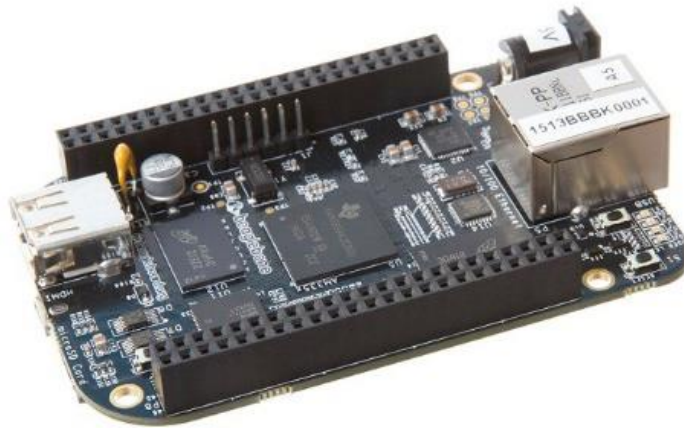


*Figura 7.- Placa de Raspberry PI*

*Fuente.- A. Sola - Pucomí*

**Beagleboard.-** Es una pequeña computadora de hardware libre de bajo consumo, la cual puede ejecutar tareas similares a un Pc de escritorio, se puede realizar hojas de cálculos, reproducir videos, procesar texto, juegos. Cuenta con varias I/O digitales, entradas analógicas, salidas PWM, puerto Ethernet, soporta comunicación I2C y SPI. Se la puede programar en varios lenguajes, como son: C/C++, java, Python, Ruby, PHP, Octave, Node.js, entre otros.

Funciona en base a un procesador ARM, para ser más específico Sitara de Texas Instrument, posee un CPU de 2 nucleos, Cortex A8 y se ejecuta instrucciones a 1GHZ con sus 512Mb de RAM y un procesador gráfico para 3D. El sistema operativo que trae por defecto es Debian Linux, pero se puede instalar otros sistemas como son Android o Ubuntu.



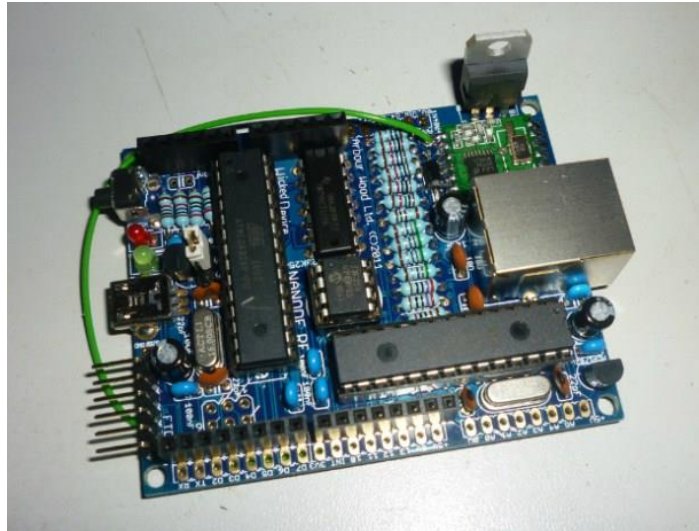
*Figura 8.- Placa Beaglebone Black*

*Fuente.- G.Coley- elinux*

**Nanode.-** Es una placa electrónica de código abierto que permite la creación de proyectos emocionantes debido a la posibilidad de conectarse a internet, e interactuar con aplicaciones basadas en la nube. Nanode permite conectarse a internet mediante un API o un navegador de internet.

Nanode puede ser empleado como servidor de páginas web sencillas, y permitir al usuario interactuar con su hardware a través de una interfaz de navegador. Nanode también puede ser empleado para la obtención de datos ambientales como la temperatura, el clima o la calidad de aire con la ayuda sensores, y estos datos almacenar en un servidor basado en la nube como Cosm.

Nanode posee iguales conectores que la placa de arduino, permitiendo así la posibilidad de expandirse mediante el empleo de placas de expansión compatibles con este. (Nanode, 2016)



*Figura 9.- Placa de Nanode*

*Fuente.- Nanode*

**Pandaboard.-** Es un mini ordenador orientado al procesamiento digital de señales, se puede emplear diferentes sistemas operativos como: Android, Ubuntu, MeeGeo, entre otros.



*Figura 10.- Placa Pandaboard*

*Fuente.- J. Blanco – La mirada del golem*

**Libelium Waspote.-** Es empleado para la creación de redes inalámbricas de sensores. Trabaja con diferentes protocolos como son: LoRaWAN, Sigfox, ZigBee, WiFi.



*Figura 11.- Placa Wasmote*

*Fuente.- blogthinkbig*

## **1.5. Comparación entre Arduino, Raspberry Pi y Beaglebone Black**

Estas tres plataformas son las más empleadas a la hora de desarrollar cualquier proyecto electrónico, debido a que cada una de ellas posee fortalezas y debilidades haciéndolas unas mejores que otras.

A continuación se resumirá algunas ventajas y desventajas de usar estas plataformas:

### **1.5.1. Arduino, ventajas y desventajas**

Ventajas

- ✓ Posee el programador incorporado en la plataforma
- ✓ Es de bajo consumo, ideal para usos de larga duración
- ✓ Se encuentra amplia documentación de apoyo
- ✓ Flexibilidad para distintos tipos de interfaces
- ✓ Es de bajo costo, es decir es relativamente barato
- ✓ Al usar librerías de programación es más sencilla y fácil de aprender
- ✓ Es de código y hardware abierto por lo que se puede usar libremente
- ✓ Posee diferentes placas que se ajustan a las necesidades del diseñador

- ✓ Se puede expandir mediante el uso de shields, diseñados para aplicaciones específicas

#### Desventajas

- ✓ No es capaz de manejar diferentes procesos al mismo tiempo
- ✓ Debido a que en la programación se emplea librerías, se produce un retraso en la ejecución de las instrucciones
- ✓ Debido a que la plataforma viene ensamblada, quita flexibilidad en el proyecto debido al tamaño de la placa

### **1.5.2. Raspberry Pi, ventajas y desventajas**

#### Ventajas

- ✓ Es una mini computadora, su procesador de 700 MHZ
- ✓ Posee un puerto Ethernet para una fácil conexión a internet
- ✓ Posee el sistema operativo en una tarjeta SD, por lo que se puede cambiar fácilmente solo cambiando la tarjeta SD
- ✓ La placa permite su expansión, debido a que posee compatibilidad con los shields de arduino
- ✓ Posee dos puertos USB

#### Desventajas

- ✓ No posee suficientes interfaces para la conexión de sensores externos o botones

### **1.5.3. Beaglebone Black, ventajas y desventajas**

#### Ventajas

- ✓ Posee una memoria flash y un sistema operativo ya instalado
- ✓ Posee un gran número de entradas y salidas (GPIO)
- ✓ Tiene un procesador de 1 GHZ

#### Desventajas

- ✓ Posee un solo puerto USB
- ✓ Es complicado encontrar tutoriales o material de apoyo para el desarrollo de proyectos
- ✓ No posee codificación de video

#### 1.5.4. Tabla resumen de las principales características de las tarjetas

	<b>Arduino uno</b>	<b>Raspberry Pi (Model B)</b>	<b>Beaglebone Black</b>
<b>Procesador</b>	ATMega 328	ARM 11	AM 33x
<b>Velocidad</b>	16 MHz	700 MHz	1 GHz
<b>RAM</b>	2 Kb	512 Mb	512 Mb
<b>USB</b>	No tiene	2	1
<b>Audio</b>	No tiene	HDMI analógico	HDMI analógico
<b>Video</b>	No tiene	HDMI analógico	HDMI mini
<b>Ethernet</b>	No tiene	10/100	10/100
<b>Entradas/Salidas</b>	14 digitales 6 analógicos	8 GPIO	69 GPIO LCD GMPC MMC1, MMC2 7 entradas analógicas 4 temporizadores Puerto serial CAN 0
<b>Sistema operativo</b>	No tiene	Linux	Android, Linux, Windows, etc
<b>Programa</b>	Arduino IDE	Linux, IDLE, Scratch, Eclipse, QEMU	Python, Scratch, Linux, Eclipse Android ADK

*Tabla 1.- Comparación entre Arduino uno, Raspberry pi (Modelo B) y Beaglebone black*

*Fuente.- (Casco, 2014)*

#### 1.6. Procesamiento de imágenes

El procesamiento de imágenes ha ido evolucionando con el avance de la tecnología, permitiendo así la creación de algoritmos de procesamiento más potentes. El



reconocimiento de imágenes es un proceso complejo que necesita una serie de pasos sucesivos que transforman los datos a información que el computador pueda entender. Para el procesamiento de la imagen se emplea un sin número de recursos como puede ser matlab, aplicaciones desarrolladas en java, eclipse, opencv, etc.

El procesamiento digital de imágenes es un conjunto de técnicas aplicadas a imágenes digitales con el fin de obtener información útil que permita al ordenador reconocer una imagen.

### **1.6.1. Imagen digital**

Una imagen digital es una representación de una imagen a través de una matriz numérica en la mayoría de los casos unos y ceros (bits), que contienen información de color e intensidad de la imagen. El elemento más pequeño de una imagen es el pixel, es cada una de las casillas o celdas en que se descomponen una imagen digital.

### **1.6.2. Aplicación de la Transformada de Fourier al procesamiento de imágenes**

La Transformada de Fourier es una herramienta matemática importantísima para el procesamiento de señales digitales, permite descomponer una imagen en sus componentes de senos y cosenos, es decir transforma una imagen del dominio espacial al dominio de la frecuencia.

#### **1.6.2.1. Transformada de Fourier continúa**

Sea  $f(x)$  una función continua, la transformada de Fourier está dada por:

$$F[f(x)] = F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux} dx \quad \text{Ec. (1)}$$

Donde  $j$  es la raíz cuadrada de (-1) y  $u$  la variable de frecuencia. Dada  $F(u)$ , se puede encontrar  $f(x)$ , empleando la transformada inversa de Fourier

$$F^{-1}(u) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux} du \quad \text{Ec. (2)}$$

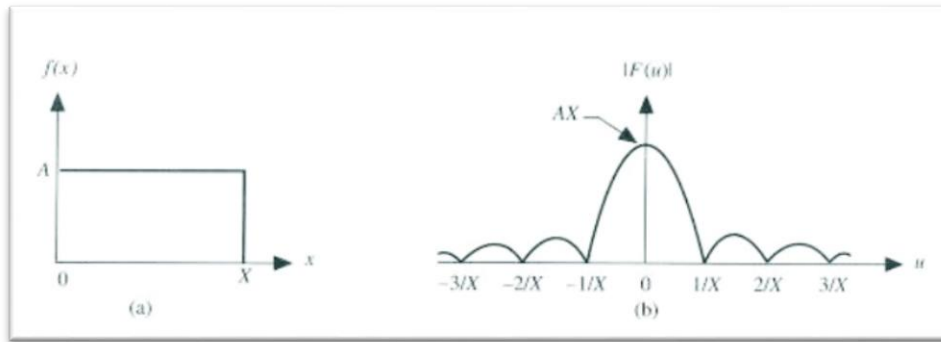


Figura 12.- a) Función impulso, b) Transformada de Fourier de una función impulso

Fuente.- El autor

Se puede expresar la transformada de Fourier en forma exponencial:

$$F(u) = |F(u)|e^{j\phi(u)} \quad \text{Ec. (3)}$$

En donde:

$$|F(u)| = \sqrt{R^2(u) + I^2(u)} \quad \text{Ec. (4)}$$

$$\phi(u) = \arctan \left[ \frac{I(u)}{R(u)} \right] \quad \text{Ec. (5)}$$

La función modulo se llama *Espectro de Fourier* de  $f(x)$  y  $\phi(u)$  es el ángulo de fase.

