



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**

**FACULTAD DE INGENIERÍA**

**CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

“Trabajo de Grado previo a la obtención del Título de Ingeniero e Ingeniera en  
Electrónica y Telecomunicaciones”

**TRABAJO DE GRADUACIÓN:**

“DISEÑO Y CONSTRUCCION DE UN ROBOT PARA REALIZAR  
RUTINAS DE EJERCICIOS DE FISIOTERAPIA”

**AUTOR(ES):**

LUIS FERNANDO GUSQUI MAIGUA

BYRON DAVID VILEMA CHUIZA

**DIRECTOR:**

Ing. Fabián Gunsha

**AÑO:**

2016

## INFORME DEL TUTOR

Yo Ing. FABIÁN CELSO GUNSHA MAJI, en mi calidad de Tutor, del trabajo investigativo titulado: DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PARA REALIZAR RUTINAS DE EJERCICIOS DE FISIOTERAPIA, tengo a bien informar que el mencionado trabajo cumple con los requisitos exigidos para ser expuesto público, luego de ser evaluado por el tribunal designado.

Riobamba, Septiembre del 2016

Atentamente:



Ing. Fabián Gunsha

TUTOR

Los miembros del tribunal de graduación del proyecto de investigación: DISEÑO Y CONSTRUCCIÓN DE UN ROBOT PARA REALIZAR RUTINAS DE EJERCICIOS DE FISIOTERAPIA, presentado por Luis Fernando Gusqui Maigua, Byron David Vilema Chuiza y dirigida por el Ing. Fabián Gunsha Maji.

Una vez escuchada la defensa oral y revisada 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 la presente por uso y custodia en la Biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

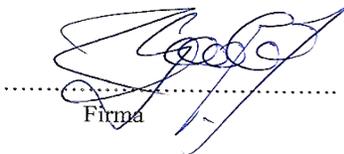
Ing. Paulina Vélez  
Presidenta del Tribunal

  
Firma

Ing. Fabián Gunsha  
Director del Proyecto

  
Firma

Ing. Giovanni Cuzco  
Miembro del Tribunal

  
Firma

## AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Luis Fernando Gusqui Maigua, Byron David Vilema Chuiza y del Director del proyecto: Ing. Fabián Gunsha; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

Autores:

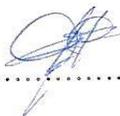


Luis Fernando Gusqui Maigua



Byron David Vilema Chuiza

Director del Proyecto:



Ing. Fabián Gunsha.

## *Agradecimiento I*

*Agradezco a Dios por haberme guiado por el camino de la felicidad, por cada minuto de vida y compañía, a cada uno de los que son parte de mi familia, a mi madre, mi amiga y mi compañera que siempre hemos saliendo de los buenos y malos momentos, a mi hermana que desde niños salimos por sí solos adelante.*

*A mis primos Javi, Maykel, Mony, Joel, Ximena y Yadira que siempre estamos con aventuras y sueños grandes por cumplir.*

*Al Ingeniero Fabián Gunsha, por sus conocimientos compartidos en este proyecto de titulación.*

*A Byron por ser un gran amigo y compañero, en el desarrollo de este proyecto de titulación.*

*A la Universidad Nacional de Chimborazo por los conocimientos y aprendizajes adquiridos en mi carrera universitaria, a mis compañeros de clases, amigos de barrio quienes compartí grandes momentos.*

*Luis Fernando Gusquí*

## *Dedicatoria I*

*A Dios y la Virgencita del Cisne, por regresarme mi salud y a cumplir este sueño que tuve desde hace mucho tiempo.*

*Con mucho amor y cariño a mi mamá Rosa y mi hermana Verónica que los tres somos el amor, la confianza y la alegría de nuestro hogar.*

*A mi abuelita María, que siempre que estado solo, ella ha sido mi confidente y mi madre cuando mi mamá está lejos.*

*A mis abuelitos Vicente y Angelita que son mi motivo de lucha para salir adelante.*

*A mis tíos que han sido mis hermanos y fieles amigos que siempre están apoyándome Viche, Nelson, Sandra, Jorge, Alberto, Anita, Margarita y Hugo que siempre les llevare en mi corazón.*

*A mis tíos Luis y Martha que han sido mis segundos padres y nunca me dejaron solo en los momentos más difíciles de mi vida.*