Si, estas ecuaciones son aplicadas a funciones bidimensionales del tipo  $f(x, y)$ , al aplicar la transformada de Fourier tenemos:

$$F(u, v) = \iint_{-\infty}^{\infty} f(x, y)e^{-j2\pi(ux+vy)} dx dy \quad \text{Ec. (6)}$$

La transformada de Fourier tiene diversas aplicaciones como son: análisis, filtrado, reconstrucción y compresión de la imagen. Cuando se habla de imágenes digitales se emplea la Transformada discreta de Fourier (DFT).

### 1.6.2.2. Transformada de Fourier Discreta

Tenemos la función  $f(x)$

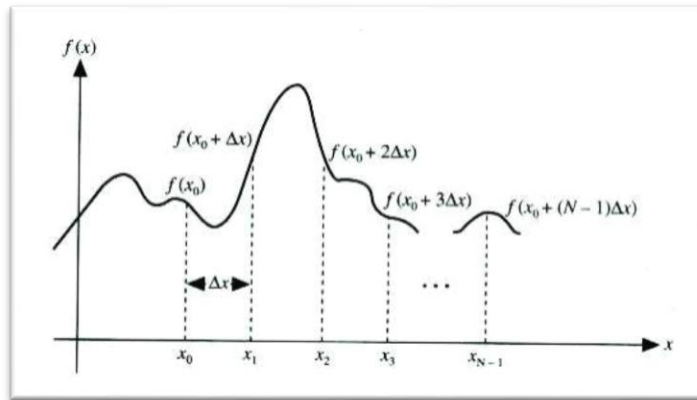


Figura 13.-Discretizacion de la función  $f(x)$

Fuente.- El autor

Después de discretizar la función  $f(x)$  tenemos:

$$\{f(x_0), f(x_0 + \Delta x), f(x_0 + 2\Delta x), \dots, f(x_0 + (N - 1)\Delta x)\} \quad \text{Ec. (7)}$$

Se toma  $N$  muestras separadas una distancia  $\Delta x$ , como se puede observar en la figura 13. La ecuación 7 se puede representar de la siguiente manera:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-j2\pi \frac{ux}{N}} \quad \text{Ec. (8)}$$

Los valores de  $u$  de la transformada discreta de Fourier corresponde a las muestras  $0, \Delta u, 2\Delta u, \dots, (N-1) \Delta u$ . Los términos  $\Delta x$  y  $\Delta u$  están relacionados de la siguiente manera:

$$\Delta u = \frac{1}{N\Delta x} \quad \text{Ec. (9)}$$

La DFT no posee todas las frecuencias que forman una imagen, pero si, un conjunto lo suficientemente grande como para describir completamente la imagen. Para una imagen de tamaño  $N \times N$ , la DFT está dada por la siguiente ecuación:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right)} \quad \text{Ec. (10)}$$

Donde  $f$  es el valor de la imagen en el dominio espacial y  $F$  en su dominio de la frecuencia.

La transformada de Fourier de una imagen produce tres términos: frecuencia espacial, magnitud (positiva o negativa) y fase. La frecuencia espacial es la frecuencia a través del espacio, con el cual se modula el brillo, la magnitud controla el contraste, y la fase representa como la onda se desplaza en relación al origen, en este caso cuanto se desplaza a la izquierda o a la derecha.

Para los algoritmos de procesamiento la magnitud de la imagen es lo más importante, ya que contiene toda la información que se necesita sobre la estructura geométrica de la imagen. En el caso que se desee realizar una modificación en la imagen, entonces se necesitara de estos tres componentes.

La transformada de Fourier es empleado para la elaboración de filtros. Se aplica filtros a una imagen para obtener una nueva imagen mejorando ciertas características que permitan efectuar operaciones de procesado sobre ella y pueda ser empleada en una aplicación específica.

Con la aplicación de los filtros se busca:

- ✓ Suavizar la imagen, para reducir la cantidad de variaciones de intensidad entre pixeles
- ✓ Eliminar ruido, para eliminar aquellos pixeles que son diferentes a los pixeles adyacentes
- ✓ Realzar bordes, para resaltar los bordes de una imagen
- ✓ Detectar bordes, para detecta los pixeles donde se produce un cambio notable en la intensidad

### **1.6.2.3. Filtros empleando Fourier**

La transformada de Fourier permite realizar filtros que nos permita ajustar el contenido de la frecuencia espacial de la imagen.

Con la Transformada de Fourier podemos realizar tres tipos de filtros:

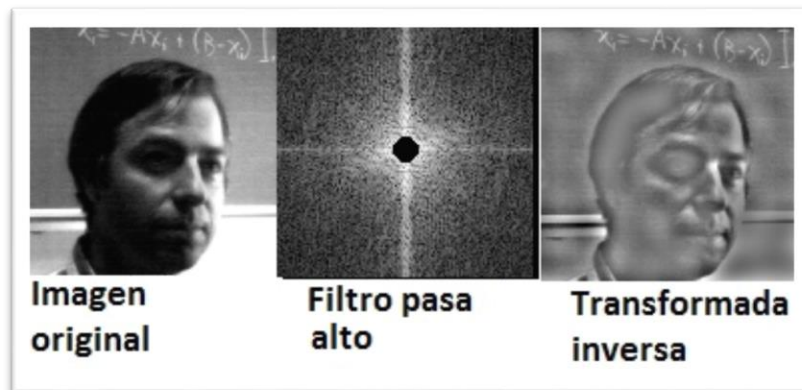
**Filtro pasa bajos:** Permite a los componentes de baja frecuencia espacial pasar a través de él, pero recorta las frecuencias espaciales altas, el resultado es equivalente a un filtro de suavizado, ya que elimina los cambios fuertes de intensidad, y reduce el ruido suavizando transiciones existentes.



*Figura 14.- Aplicación de un filtro pasa bajos*

*Fuente.- El autor*

**Filtro pasa alto.-** Conserva los componentes de frecuencia espacial alta y atenúa las frecuencias bajas. Este tipo de filtro es usado en detección de bordes



*Figura 15.- Aplicación de un filtro pasa alto*

*Fuente.- El autor*

**Filtro paso banda.-** Atenúa frecuencias muy altas o muy bajas, conservando un rango medio de frecuencias.



Figura 16.- Aplicación de un filtro pasa banda

Fuente.- El autor

### 1.6.3. Procesamiento de imágenes con OpenCV

OpenCV es una librería de visión por computador, escrita en lenguaje C/C++ desarrollada por Intel para el procesamiento de imágenes, tiene un sin número de algoritmos que abarcan muchas áreas, permiten al programador desarrollar aplicaciones sofisticadas rápidamente. OpenCV está diseñado para ser eficiente en cuanto a gasto de recursos computacionales y con enfoque a aplicaciones a tiempo real.

OpenCV posee una librería de aprendizaje automático (Machine Learning Library o MLL) útil para cualquier problema de aprendizaje automático, especializada en el reconocimiento estadístico de patrones y clustering.

#### 1.6.3.1. Filtrado de una imagen en OpenCV

**Smoothing Images.-** Permite difuminar una imagen mediante filtros pasa bajo, OpenCV posee una función `cv2.filter2D ( )`, la cual realiza una convolución entre un filtro kernel y una imagen. Esto lo que hace es sacar un promedio entre los pixeles, lo que produce un desenfoque en la imagen. Esta función es útil para eliminar el ruido.

OpenCV proporciona 4 métodos de desenfoque o suavizado de la imagen:

- ✓ Averaging
- ✓ Gaussian Filtering
- ✓ Median Filtering
- ✓ Bilateral Filtering

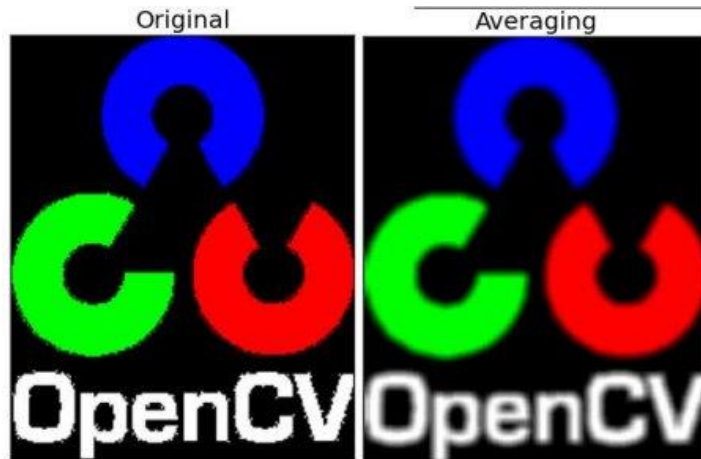


Figura 17.- Suavizado de una imagen mediante Avering

Fuente.-OpenCV-Python

**Image Gradients.-** Permite encontrar los bordes de una imagen mediante filtros paso alto, OpenCV posee tres funciones *cv2.Sobel ( )*, *cv2.Scharr ( )*, *cv2.Laplacian ( )*.

**Laplacian Derivatives.-** Permite calcular el laplaciano de una imagen mediante la siguiente ecuación:

$$\Delta src = \frac{d^2 src}{dx^2} + \frac{d^2 src}{dy^2} \quad Ec. (11)$$

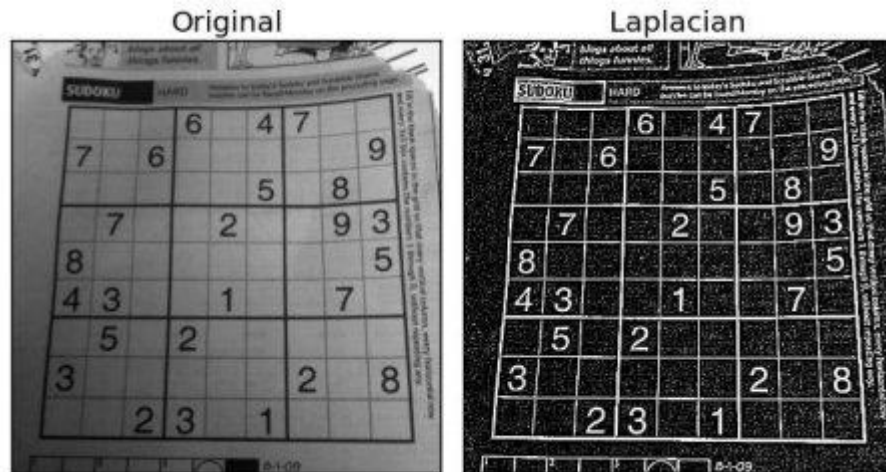


Figura 18.- Imagen aplicado el Laplaciano para encontrar los bordes

Fuente. - OpenCV-Python

**Canny Edge Detection.** - Es un algoritmo de detección de bordes, consta de múltiples, las cuales son:

*Noise Reduction.*- Eliminar el ruido por medio de un filtro gaussiano

Finding Intensity Gradient of the Image. - Se filtra la imagen mediante un filtro Sobel, para obtener la primera derivada tanto en dirección horizontal como vertical. De estas dos imágenes se encuentra la pendiente y la dirección del borde para cada pixel mediante la siguiente ecuación:

$$Edge\_Gradient(G) = \sqrt{G_x^2 + G_y^2} \quad Ec. (12)$$

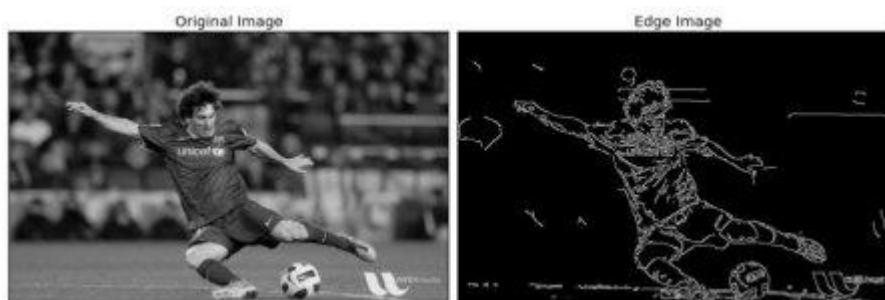
$$Angle(\theta) = \tan^{-1} \left( \frac{G_y}{G_x} \right) \quad Ec. (13)$$

*Non-maximun Suppression.*- Luego de encontrar la magnitud y dirección del gradiente, se procede a eliminar los pixeles no deseados, que no forman parte del borde.

*Hysteresis Thresholding.*- En esta etapa se decide cuáles son los bordes de la imagen.



OpenCV resume los pasos anteriores en una sola función, `cv2.canny ( )`.



*Figura 19.- Obtención de bordes mediante canny*

*Fuente.- OpenCV-Python*

## **1.7. Comunicación inalámbrica**

La comunicación inalámbrica es aquella que no emplea cables o medios físicos visibles de propagación, emplea como medio de propagación el aire, usa las ondas de radiofrecuencia para transportar la información. Entre los tipos de comunicaciones inalámbricas tenemos:

**Bluetooth.-** Es una tecnología inalámbrica de ondas de radio de corto alcance permite la transmisión de voz y datos entre diferentes dispositivos, permite la comunicación incluso a través de obstáculos

**Wifi.-** Es una tecnología inalámbrica que permite a los dispositivos conectarse a internet, está basado en las especificaciones IEEE 802.11, lo cual permite compatibilidad entre equipos.

## **1.8. Entorno de programación en Beaglebone Black**

### **1.8.1. Introducción a la beaglebone**

Beaglebone Black es un mini pc que nos permite realizar múltiples tareas como si fuese una computadora normal. Para empezar a utilizar la beaglebone debemos seguir los siguientes pasos:

Conectamos la tarjeta a nuestra pc mediante el cable USB, como se muestra en la figura 20.



Figura 20.- Conexión de la beaglebone a la pc

Fuente.- El autor

Ingresamos a la página <http://beagleboard.org> y descargamos e instalamos los driver de la tarjeta.

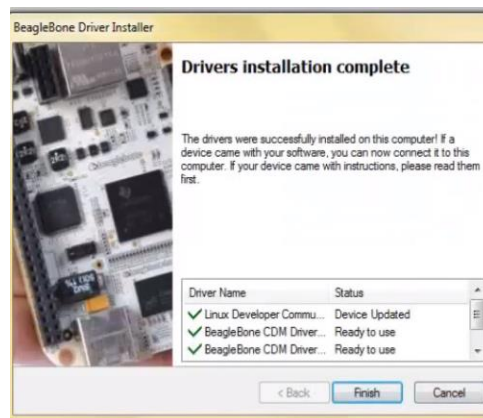


Figura 21.- Instalación de drivers de la beaglebone

Fuente.- El autor

Una vez instalado los drivers, ingresamos en cualquier navegador y colocamos la siguiente dirección: 192.168.7.2, si todo va bien, nos saldrá esta pantalla.

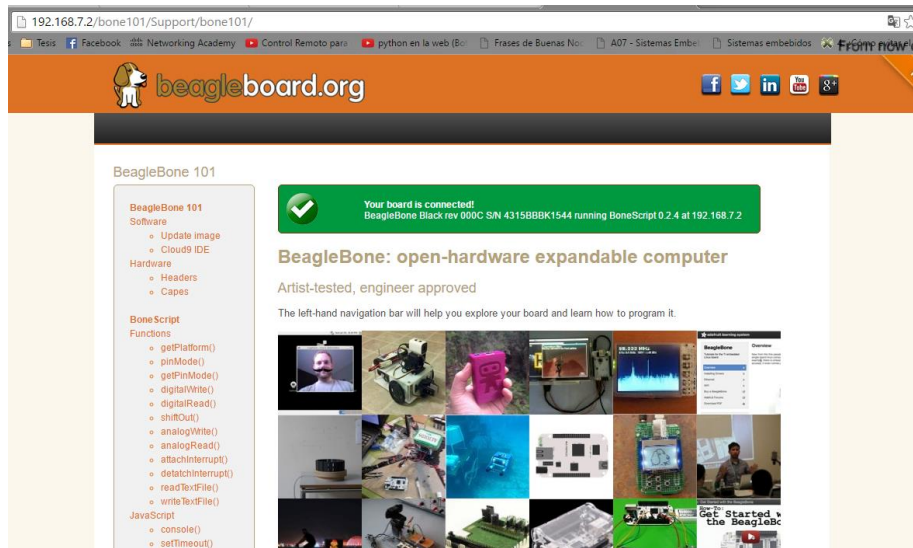


Figura 22.- Comprobación de los drivers instalados sea correctos

Fuente.- El autor

En esta página podemos encontrar información útil, trabajos desarrollados en esta tarjeta.

Se puede emplear la Beaglebone Black de varias maneras, una de ellas es conectando a la tarjeta periféricos externos, como es el caso de una pantalla, teclado y un mouse para usarla como una computadora de escritorio, lo cual nos implicaría el uso de muchos recursos, u otra manera es mediante una conexión remota, ya sea empleando el programa PuTTY<sup>13</sup> o un escritorio remoto mediante VNC<sup>14</sup>. Primero debemos descargarnos e instalar el programa PuTTY. Abrimos el programa y colocamos los datos que se muestran en la *figura 23*, damos click en open.

<sup>13</sup> Sirve para iniciar seccion remota con una maquina o servidor

<sup>14</sup> Control Remoto Grafico.- Permite controlar una maquina remoramente

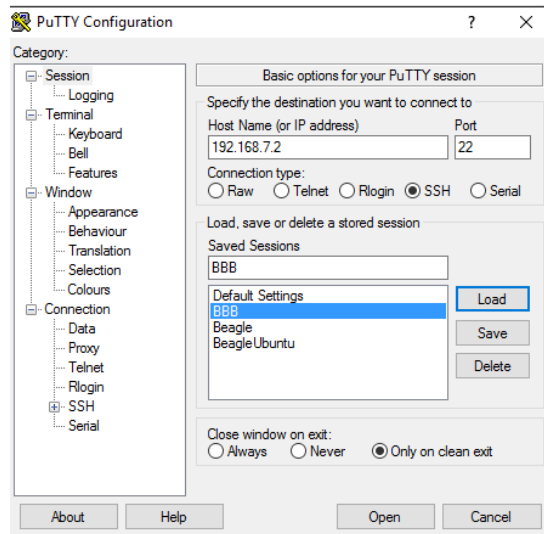


Figura 23.- Configuración del programa PuTTY

Fuente.- El autor

En la ventana que nos aparece escribimos **root** luego damos enter

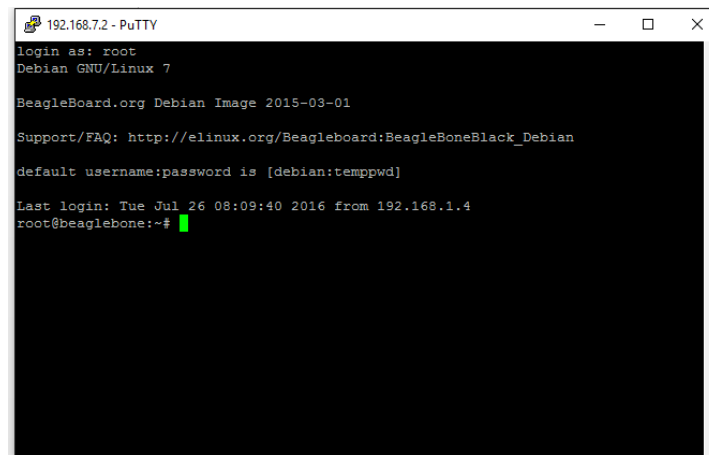


Figura 24.- Conexión de la beaglebone mediante SSH

Fuente.- El autor

En este momento la beaglebone está lista para ser usada, pero en este caso vamos a instalar un escritorio remoto para que sea más fácil su utilización. Primero debemos ingresamos el siguiente código: **nano /etc/lightdm/lightdm.conf** y pulsamos enter, y colocamos los siguientes valores:

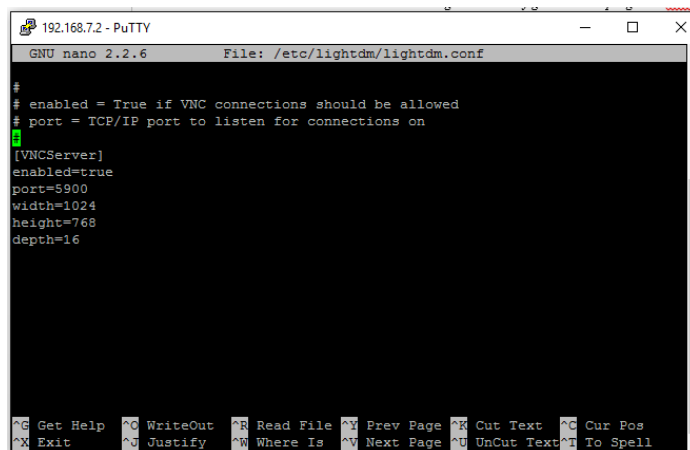


Figura 25.-Configuración de VNC en la beaglebone

Fuente.- El autor

En nuestra computadora instalamos VNC viewer, ponemos la siguiente dirección 192.168.7.2:5900 damos click en conectar y listo.