*Luis Fernando Gusquí*

## ***Agradecimiento II***

*Agradezco de la manera más sincera a la Universidad Nacional de Chimborazo, a todos los docentes de la Carrera de Electrónica y Telecomunicaciones, tutor de tesis, amigos, familiares, a mis padres que han hecho todo lo posible para culminar esta meta trazada en mi vida, a todas las personas que con sus experiencias han logrado hacer de mí una mejor persona.*

***David Vilema***

## ***Dedicatoria II***

*A Dios, por brindarme salud para superar todas las adversidades que se presentan en la vida, a mi madre Rosa por el apoyo incondicional, a mi padre Ángel por los consejos impartidos en esta etapa de mi vida, para mi pequeña Leslie Nicole, para Laura, para toda mi familia en especial para mi hermana Nancy ya que sin el apoyo de ella esto no hubiese sido posible.*

***David Vilema***

# ÍNDICE GENERAL

## CONTENIDO

ÍNDICE GENERAL.....	9
ÍNDICE DE TABLAS .....	xv
RESUMEN.....	xvi
INTRODUCCIÓN .....	xvii
CAPÍTULO 1 .....	2
1. FUNDAMENTACIÓN TEÓRICA.....	2
1.1. Raspberry pi 2 .....	2
1.1.1. Evolución de Raspberry Pi.....	2
1.1.2. Software .....	4
1.1.2.1. Sistema Operativo Raspbian Wheezy .....	5
1.1.2.2. Sistema Operativo Openelec .....	6
1.1.2.3. Sistema Operativo Risc Os.....	6
1.1.2.4. Sistema Operativo Pidora.....	6
1.1.2.5. Sistema Operativo Windows 10 Pi 2 .....	6
1.1.3. Hardware.....	7
1.1.3.1. Componentes Básicos .....	7
▪ Conector de alimentación.....	7
▪ Puerto HDMI (High Definition Multimedia Interface).....	7
▪ Conector de audio.....	8
▪ Puerto Ethernet.....	8
▪ Cable FFC (15 pines) .....	8
▪ Puerto USB.....	8
▪ GPIO .....	9
▪ Diodos emisores de luz (LEDs).....	9
▪ Ranura para tarjeta SD .....	9
▪ Conector de vídeo (RCA).....	9
1.1.4. Funcionamiento de los pines GPIO .....	9
1.1.5. Cámara Pi.....	11

1.1.6.	Adaptador USB Wi Fi .....	12
1.2.	Controlador de servo Motores – Torobot .....	13
1.2.1.	Especificaciones .....	13
1.2.2.	Software del controlador de Servos.....	15
1.3.	Servo Motor .....	16
1.3.1.	Características .....	16
1.3.2.	Funcionamiento de un Servo Motor .....	17
1.3.3.	Partes De Un Servo Motor .....	18
1.4.	Matlab .....	18
1.4.1.	Programación de Raspberry Pi con Matlab.....	19
1.5.	Python .....	22
1.5.1.	IDLE, el intérprete de Python de Raspberry Pi .....	23
1.6.	Fisioterapia .....	24
CAPÍTULO 2 .....		31
2.	METODOLOGÍA .....	31
2.1.	Tipo de Estudio .....	31
2.1.1.	Descriptivo .....	31
2.2.	Métodos, Técnicas e Instrumentos .....	31
2.2.1.	Métodos.....	31
2.2.1.1.	Analítico/ Deductivo .....	31
2.2.2.	Técnicas.....	31
2.2.2.1.	Observación.....	31
2.2.3.	Instrumentos .....	31
2.3.	Población y Muestra.....	31
2.4.	Hipótesis.....	32
2.5.	Operacionalización de Variables.....	32
2.6.	Procedimientos .....	33
Diagrama de Funcionamiento .....		33
2.6.1.	Diseño de Robot .....	34
2.6.1.1.	Diseño de Robot en forma de Humanoide .....	34
2.6.1.2.	Construcción de Robot .....	34
2.6.1.3.	Implementación de Circuito Electrónico.....	35

2.6.2.	Configuración y Programación de Raspberry Pi 2 .....	36
2.6.2.1.	Conexión Remota PC – Raspberry Pi .....	37
2.6.2.2.	Actualización de Comandos y Librerías en Raspberry Pi .....	37
2.6.2.3.	Programación en Python para ejecutar ejercicios de Fisioterapia .....	38
2.6.3.	Programación en Matlab .....	40
2.6.3.1.	Programación en Matlab – Simulink.....	40
2.6.3.2.	Programación en Matlab – Guide para Procesamiento de Imágenes. ....	41
2.7.	Comprobación de Hipótesis .....	46
2.7.1.	Técnicas de Procedimiento para el Análisis .....	46
2.7.2.	Comprobación de Hipótesis .....	46
2.7.3.	Elección de la Prueba de Hipótesis .....	46
2.7.4.	Muestra.....	49
CAPÍTULO III.....		50
3.	RESULTADOS.....	50
3.1.	Resultados de la funcionalidad del sistema y software desarrollado.....	50
3.2.	Análisis Financiero.....	54
CAPÍTULO IV.....		55
4.	CONCLUSIONES Y RECOMENDACIONES .....	55
4.1.	Conclusiones .....	55
4.2.	Recomendaciones.....	56
CAPÍTULO V .....		57
5.	PROPUESTA.....	57
5.1.	Título de la Propuesta.....	57
5.2.	Introducción .....	57
5.3.	Discusión.....	57
CAPÍTULO VI.....		59
6.	OBJETIVOS .....	59
6.1.	Objetivo General .....	59
6.2.	Objetivos Específicos.....	59
6.3.	Fundamentación Científico – Técnica.....	59
6.4.	Descripción de la Propuesta .....	59
6.5.	Diseño Organizacional.....	60

CAPITULO VII .....	61
7. BIBLIOGRAFÍA.....	61
CAPITULO VIII .....	63
8. ANEXOS.....	63
ANEXO 1.....	63
Diseño de robot en forma de Humanoide.....	63
ANEXO 2.....	64
Construcción de robot .....	64
ANEXO 3.....	65
Implementación y Conexión de Servos, Rasperry y Baterías .....	65
ANEXO 4.....	66
Configuración y Programación de Rasperry Pi 2.....	66
ANEXO 5.....	75
Conexión Remota PC – Rasperry Pi .....	75
ANEXO 6.....	78
Programación en Python para ejecutar ejercicios de Fisioterapia.....	78
ANEXO 7.....	81
Programación en Matlab – Simulink.....	81
ANEXO 8.....	88
Programación en Matlab – Guide para el Análisis de Imágenes.....	88
ANEXO 9.....	99
Datasheet Rasperry Pi 2 .....	99
ANEXO 10.....	100
Datasheet Rasperry Pi Camera v2.0.....	100
ANEXO 11.....	101
Datashet de Controlador de Servo Motor Torobot.....	101
ANEXO 12.....	108
Servo Motor RDS 3115MG .....	108

# ÍNDICE DE FIGURAS

Figura 1 Modelos lanzados de Raspberry Pi .....	3
Figura 2 Arquitectura de software Raspberry Pi .....	4
Figura 3 Sistemas Operativos para Raspberry Pi .....	5
Figura 4 Raspbian Wheezy - Sistema Operativo.....	5
Figura 5 Componentes básicos mostrados en Raspberry Pi 2 .....	7
Figura 6 Conectores para Cámara Pi y PiTFT .....	8
Figura 7 Raspberry Pi Diagrama de GPIO .....	9
Figura 8 Comunicación UART .....	10
Figura 9 Conexión de SPI .....	11
Figura 10 Cámara para Raspberry Pi 2 .....	12
Figura 11 Adaptador USB Wi-Fi .....	12
Figura 12 Controlador de 32 Servo Motores.....	13
Figura 13 Software Torobot .....	15
Figura 14 Terminales de un Servo Motor .....	16
Figura 15 Funcionamiento de un Servo Motor .....	17
Figura 16 Partes de un Servo Motor.....	18
Figura 17 Logo Matlab 2015.....	19
Figura 18 Interacción Matlab - Raspberry Pi .....	19
Figura 19 Herramientas de Raspberry Pi Matlab - Simulink .....	20
Figura 20 Datos de Raspberry Pi en Matlab.....	20
Figura 21 Logo Python.....	22
Figura 22 New Model – Python .....	23
Figura 23 Tipos de Datos – Python .....	23
Figura 24 Ejercicios de Fisioterapia.....	24
Figura 25 Ejercicio Sentadilla .....	25
Figura 26 Ejercicio Flexión de Pie.....	26
Figura 27 Flexión de Hombro .....	27
Figura 28 Ejercicio de Abducción.....	28
Figura 29 Ejercicio de Flexión de Brazos .....	29
Figura 30 Ejercicio Extensión de Brazos .....	30
Figura 31 Diagrama de Procedimientos .....	33
Figura 32 Diagrama de Conexión .....	34
Figura 33 Modelo a Construir .....	35
Figura 34 Diagrama de Conexión .....	35
Figura 35 Actualización de Librerías y Paquetes .....	36
Figura 36 Conexión Remota de PC.....	37
Figura 37 Diagrama de Flujo - Programación en Python.....	38
Figura 38 Programación en Matlab - Simulink.....	40
Figura 39 Programación Matlab – Guide .....	41
Figura 40 Diagrama de Flujo - Procesamiento de Imágenes.....	43