Figura 26.- Escritorio de la beaglebone con el sistema operativo Debían

Fuente.- El autor

## 1.8.2. Actualización e instalación de librerías

Es necesario tener conectada la tarjeta a internet, para actualizar colocamos los siguientes comandos: ***sudo apt-get install update***, después ingresamos el comando ***sudo apt-get install upgrade***. Para la instalación de librerías es muy parecido debemos ingresar ***sudo apt-get install (librería a instalar)***, por ejemplo vamos a instalar Python, entonces tenemos ***sudo apt-get install Python***

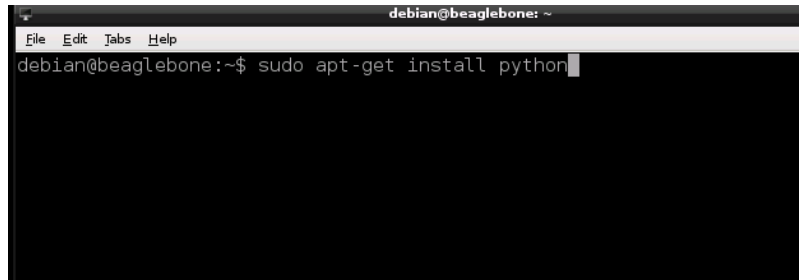


Figura 27.- Instalación de librerías en beaglebone usando Debian

Fuente.- El autor

## 1.9. Creación de un proyecto en Python en la tarjeta Beaglebone Black

Python es un lenguaje gratuito, multiplataforma, de código abierto muy potente y fácil de aprender, orientado a objetos, preparado para desarrollar todo tipo de programas, incluso se puede desarrollar servidores web y páginas web. Es un lenguaje interpretado, no necesita compilar el código fuente para ejecutarlo.

La creación de un proyecto en Python se presentara a continuación:

Para abrir la consola de línea de comandos en Debian hacemos click en:

- Inicio
  - Accesorios
    - LXTerminal

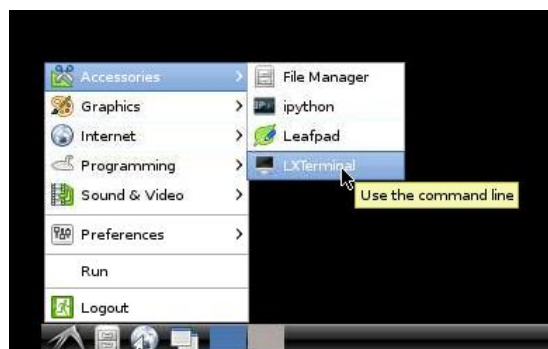


Figura 28.- Abrir LXTerminal

Fuente.- El autor

Para la creación de un programa en Python escribimos en la consola el siguiente comando:

- `sudo nano nombre.py`

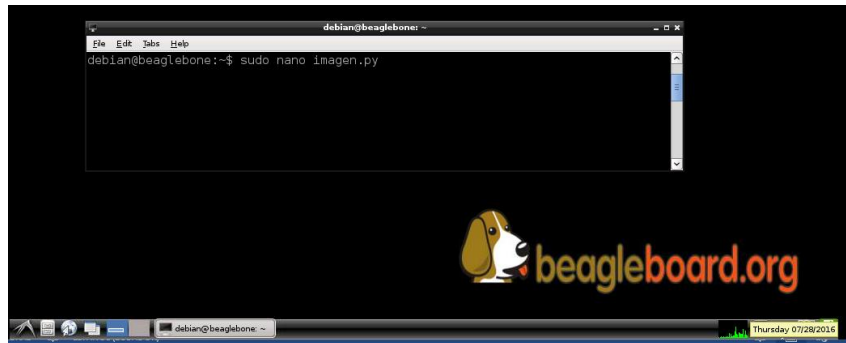


Figura 29.- Creación de un nuevo programa python en beaglebone

Fuente.- El autor

Donde *nombre* es el nombre que le vamos a dar al programa en este caso se llamara *imagen*, presionamos enter y nos aparecerá otra ventana en la cual procedemos a escribir nuestro código. A continuación se realizara un programa que permita leer una imagen mediante python empleando las librerías de OpenCV

- Importamos las librerías a emplear con el comando “*import*”
  - Empleamos la función “*cv2.imread( )*” para leer la imagen
    - La función “*cv2.imshow( )*” nos permite mostrar la imagen en una ventana
      - La función “*cv2.destroyAllWindows( )*” cierra todas las ventanas abiertas

```
import numpy as np
import cv2

#Leer una imagen
img = cv2.imread('portada.jpg',0)
cv2.imshow('Imagen',img)

cv2.destroyAllWindows
```

Figura 30.- Programa en python

Fuente.- El autor

## **CAPÍTULO II**

### **2. METODOLOGÍA**

#### **2.1. Tipo de estudio**

##### **2.1.1. Descriptivo-Correlacional**

Este proyecto emplea el tipo de estudio descriptivo-correlacional, para analizar el problema identificando cada una de sus partes obteniendo un mayor conocimiento y una mejor comprensión del objeto de estudio. Mediante la observación se busca describir el comportamiento del usuario ante las instrucciones proporcionadas por el sistema de ayuda y conocer los detalles de funcionamiento del sistema.

#### **2.2. Métodos, Técnicas e Instrumentos**

##### **2.2.1. Métodos**

###### **2.2.1.1. Analítico**

Se emplea el método analítico, debido a que se inicia analizando cada uno de los algoritmos empleados en el software embebido del sistema y la manera de interactuar entre sí, para que el software funcione de manera optima

##### **2.2.2. Técnicas**

###### **2.2.2.1. Observación**

Con esta técnica podemos recolectar información para luego analizarla y de esta manera obtener datos que servirán de pautas para el desarrollo del sistema de ayuda.

##### **2.2.3. Instrumentos**

Los instrumentos necesarios son datasheets, video tutoriales de sitios web, blogs, libros, folletos. Archivos útiles para el desarrollo del sistema.



## 2.3. Población y muestra

### 2.3.1. Población

La población está orientada a personas no videntes, sin embargo se trata de un prototipo, por el cual no se determina un valor como población.

### 2.3.2. Muestra

El proyecto consta del desarrollo de un prototipo por lo cual no se determina una muestra.

## 2.4. Hipótesis

El desarrollo de un sistema embebido para ayuda a la natación de personas no videntes mediante el reconocimiento de imágenes incentiva al no vidente a realizar este deporte debido a que le otorga seguridad a través de su sistema de alerta mediante voz en tiempo real.

## 2.5. Operacionalización de variables

Variables	Dimensiones	Indicadores
<b>Variable independiente:</b> Software embebido	Algoritmo para el reconocimiento de imágenes  Sistema de alerta	Detección de objetos Seguimiento de objetos  Audio
<b>Variable dependiente:</b> Ubicación del usuario	Coordenadas	Trayectoria

Tabla 2.- Operacionalización de las variables

Fuente.- El autor

## 2.6. Procedimientos

Para el desarrollo del sistema de ayuda de natación se emplea una serie de pasos:

- ✓ Identificación de los principales inconvenientes que tiene el nadador no vidente.

- ✓ Diseñar un sistema de apoyo para que el no vidente pueda nadar de una manera más confortable.
- ✓ Selección de los componentes electrónicos para el desarrollo del sistema
- ✓ Desarrollo del software embebido para el reconocimiento de imágenes.
- ✓ Desarrollo del software embebido para el sistema de comunicación.
- ✓ Desarrollo del software embebido para el sistema de alertas.
- ✓ Pruebas de funcionamiento del sistema para verificar si cumple con los objetivos planteados.
- ✓ Análisis de resultados.

## **2.7. Procedimiento y análisis**

### **2.7.1. Inconvenientes que presenta un nadador no vidente**

Las personas no videntes o con baja visión presentan problemas para desplazarse en entornos desconocidos, por lo que requieren de una persona que los ayude. Debido a que en el agua no existen puntos de referencia, no puede establecer una posición de su ubicación o la posición de algún objeto o la distancia a la que se encuentra, por estos y otros motivos se requiere de un sistema que realice estas actividades.

#### **2.7.1.1. Diseño y selección de los componentes para el desarrollo del sistema de apoyo**

Una vez analizados los inconvenientes antes mencionados se diseñó el siguiente sistema:

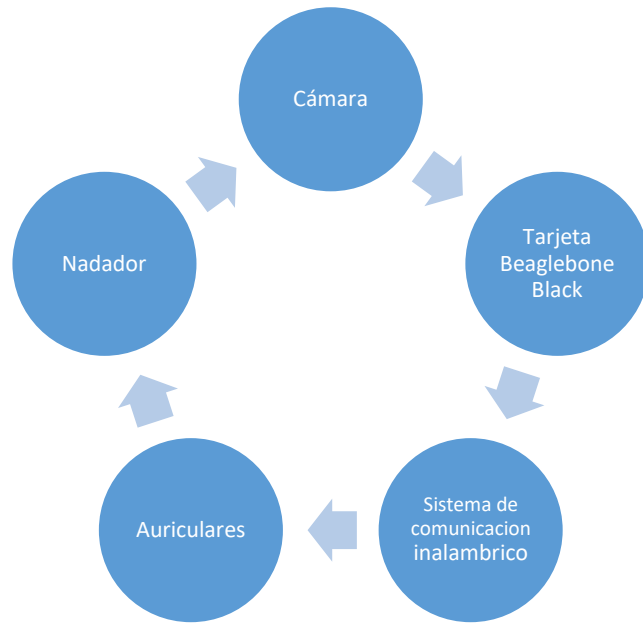


Figura 31.- Diagrama de bloques del sistema a desarrollar

Fuente.- El autor

Para el desarrollo de este sistema se va a emplear los siguientes materiales:

Cantidad	Material
1	Cámara web USB
1	Tarjeta Beaglebone Black
1	Auricular
1	Tarjeta Intel de Edison (Sistema de comunicación)

Tabla 3.- Listado de materiales para el sistema

Fuente.- El autor

### 2.7.2. Implementación del sistema embebido para ayuda a la natación

Para el desarrollo de este sistema se emplea la mini pc Beaglebone Black, esta será la encargada de realizar los procesos lógicos. Se emplea el lenguaje de programación Python y la librería OpenCV para el procesamiento de las imágenes.

La imagen es captada por una cámara web conectada a la Beaglebone black, la información resultante del procesamiento de la imagen será enviada al usuario mediante una comunicación inalámbrica.

El usuario recibe instrucciones de audio mediante auriculares conectados a través del bluetooth mediante la tarjeta Intel de Edison.. A continuación se muestra un esquema del sistema:

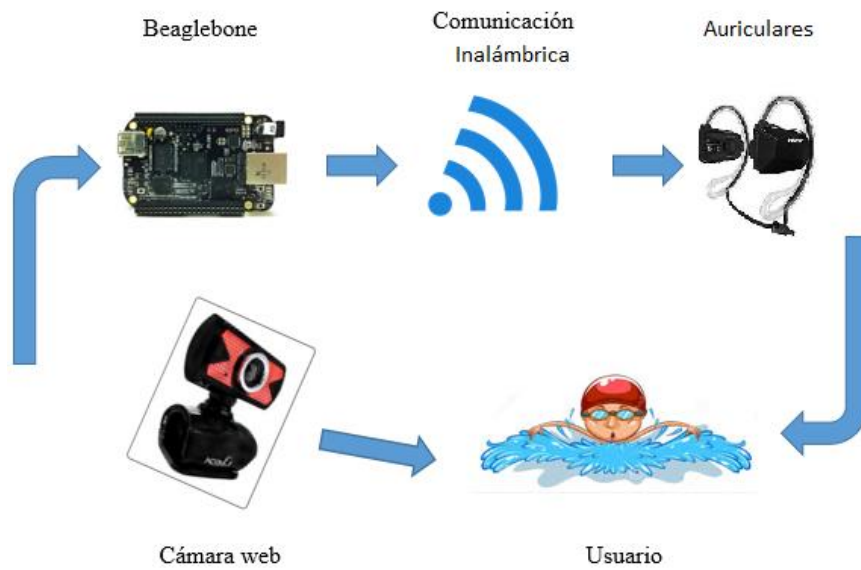


Figura 32.- Esquema del sistema de ayuda

Fuente.- El autor

### 2.7.3. Desarrollo del software embebido para el reconocimiento de imágenes

Para el desarrollo del software se realizara una serie de procesos de manera estructurada como se muestra en la siguiente figura:

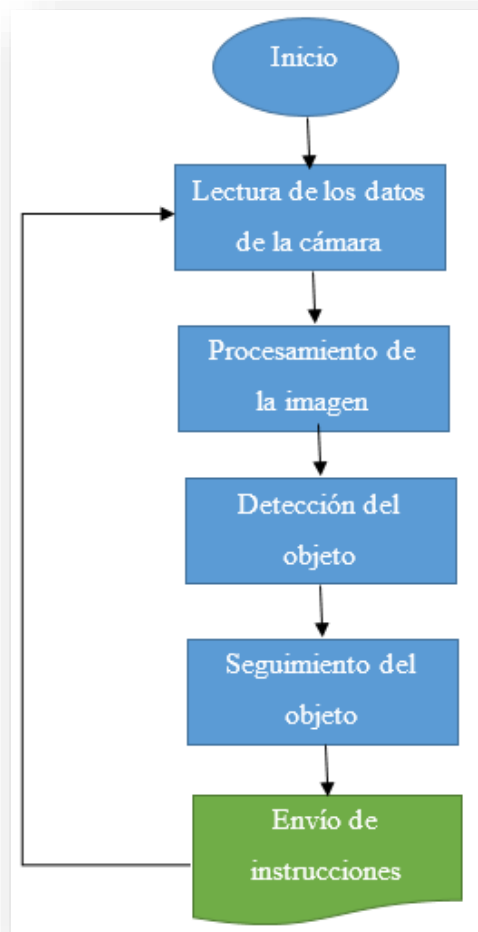


Figura 33.- Diagrama del proceso ejecutado en la beaglebone

Fuente.- El autor

### Inicialización del programa

Para iniciar el programa se debe importar las librerías a utilizar y otros parámetros iniciales.

Las librerías que se emplean en este proyecto son las siguientes:

```

import cv2
import numpy as np
import math
  
```

*cv2*: Contienen las librerías de OpenCV

*numpy*: Contiene soporte para vectores y matrices

*math*: Empleada para utilizar funciones matemáticas

Además se debe configurar parámetros adicionales como:

Tamaño de la ventana de video

```
frame_w = 320  
frame_h = 240
```

Índice de propiedades

```
FRAME_PROP_WIDTH = 3  
FRAME_PROP_HEIGHT = 4
```

### **Lecturas de los datos de la cámara**

La cámara está conectada mediante el puerto USB, es recomendable usar una cámara de alta definición para obtener una mejor resolución de la imagen.

Para poder emplear la cámara se debe iniciar la captura de video

```
cap = cv2.VideoCapture(0)  
cap.set(FRAME_PROP_WIDTH, frame_w)  
cap.set(FRAME_PROP_HEIGHT, frame_h)
```

Para leer los frames proporcionados por la cámara se usa el siguiente comando:

```
_, img = cap.read( )
```

### **Procesamiento de la imagen**

Para realizar la detección de objetos primero se debe tratar la imagen de tal manera que la imagen resultante contenga datos útiles para el diseñador, en este caso particular se aplica filtros a la imagen. Se realiza dos tipos de filtros.

**Caso 1:** Filtros empleando la transformada de Fourier.

Se aplicara un filtro pasa bajos con el objetivo de eliminar las frecuencias altas y obtener una imagen simple, es decir una imagen libre de bordes.

**Caso 2:** Filtros empleando las funciones de OpenCV

Como ya se mencionó anteriormente OpenCV posee un sin número de funciones que permiten el tratado de las imágenes, en este caso se emplea la función “*cv2.blur*” con el objetivo de eliminar el ruido

### **Detección de objetos**

Para detectar un objeto es necesario extraer de la imagen los pixeles de interés. Para ello se emplea el siguiente proceso:

- Transformar la imagen de BGR<sup>15</sup> a HSV
- Aplicar la técnica de umbral a la imagen para extraer los pixeles de interés
- Encontrar los contornos del objeto
- Encontrar el área del objeto para determinar si es nuestro objeto de interés

### **Seguimiento del objeto**

Una vez detectado el objeto a seguir, se determina su ubicación, para ello se encuentra las coordenadas del objeto, se emplea la técnica de detección de contornos, posteriormente encontramos el centro del contorno que en este caso es las coordenadas del objeto.

Una vez obtenidas las coordenadas del objeto es posible saber su ubicación y distancia en la que se encuentra, lo que facilita el envío de instrucciones para guiarlo de manera segura a través de la ruta establecida por el instructor.

### **Elaboración de la ruta para la prueba del software**

Para la elaboración de la ruta se emplea las librerías de OpenCV. Empleamos la función mouse callback, que dependiendo de los eventos se puede realizar diferentes tareas. Entre las funciones tenemos:

---

<sup>15</sup> Composición del color en términos de la intensidad de los colores primarios de la luz.

- LBUTTONDBLCLK
- LBUTTONDOWN
- MOUSE MOVE
- LBUTTONUP

Antes de usar estas funciones se crea una ventana con una plantilla como vemos en la siguiente figura:

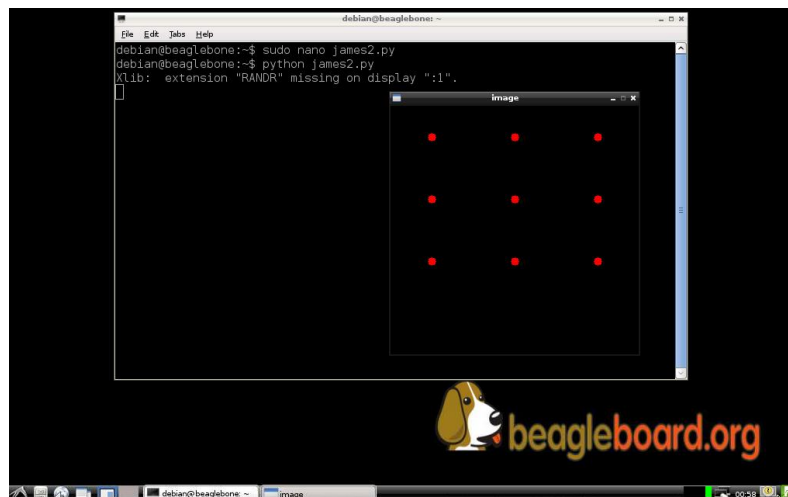


Figura 34.- Plantilla para dibujar la ruta

Fuente.- El autor

Después se procede a dibujar la ruta como se observa en la siguiente figura:

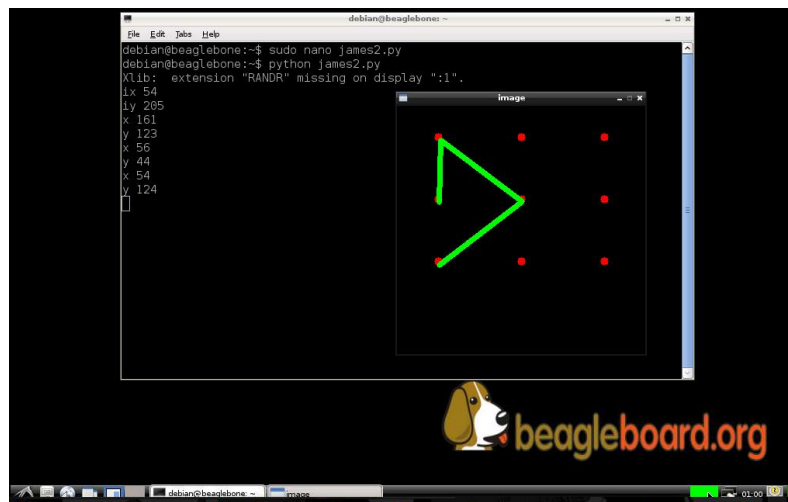


Figura 35.- Dibujo de la ruta

Fuente.- El autor



Configuración de la tarjeta Intel Edison para la comunicación inalámbrica:

Se debe descargar el software de la página oficial de Intel para configurar la tarjeta de manera rápida, seguimos los siguientes pasos:

- ✓ Instalar los drivers USB
- ✓ Actualizar el Firmware
- ✓ Crear un usuario y contraseña para la conexión SSH
- ✓ Conectamos a la tarjeta a una red WiFi

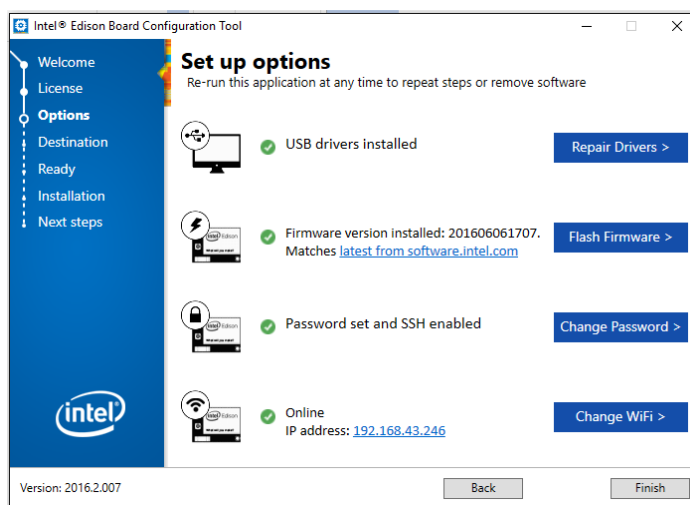


Figura 36.- Resultado de la configuración de la tarjeta Intel Edison

Fuente.- El autor

Para ingresar a la tarjeta se emplea el software PuTTY, con el usuario y contraseña creado en el paso anterior. Se habilita el modulo bluetooth para conectar la tarjeta a los auriculares bluetooth. Se emplea los siguientes comandos:

```
rftkill unblock bluetooth  
bluetoothctl  
[bluetooth] scan on  
pair 30:21:85:A1:46:3D
```

### Calculo de la distancia entre puntos

Para calcular la distancia necesitamos las coordenadas de los puntos:

$$P_1 = (x_1 + y_1)$$

$$P_2 = (x_2 + y_2)$$

Distancia entre  $P_1$  y  $P_2$  es:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \text{Ec. (14)}$$

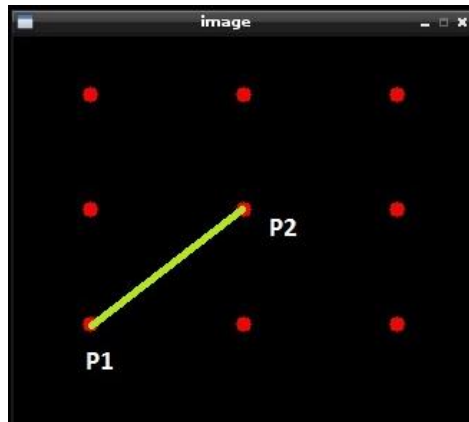


Figura 37.- Calculo de la distancia entre dos puntos

Fuente.- El autor

### Calculo del ángulo entre dos rectas

Lo primero que se necesita es las distancias entre los puntos empleando la *ecuación 14*.