Figura 41 Análisis de Resultados .....	45
Figura 42 Tabla T- Student .....	48
Figura 43 Sistema Desarrollado para Aplicaciones de Ejecución.....	50
Figura 44 Resultados del Primer día .....	51
Figura 45 Resultados del segundo día.....	52
Figura 46 Resultados del tercer día .....	53
Figura 47 Resultados Esperados .....	53

## ÍNDICE DE TABLAS

Tabla 1 Las principales características de los modelos lanzados de Raspberry Pi.....	3
Tabla 2 Características de Camera Pi V2 .....	11
Tabla 3 Características de Adaptador USB WIFI .....	12
Tabla 4 Sintaxis para Controlar Servo Motores .....	14
Tabla 5 Características de Servo Motor .....	16
Tabla 6 Funciones de Matlab con Raspberry .....	21
Tabla 7 Operacionalización de Variables.....	32
Tabla 8 Resultados de Ejercicios de Fisioterapia.....	47
Tabla 9 Frecuencias Esperadas .....	47
Tabla 10 Grado de Libertad .....	47
Tabla 11 Cálculos de Chi Cuadrado.....	48
Tabla 12 Análisis Financiero.....	54

## RESUMEN

El presente trabajo de investigación muestra el diseño y construcción de un robot en forma de humanoide que permita el monitoreo y evaluación de ejercicios de rehabilitación para fisioterapia mediante una cámara y una red Wi-fi.

El robot posee 10 grados de libertad y está construido con servos metálicos de 15 Kg marca RDS3115MG, un controlador de servo motores de marca Torobot, pilas LifePo 3.2V, piezas de aluminio ensambladas en forma de humanoide que permite realizar diferentes tipos de ejercicios de rehabilitación como zancadillas, flexión de piernas, aducción y flexión de brazos-hombros, que están programadas en la tarjeta Raspberry Pi 2 modelo B en el software Python 2.7 y controlados de manera remota mediante una aplicación en Matlab.

El monitoreo y captura de imagen se realiza por una cámara de video ubicada en la cabeza del robot, y mediante una red Wi-fi se envían los datos a una Laptop en el que se encuentra un programa desarrollado en Matlab-Guide que permite visualizar al paciente vía “Online”, para verificar si realiza la rutina asignada.

Para diagnosticar y determinar si el ejercicio es correcto o incorrecto, se realiza la captura de una secuencia de 12 imágenes que son almacenadas en una carpeta, de tal manera que cada vez que el fisioterapeuta necesite verificar, solo se pulsa un botón de análisis en la aplicación que permite capturar una imagen e inmediatamente realiza una correlación con las imágenes almacenadas en escala de grises, para obtener un valor entre 0 y 1, con varias pruebas realizadas se determinó que con un valores superiores al 0.5 se puede verificar el mejoramiento del paciente al aumentar este valor hasta llegar a 1, de tal manera que se puede hacer una evaluación constante por cada rutina realizada y a través de curvas se puede valorar el avance en la rehabilitación.

## ABSTRACT

This research shows the design and construction of a humanoid robot shaped to allow the monitoring and evaluation of rehabilitation physiotherapy exercises using a camera and a Wi-Fi network.

The robot has 10 degrees of freedom and is built with metal servos 15kg RDS3115MG brand, a servo controller Torobot brand engines, LiFePo 3.2V batteries, aluminum parts assembled in the form of humanoid that allows different types of rehabilitation exercises as tripping, leg flexion, adduction and flexion of arms-shoulders, which are scheduled in the Raspberry Pi 2 model B card in Python 2.7 software and controlled remotely through an application in Matlab.