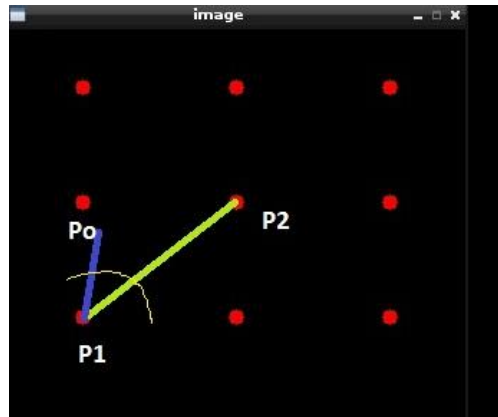
$$\begin{aligned} a &= P_2 - P_0 \\ b &= P_0 - P_1 \\ c &= P_2 - P_1 \end{aligned}$$

Para calcular el ángulo se emplea la siguiente formula trigonométrica

$$a^2 = b^2 + c^2 - 2bc \cos(P_1) \quad \text{Ec. (15)}$$

Despejando  $P_1$  tenemos:

$$P_1 = \cos^{-1} \left( \frac{b^2 + c^2 - a^2}{2bc} \right) \quad \text{Ec. (16)}$$



*Figura 38.- Cálculo del ángulo entre dos rectas*

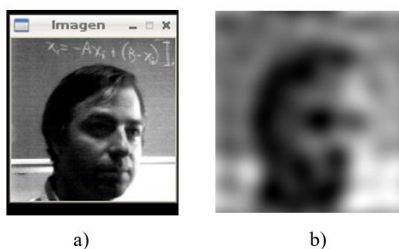
*Fuente.- El autor*

## CAPÍTULO III

### 3. RESULTADOS

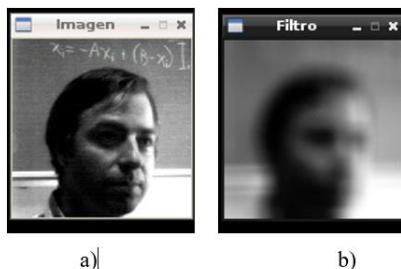
#### 3.1. Resultados de la prueba de los filtros a emplear para procesar las imágenes

Después de realizar las pruebas de los filtros se obtuvo como resultado que los filtros empleando las funciones de OpenCV tienen un mejor rendimiento, como se muestra a continuación en la *figura 39* y *figura 40*.



*Figura 39.- Aplicación de un filtro pasa bajo empleando Fourier*

*Fuente.- El autor*



*Figura 40.- Aplicación de un filtro empleando las funciones de OpenCV*

*Fuente.- El autor*

Como podemos ver en la *figura 39 b)* la imagen es muy borrosa y hasta un poco distorsionada, a diferencia de la *figura 40 b)* en la que se puede observar de una mejor manera a la persona de la fotografía. Por lo tanto los filtros para procesar la imagen serán empleando las funciones de OpenCV mas especifico la función `cv2.blur ( )` debido a nos permite obtener una imagen suavizada.

Otro de los motivos para emplear los filtros con funciones de OpenCV es las líneas de código, solo se emplea la función, caso contrario de los filtros desarrollados mediante Fourier emplean extensas líneas de código ocupando mayor memoria y produciendo un mayor tiempo en la ejecución del programa.

### 3.2. Resultados de la prueba del sistema embebido para la ayuda a la natación

Se realizó pruebas de funcionamiento del sistema embebido en dos escenarios:

#### Escenario 1: Empleando un objeto de un color específico

Se coloca una pelota de color azul frente a la cámara, el sistema detecta con facilidad a la pelota como podemos ver en la *figura 41*, además nos proporciona las coordenadas del objeto.



Figura 41.- Detección de la pelota

Fuente.- El usuario

El sistema presenta una respuesta rápida al mover la pelota frente la cámara, reaccionando de una manera óptima. Además al momento de guiarlo nos proporciona las instrucciones correctas para encaminarlo a través de la ruta como se observa en la *figura 42* y *figura 43*.

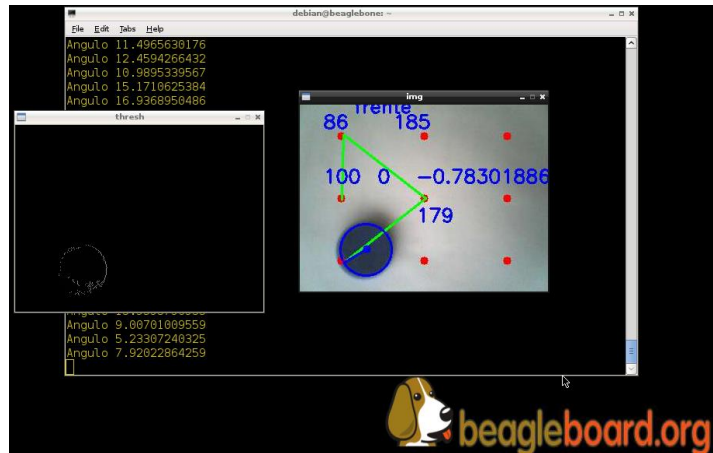


Figura 42.- Seguimiento de la pelota

Fuente.- El autor

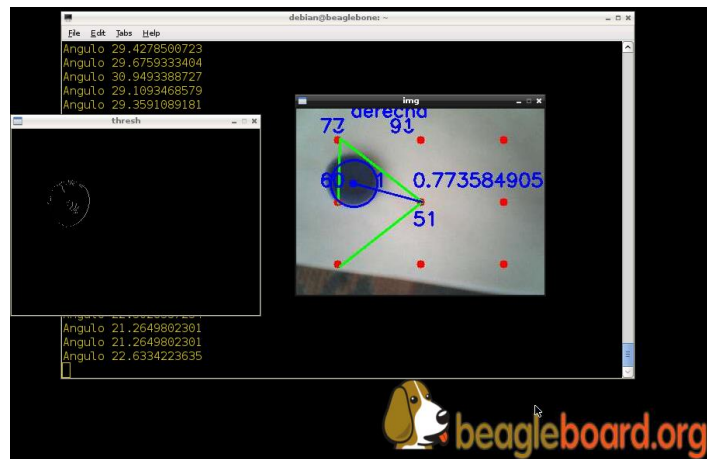


Figura 43.- Seguimiento de la pelota

Fuente.- El autor

## Escenario 2: Pruebas efectuadas en el usuario

Para realizar estas pruebas, primero se acondiciona al usuario para que sus respuestas sean lo mayor parecidas a las de un no vidente, se le coloca una venda en los ojos para que no observe su entorno y solo pueda desplazarse siguiendo las instrucciones proporcionadas por el sistema.

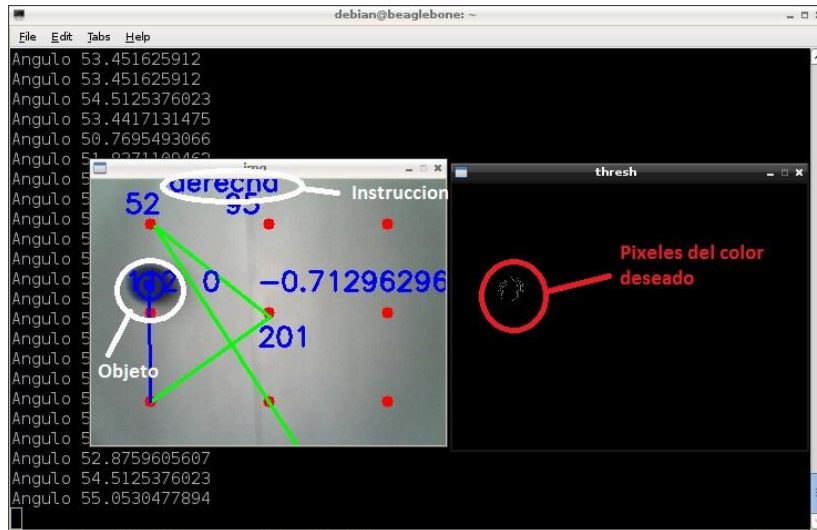


Figura 44.- Prueba del software con el objeto

Fuente.- El autor

En la *figura 44* se observa una serie de datos en la pantalla, como son el objeto, que es el objeto a seguir, las instrucciones, que son las guías para que el objeto siga la ruta diseñada. Para determinar estos datos necesitamos saber la distancia que existe entre el objeto y los puntos de referencia, así como el ángulo que existe entre la trayectoria del objeto (línea azul) y la trayectoria de la ruta (línea verde) para determinar si el objeto debe ir a la izquierda o a la derecha.



Figura 45.- Prueba del sistema en el usuario

Fuente.- El autor

## CAPÍTULO IV

### 4. DISCUSIÓN

Los sistemas de reconocimiento de imágenes son empleados en muchos campos, reconocimiento de caracteres, identificación de personas, biometría, entre otros. Por qué no emplear este método en beneficio de las personas, para hacer su vida más llevadera.

El sistema embebido de ayuda para la natación constituye un medio útil para guiar a las personas no videntes en el agua, debido a que en el agua no existen puntos de referencia, no pueden establecer una posición de su ubicación o la posición de algún objeto o la distancia a la que se encuentra, este sistema localiza al usuario y le otorga la posibilidad de desplazarse en el agua de manera segura, incluso el sistema es capaz de guiar al usuario a través de una ruta previamente diseñada, de esta manera se lo logra que la natación no sea solo recreativa sino también educativa.

Debido a que el sistema posee la capacidad de reconocer imágenes para sortear obstáculos, y así guiar al usuario en ambientes desconocidos para ellos como es una piscina, podría no solo ser empleado para el agua, sino también utilizado en otros ámbitos de la vida del usuario.

Imagínese una vida donde gracias a la tecnología las personas con capacidades diferentes puedan realizar las mismas actividades que las demás personas. La tecnología avanza a pasos agigantados y está en nosotros aprovecharla para nuestro beneficio o en beneficio de los demás.



## CAPÍTULO V

### 5. Conclusiones y Recomendaciones

#### 5.1. Conclusiones

Una vez finalizada la investigación se concluye lo siguiente:

- El sistema de ayuda para la natación provee al usuario la orientación necesaria para que este se movilice en el agua mediante instrucciones por voz en tiempo real y de esta manera evitando cualquier clase de accidentes, generando así autonomía
- El sistema presenta un tiempo de respuesta óptimo, en el caso de perder de vista al usuario lo detecta de manera casi inmediata, otorgando al usuario seguridad ante cualquier riesgo, ya que mientras este en el rango de la cámara el sistema le proporcionara las instrucciones pertinentes según sea el caso.
- Con un ligero ajuste en la programación, el sistema es capaz de guiar a más de un usuario debido a que emplea el reconocimiento de objetos por color para localizar a cada usuario, el usuario debe llevar un gorro de distinto color que lo diferencie el uno del otro.
- El sistema es compacto por lo que puede ser trasladado y ensamblado de manera sencilla en cualquier otro sitio.