Monitoring and image capture is performed by a video camera located in the robot's head, and by a Wi-fi data is sent to a Laptop in which is a program developed in Matlab-Guide that displays the patient via "Online" to check performs of the assigned routine.

To diagnose and determine if the exercise is right or wrong, capturing a sequence of 12 images that are stored in a folder, so that each time the physiotherapist needs to verify, only one button analysis is click performed the application to capture an image and immediately makes a correlation with the images stored in grayscale to obtain a value between 0 and 1, with several tests it was determined that with values above 0.5 can be verified improving patient increase this value up to 1 so that you can make a constant assessment for each routine performed and through curves can assess progress in rehabilitation.



Reviewed By: Geovanna Vallejo

ENGLISH LANGUAGE CENTER



## INTRODUCCIÓN

En actualidad la rehabilitación de enfermos y accidentados, necesitan terapias que a veces son repetitivas y los pacientes deben realizar un gran esfuerzo y también el fisioterapeuta.

La rehabilitación con aparatos robóticos es un área emergente que ofrece múltiples ventajas tales como la objetividad, repetitividad, homogeneidad de los tratamientos, autonomía, interacción y la posibilidad de terapias más intensas y personalizadas.

Con el avance de la tecnología, los robots de rehabilitación se han convertido en una ayuda para el fisioterapeuta que ahora sólo se debe encargarse del seguimiento y evolución del paciente, para dedicar mayor tiempo a conseguir terapias más eficientes,

El elevado costo de los sistemas que existe en el mercado y la dependencia de una participación directa de profesionales en el área, surge la necesidad de realizar el diseño y construcción de un robot humanoide para realizar rutinas de ejercicios de fisioterapia controlado por una tarjeta Raspberry PI que muestre al paciente la rutina que se debe realizar, y además que permita el monitoreo y evaluación del ejercicio realizado a través de un programa en Matlab-Guide, que determina el avance de la rehabilitación

# CAPÍTULO 1

## 1. FUNDAMENTACIÓN TEÓRICA

### 1.1. Raspberry pi 2

Raspberry Pi, es un ordenador en una placa del tamaño de una tarjeta de crédito, fue creado en el año 2006 por un grupo de académicos e ingenieros de Universidad de Cambridge, que permita crear un ambiente de aprendizaje para la programación al alcance de todos debido al bajo costo.



**Figura 1** Raspberry Pi 2

**Fuente:** <https://www.raspberrypi.org/magpi-issues>

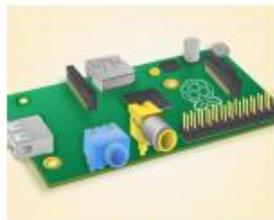
#### 1.1.1. Evolución de Raspberry Pi

Raspberry pi evoluciona en 5 modelos en 2 generaciones las mismas que se presenta en la Tabla 1. Sus componentes y características son eficientes para los usuarios debido a que poseen más almacenamiento y facilidad de aplicaciones para su evolución. La generación uno posee cuatro modelos de Raspberry Pi A, B, A+, B+ y en la segunda generación posee un solo modelo que es el Raspberry Pi modelo B conocido comercialmente Raspberry pi 3.

La evolución final en Raspberry viene con renovaciones basadas en las peticiones de los usuarios, se modernizara como la inclusión de puertos USB extras, la transición de memorias SD a microSD, el consumo de electricidad y en el audio que tendrá una fuente propia de energía.

**Tabla 1 Las principales características de los modelos lanzados de Raspberry Pi**  
**Fuente: Autores**

Raspberry Pi	Generación 1				Generación 2
	Modelo A	Modelo B	Modelo A+	Modelo B+	Modelo B
Especificaciones					
Power	300mA	700mA	200mA	600mA	900mA
Ethernet Port	No	Yes	No	Yes	Yes
USB Port	1	2	1	4	4
GPIO	26	36	40	40	40
SD Card Slot	SD	SD	Micro SD	Micro SD	Micro SD
SoC	BCM2835	BCM 2835	BCM2835	BCM2835	BCM2836
CPU	700MHz ARM11 Single-Core	700MHz ARM11 Single-Core	700MHz ARM11 Single-Core	700MHz ARM11 Single-Core	900MHz ARM Cortex-A7 quad -Core
RAM	256MB	512MB	256MB	512MB	1GB



(a) Raspberry Pi model A



(b) Raspberry Pi model B



(c) Raspberry Pi model B+



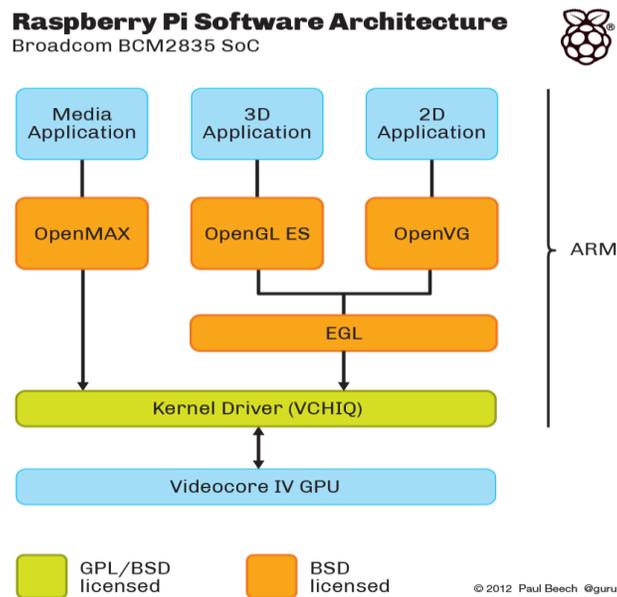
(d) Raspberry Pi model A+

**Figura 2** Modelos lanzados de Raspberry Pi

**Fuente:** <https://www.intorobotics.com/raspberry-pi-a-plus-get-less-pursue-similarities-b/>

### 1.1.2. Software

El Raspberry Pi utiliza sistemas operativos basados en el núcleo Linux, con una ARM que posee Unidad de procesamiento gráfico (GPU) accede mediante una imagen del firmware de código cerrado, que se carga dentro de la GPU al arrancar desde la tarjeta SD. El blob binario está asociado a los drivers Linux que también son de código cerrado. Las aplicaciones hacen llamadas a las librerías de tiempo de ejecución que son de código abierto, y estas librerías hacen llamadas a unos drivers de código abierto en el kernel de Linux. La API (Interfaz de Programación de Aplicaciones) del driver del kernel es específica para estas librerías.



**Figura 3** Arquitectura de software Raspberry Pi

**Fuente:** <http://www.genbetadev.com/software-libre-y-licencias/la-gpu-de-la-raspberry-pi-ya-es-codigo-abierto>

Las aplicaciones que usan vídeo hacen uso de OpenMAX, las aplicaciones 3D usan OpenGL ES y las aplicaciones 2D usan OpenVG; OpenGL ES y OpenVG hacen uso de EGL y éste último, del driver de código abierto del kernel.

Los sistemas operativos que fueron desarrollados hasta la actualidad son:



**Figura 4** Sistemas Operativos para Raspberry Pi  
**Fuente:** Autores

### 1.1.2.1. Sistema Operativo Raspbian Wheezy

Es un sistema operativo libre basado en Debian optimizado para el hardware de Raspberry Pi. El sistema operativo es el conjunto de programas básicos y utilidades que hacen que funcione Raspberry Pi. Sin embargo, Raspbian ofrece más que un Sistema Operativo puro, viene con más de 35.000 paquetes, software precompilado en un formato que hace más fácil la instalación en su Raspberry Pi.



**Figura 5** Raspbian Wheezy - Sistema Operativo  
**Fuente:** Autores

Sus aplicaciones cargadas en el Sistema Operativo Wheezy son:

- Aplicación Java
- Aplicación Python
- Aplicaciones de Office
- Accesorios
- Herramientas Juegos etc.

### **1.1.2.2.Sistema Operativo Openelec**

Openelec (Open Embedded Linux Entertainment Center) es una distribución muy ligera con la que implementa una Media Center sobre nuestra Raspberry Pi, teniendo en cuenta que posee una salida HDMI, para conectarla a un televisor para usar como reproductor multimedia de bajo costo con una distribución especialmente optimizada para este tipo de tareas y con un interfaz orientado también a hacer sencillo su manejo.