## 5.2. Recomendaciones

Una vez realizadas las conclusiones del proyecto se proponen las siguientes recomendaciones:

- Ampliar el rango de visión del sistema para que abarque un área más grande y de esta manera el usuario pueda desplazarse con mayor libertad.
- Diseñar una interfaz que le permita al instructor elaborar la ruta a seguir por el usuario de una manera más fácil y rápida
- Mejorar el sistema de alertas adhiriendo o quitando comandos
- Mejorar el sistema de comunicación empleado para que llegue los comandos de voz al usuario

## **CAPÍTULO VI**

### **6. PROPUESTA**

#### **6.1. Título de la propuesta**

DESARROLLO DE UN SISTEMA EMBEBIDO PARA AYUDA A LA NATACIÓN DE PERSONAS NO VIDENTES MEDIANTE EL RECONOCIMIENTO DE IMÁGENES

#### **6.2. Introducción**

La tecnología siempre ha ido en busca de mejoras para el desarrollo del ser humano, una enorme contribución es la natación para personas no videntes que mejora su estilo de vida, aumenta la capacidad física, levanta la autoestima, acrecienta la coordinación y orientación. Para las personas no videntes es muy difícil orientarse en ambientes no conocidos como en este caso el agua debido a la falta de puntos de referencia que le permitan orientarse, por lo tanto necesita de personas que le ayuden, lo cual resulta no ser optimo debido a que necesita la presencia permanente de alguien supervisándolo.

Una manera de optimizar esta actividad es mediante el uso de un sistema de ayuda a la natación que le permita a la persona orientarse en el agua, haciéndolo sentir seguro mientras realiza esta actividad. Un método para el desarrollo de este sistema es mediante el procesamiento de imágenes, este sistema se enfoca en detectar objetos y alertar al usuario de la presencia de los mismos, estas alertas son mediante voz para que sea de fácil comprensión e interpretación. El sistema es capaz de guiar al usuario a través de piscina evitando que este colisione con obstáculos.

La gran ventaja de este sistema es que se ejecuta en tiempo real, la rapidez con la que realiza los procesos ya sea procesamiento de imágenes o cálculos matemáticos

es notable, permitiendo dar respuesta casi inmediata a cualquier inconveniente, su sistema de alerta es en forma audible, permitiendo que sea de fácil interpretación lo cual lo convierte en un sistema de gran eficiencia.

### **6.3. Objetivos**

#### **6.3.1. Objetivo general**

- Desarrollar un sistema embebido para ayuda a la natación de personas no videntes mediante el reconocimiento de imágenes

#### **6.3.2. Objetivos Específicos**

- Investigar acerca de los diferentes tipos de single board computers
- Seleccionar la single board computers a utilizar
- Desarrollar el software embebido que permita realizar el reconocimiento de imágenes
- Desarrollar el software para la comunicación entre la single board computers y el usuario, considerando que el usuario se encuentra en el agua
- Guiar al usuario por medio de comandos de voz

### **6.4. Fundamentación Científico – Técnico**

El sistema embebido para ayuda a la natación de personas no videntes mediante el reconocimiento de imágenes tiene una significativa representación en la población no vidente, ya que le otorga la posibilidad de orientarse en el agua, permitiéndole movilizarse de una manera segura.

### **6.5. Descripción de la propuesta**

Este proyecto consiste en guiar al nadador no vidente a través de la piscina mediante comandos de voz emitidos por una mini pc. Una cámara capta características del

entorno del usuario y la mini pc procesa la información mediante técnicas de reconocimiento de imágenes, de esta manera es posible la obtención de las coordenadas de ubicación del usuario y la detección de obstáculos. La mini pc proporciona instrucciones para que pueda orientarse y moverse de manera segura en la piscina, estas instrucciones son emitidas mediante comunicación inalámbrica desde la mini pc hasta el usuario que se encuentra en el agua.

## 6.6. Diseño organizacional

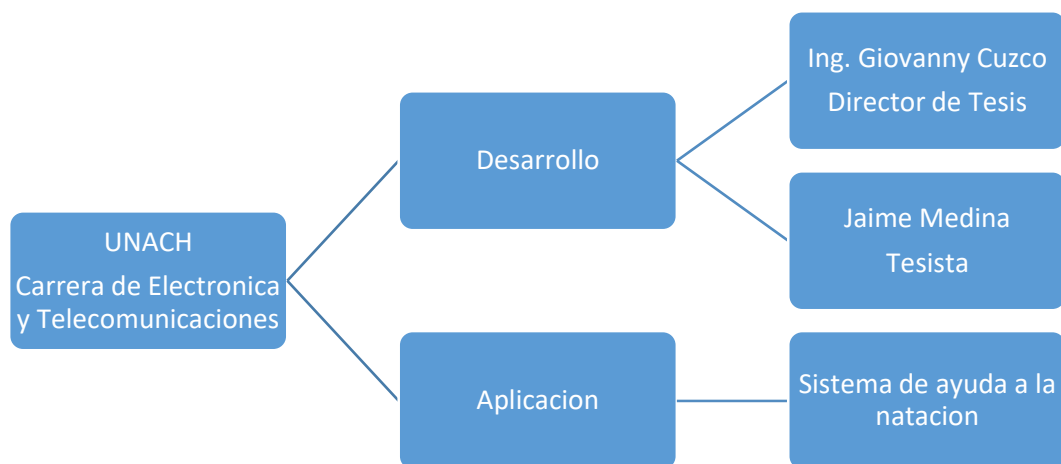


Figura 46.- Esquema Organizacional

Fuente.- El autor

## 6.7. Monitoreo y Evaluación de la Propuesta

La evaluación de la propuesta se realizara sometiendo al sistema a pruebas con personas para así determinar si el uso de este sistema de ayuda es útil.

El impacto que produce el desarrollo del sistema de ayuda a la natación es de gran impacto al lograr guiar al usuario a través de la trayectoria de una manera adecuada.

## 7. BIBLIOGRAFÍA

- arduinodhtics*. (3 de Julio de 2016). Obtenido de Arduino: Tecnología para todos:  
<http://arduinodhtics.weebly.com/iquestqueacute-es.html>
- Arevalo Vazquez, E. E., Zuñiga Lopez, A., Villegas Cortez, J., & Aviles Cruz, C. (2015). *Implementacion de reconocimiento de objetos por color y forma en un robot movil*. Mexico, D.F: Universidad Autonoma Metropolitana Azcapotzalco, Departamento de Electronica.
- Blanco García-Moreno, J. L. (26 de julio de 2016). *La mirada del golem*. Obtenido de <http://miradadelgolem.blogspot.com/2013/09/ubuntu-en-pandaboard.html>
- blogthinkbig. (26 de julio de 2016). *Cuatro alternativas a Arduino*. Obtenido de <http://blogthinkbig.com/4-alternativas-arduino-beaglebone-raspberrypi-nanode-waspmote/>
- Caballero Paz, A. (2013). *DESARROLLO DE UN CONTROLADOR MIDI NO CONVENCIONAL, IMPLEMENTADO EN UN SISTEMA EMBEBIDO, UTILIZANDO EL KINECT*. Bogota, Colombia: Universidad de San Buenaventura.
- Casco, S. M. (2014). *Raspberry Pi, Arduino y Beaglebone Black*. Asuncion, Paraguay: Universidad Catolica Nuestra Señora de la Asuncion.
- Castañeda, R. (26 de julio de 2016). *Software y Hardware*. Obtenido de <http://softhardaa.blogspot.com/p/hardware-perifericos-de-entrada-y-de.html>
- Coley, G. (26 de julio de 2016). *Elinux*. Obtenido de <http://elinux.org/Beagleboard:BeagleBoneBlack>
- Echavarría, J. (26 de julio de 2016). *Designing Electronics*. Obtenido de <http://www.jechavarria.com/2014/05/22/medidas-y-dimensiones-de-la-tarjeta-beaglebone-black/?lang=es>
- Informaticahoy. (26 de julio de 2016). *Informaticahoy*. Obtenido de <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-el-sistema-operativo.php>

- Mordvintsev , A., & K, A. (27 de julio de 2016). *OpenCV-Python*. Obtenido de [http://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_imgproc/py\\_filtering/py\\_filtering.html](http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html)
- Nanode*. (5 de Julio de 2016). Obtenido de What is nanode: <http://www.nanode.eu/what-is-nanode/>
- Raponi, M., & Bonnin, R. (2011). *Dispositivo de rehabilitacion visual basado en sistemas embebidos del tipo ARM*. Buenos Aires, Argentina.
- Salgado Patron, J., Vasquez Diaz, L., & Vidal Solano, M. (2013). Diseño e implementacion de algoritmo para el procesamiento de imagenes en sistemas embebidos. *Ingenieria y Region*, 41-53.
- Sola, A. (26 de julio de 2016). *Pucomí*. Obtenido de <http://pucomi.blogspot.com/2014/08/raspberry-pi-first-steps.html>
- Solis Gonzalez, J. (1 de Julio de 2016). *Sistemas Embebidos: Innovando hacia los Sistemas Inteligentes*. Obtenido de SemanticWebBuilder: [http://www.semanticwebbuilder.org.mx/es\\_mx/swb/Sistemas\\_Embebidos\\_Innovando\\_hacia\\_los\\_Sistemas\\_Inteligentes\\_](http://www.semanticwebbuilder.org.mx/es_mx/swb/Sistemas_Embebidos_Innovando_hacia_los_Sistemas_Inteligentes_)
- Yagual, M. (26 de julio de 2016). *Conociendo tu Pc*. Obtenido de <http://todosobreensamblaje.blogspot.com/2013/12/procesador-microprocesador.html>

## 8. ANEXOS Y APÉNDICES

Pruebas del sistema en el usuario

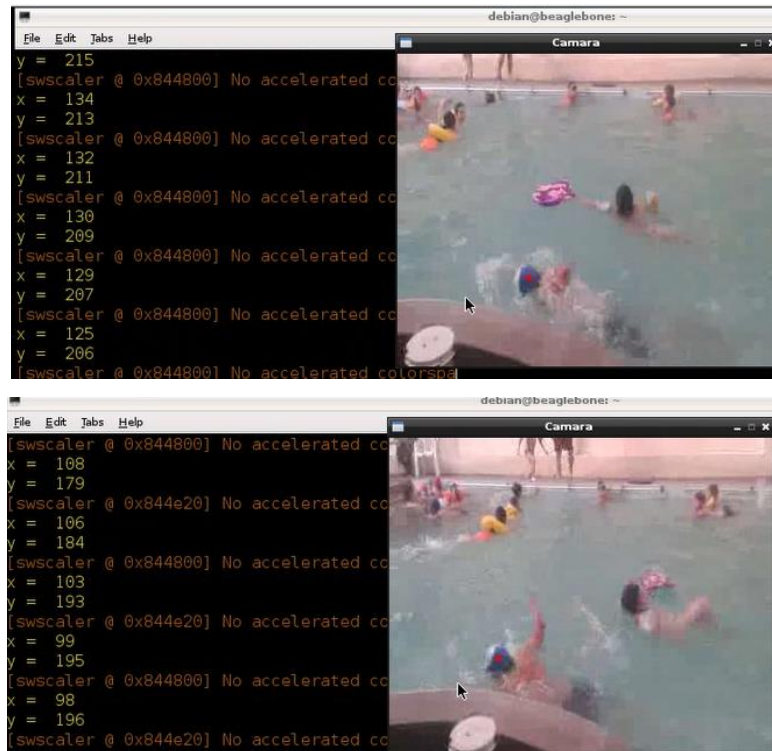


Figura 47.- Pruebas con obstáculos

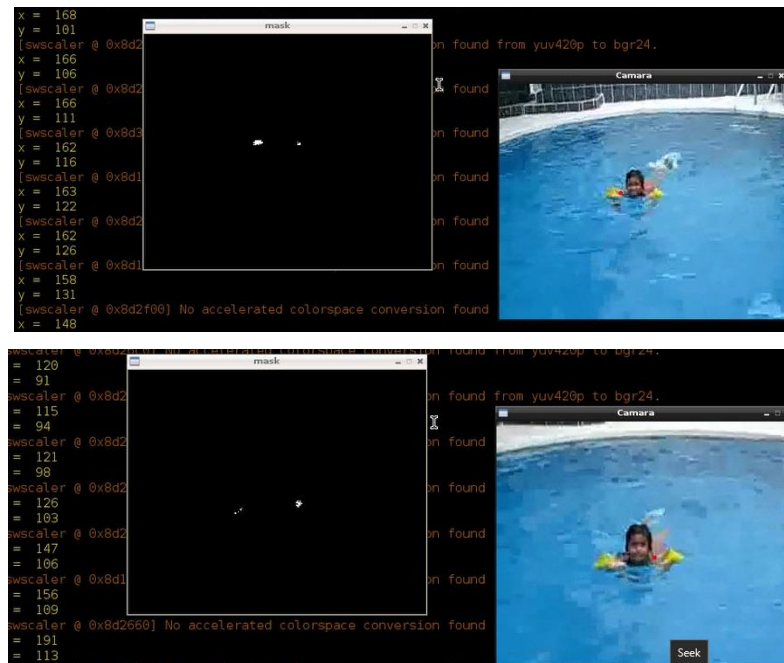


Figura 48.- Pruebas en piscina vacía



Código del Programa para procesamiento de imágenes

```
import cv2
import numpy as np
import math
#*****

# Para el mouse
draww = True
mode = False
ix,iy = -1,-1
m=[]
n=[]
c=0
x1,x2,x3 = 53,160,267
y1,y2,y3 = 40,120,200
p1 = x1,y1
p2 = x2,y1
p3 = x3,y1
p4 = x1,y2
p5 = x2,y2
p6 = x3,y2
p7 = x1,y3
p8 = x2,y3
p9 = x3,y3
# mouse callback function
def draw_circle(event,x,y,flags,param):
    global ix,iy,c,mode

    if event == cv2.EVENT_LBUTTONDOWN:
        mode = True
        m.append(x)
        n.append(y)
        ix,iy = x,y
```

```

    print "ix",ix
    print "iy",iy

if event == cv2.EVENT_LBUTTONDOWN:
    if mode == True:
        cv2.line(img,(ix,iy),(x,y),(0,255,0),5)
        m.append(x)
        n.append(y)
        c=c+1
        ix,iy = x,y
        print "x",x
        print "y",y

img = np.zeros((320,320,3),np.uint8)
cv2.namedWindow('image')
cv2.setMouseCallback('image',draw_circle)
cv2.circle(img,p1,5,(0,0,255),-1)
cv2.circle(img,p2,5,(0,0,255),-1)
cv2.circle(img,p3,5,(0,0,255),-1)
cv2.circle(img,p4,5,(0,0,255),-1)
cv2.circle(img,p5,5,(0,0,255),-1)
cv2.circle(img,p6,5,(0,0,255),-1)
cv2.circle(img,p7,5,(0,0,255),-1)
cv2.circle(img,p8,5,(0,0,255),-1)
cv2.circle(img,p9,5,(0,0,255),-1)
while(1):
    cv2.imshow('image',img)
    k = cv2.waitKey(1) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()

```

```

#.....
class Punto(object):
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def distancia(self, otro):
        dx = self.x - otro.x
        dy = self.y - otro.y
        return math.sqrt(dx**2 + dy**2)

    def pendiente(self, p):
        n = float(self.x - p.x)
        if n == 0:
            n=1
        d = float(self.y - p.y)
        return float(d/n)

def puntoY(m,x,x0,y0):
    if x == x0:
        m=0
        y0=x0
    y = m*(x-x0)+y0
    return int(y)

#*****
i=0
#*****

#Ancho y alto del video
frame_w = 320
frame_h = 240

#Coordenadas del centro del video

```

```

fcx = int(frame_w/2)
fcy = int(frame_h/2)
#Indices de las propiedades
FRAME_PROP_WIDTH = 3
FRAME_PROP_HEIGHT = 4
#Definir rango a identificar en HSV
lower_color = np.array([110,50,50], dtype=np.uint8)
upper_color = np.array([130,255,255], dtype=np.uint8)
#Iniciar captura de video
cap = cv2.VideoCapture(0)
cap.set(FRAME_PROP_WIDTH, frame_w)
cap.set(FRAME_PROP_HEIGHT, frame_h)
#Repetir mientras halla señal de video
while cap.isOpened():
    #Leer un frame
    _, img =cap.read()
    #Aplicar desenfoque para eliminar el ruido
    frame = cv2.blur(img, (15, 15))
    #Convertir frame de BGR a HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    #Aplicar umbral a la imagen y extraer los pixeles
    thresh = cv2.inRange(hsv, lower_color, upper_color)
    #Encontrar los contornos en la imagen extraida
    cnts, h = cv2.findContours(thresh, cv2.RETR_LIST,
cv2.CHAIN_APPROX_SIMPLE)
    cv2.circle(img,p1,5,(0,0,255),-1)
    cv2.circle(img,p2,5,(0,0,255),-1)
    cv2.circle(img,p3,5,(0,0,255),-1)
    cv2.circle(img,p4,5,(0,0,255),-1)
    cv2.circle(img,p5,5,(0,0,255),-1)
    cv2.circle(img,p6,5,(0,0,255),-1)
    cv2.circle(img,p7,5,(0,0,255),-1)

```

```

cv2.circle(img,p8,5,(0,0,255),-1)
cv2.circle(img,p9,5,(0,0,255),-1)

cv2.line(img,(m[0],n[0]),(m[1],n[1]),(0,255,0),2)
cv2.line(img,(m[1],n[1]),(m[2],n[2]),(0,255,0),2)
cv2.line(img,(m[2],n[2]),(m[3],n[3]),(0,255,0),2)
#Encontrar el contorno de mayor area y especificarlo como best_cnt
max_area = 0
for cnt in cnts:
    area = cv2.contourArea(cnt)
    if area > max_area:
        max_area = area
        best_cnt = cnt

#Ejecutar este bloque solo si se encontro un area
if max_area > 0:
    #Encontrar el centroide del mejor contorno
    M = cv2.moments(best_cnt)
    cx, cy = int(M['m10']/M['m00']), int(M['m01']/M['m00'])
    cv2.circle(img, (cx, cy), 5, 255, -1)
    #Dibujar un rectangulo alrededor del objeto
    (x,y),radius = cv2.minEnclosingCircle(best_cnt)
    center = (int(x),int(y))
    radius = int(radius)
    cv2.circle(img,center,radius,(255,0,0),2)
    #Calcular el error en X y Y y mostrarlo como texto
    errx = cx
    erry = cy
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(img, str(errx), (30, 30), font, 0.8, 255, 2, 8)
    cv2.putText(img, str(erry), (120, 30), font, 0.8, 255, 2, 8)
    P0 = Punto(cx,cy)

```

```

Q0 = Punto(m[i],n[i])
D0 = P0.distancia(Q0)
P1 = Punto(m[i],n[i])
Q1 = Punto(m[i+1],n[i+1])
D1 = P1.distancia(Q1)
M1 = P1.pendiente(Q1)
P2 = Punto(cx,cy)
Q2 = Punto(m[i+1],n[i+1])
D2 = P2.distancia(Q2)
Y = puntoY(M1,cx,m[i],n[i]) # Ecuacion obtener " y "
if (n[i] > n[i+1] and m[i] < m[i+1]):
    if cy > Y+10 :
        cv2.putText(img, 'izquierda', (70, 10), font, 0.8, 255, 2, 8)
    elif cy < Y-10 :
        cv2.putText(img, 'derecha', (70, 10), font, 0.8, 255, 2, 8)
    elif cy == Y :
        cv2.putText(img, 'frente', (70, 10), font, 0.8, 255, 2, 8)
    elif (cy <= Y+10 and cy >= Y-10) :
        cv2.putText(img, 'frente', (70, 10), font, 0.8, 255, 2, 8)

elif (n[i] > n[i+1] and m[i] > m[i+1]):
    if cy > Y+10 :
        cv2.putText(img, 'derecha', (70, 10), font, 0.8, 255, 2, 8)
    elif cy < Y-10 :
        cv2.putText(img, 'izquierda', (70, 10), font, 0.8, 255, 2, 8)
    elif cy == Y :
        cv2.putText(img, 'frente', (70, 10), font, 0.8, 255, 2, 8)
    elif (cy <= Y+10 and cy >= Y-10) :
        cv2.putText(img, 'frente', (70, 10), font, 0.8, 255, 2, 8)

```

```

elif (n[i] < n[i+1] and m[i] == m[i+1]):
    if Y > 0:
        cv2.putText(img, 'izquierda', (70, 10), font, 0.8, 255, 2, 8)
    elif Y < 0:
        cv2.putText(img, 'derecha', (70, 10), font, 0.8, 255, 2, 8)
    if Y == cx:
        cv2.putText(img, 'frente', (70, 10), font, 0.8, 255, 2, 8)

```

```

if D2 < 10 :
    i=i+1
    cv2.putText(img, str(int(D2)), (30, 100), font, 0.8, 255, 2, 8)
    cv2.putText(img, str(i), (100, 100), font, 0.8, 255, 2, 8)
    cv2.putText(img, str(M1), (150, 100), font, 0.8, 255, 2, 8)
    cv2.putText(img, str(Y), (150, 150), font, 0.8, 255, 2, 8)
    # Calculo de Angulo
    N=(D0**2)+(D1**2)-(D2**2)
    M=2*D0*D1
    if M == 0:
        M=1
    angle = math.acos(N/M)*180/math.pi
    print "Angulo",angle
    cv2.line(img, (cx, cy), (m[i], n[i]), 255, 2)

```

```

#mostrar la imagen original con todos los overlays
cv2.imshow('img', img)
#Mostrar la mascara con los pixeles extraidos
cv2.imshow('thresh', thresh)

```

```

#Salir del bucle cpn ESC

```

```
tecla = cv2.waitKey(1)
if tecla == 27:
    break

cv2.destroyAllWindows()
```