### **1.1.2.3.Sistema Operativo Risc Os**

Es un Sistema Operativo desarrollado por una compañía británica llamada Acorn Computers para un chip ARM de 32 bits basado en Acorn Archimedes. Fue lanzado en 1987 bajo el nombre **Risc OS 2.0**. Acorn Computers cerró sus puertas hace ya bastante tiempo.

### **1.1.2.4.Sistema Operativo Pidora**

Pidora es una mezcla con Fedora, una versión personalizada del sistema Fedora tipo Unix, que se ejecuta en el ordenado Raspberry Pi, es bastante rápido en su ejecución, tiene una larga historia y se caracteriza por su estabilidad, existen miles de paquetes disponibles en el repositorio para Pidora, se encuentra cualquier funcionalidad o servicio que necesita para cualquier proyecto.

### **1.1.2.5.Sistema Operativo Windows 10 Pi 2**

El centro de desarrolladores de Microsoft ha anunciado la compatibilidad de Windows 10 con la Raspberry Pi 2. Al instalar Windows 10 que está diseñada para instalar pequeñas aplicaciones que conecten dispositivos. Esta edición será gratuita y no será tan completa como la edición que puedes añadir a tu PC.



- **Conector de audio**

Raspberry Pi 2 implementa el sonido Inter-Ic (I2S) para la entrada y salida de audio en bus que está conectado en serie. Esto permite que el dispositivo se conecte varios aparatos de audio digital. Un conector de 3,5mm TRRS (tip-ring-ring-sleeve) está disponible y compartido con la salida de vídeo analógico. El componente HDMI también ofrece salida de audio digital

- **Puerto Ethernet**

Es un conector de red que proporciona una velocidad de 0/100 Mbit/s a través de un cable RJ45 de red de área local (LAN), está disponible en el modelo B, B+ y Raspberry Pi modelo 2B.

- **Cable FFC (15 pines)**

Los cables FFC (flexible flat cable) tienen un conector de 15 pines. Se utilizan para poder tener acceso a los puertos DSI (Display Serial Interface) y para el puerto CSI-2 (Camera Serial Interface 2) Estos puertos permiten una mejor transferencia de la señal de video según sean utilizados.



a. Conector DSI Y CSI



b. Camera Pi y PiTFT

**Figura 7** Conectores para Cámara Pi y PiTFT

**Fuente:** <http://www.electronicaestudio.com/raspberrypi.htm>

- **Puerto USB**

Compatible con cualquier dispositivo USB, como un teclado, ratón, red Wi-Fi, cámara web. El puerto USB de versión 2.0, el modelo A y A+ tiene un puerto USB, mientras que el modelo B tiene dos puertos USB, existen cuatro puertos USB en el modelo B+ y Raspberry Pi 2 modelo B.

- **GPIO**

Es un conjunto de pines de entrada y salida para propósitos generales, tales como la conexión de tarjetas de expansión o controlar LEDs, motores y otros tipos de dispositivos electrónicos, existen 26 pines en el modelo A y B y 40 pines en el modelo A+, B+ y Raspberry Pi 2 modelo B.

- **Diodos emisores de luz (LEDs)**

Indican el estado de la tarjeta, el LED rojo aparece cuando se enciende la tarjeta y el LED de color amarillo indica cuando se conecta a la red, en los modelos A y B proporcionan 5 LEDs en el que los dos primeros son para indicar el estado de tarjetas SD y el estado de la alimentación y los otros tres son para un estado de la red.

- **Ranura para tarjeta SD**

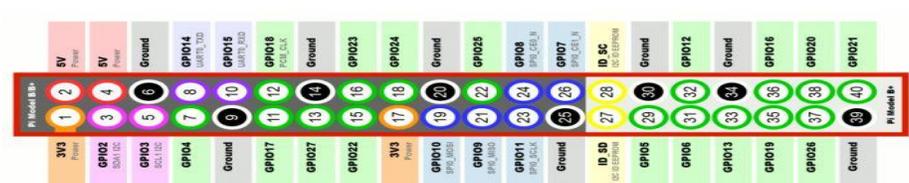
Es para insertar la tarjeta de memoria SD para almacenar un sistema operativo deseado o un programa, soporta hasta 32 GB de almacenamiento hasta con 10 velocidades. La ranura para tarjetas SD se ha reemplazado con una ranura para tarjeta micro SD en el modelo A+, B+ y Raspberry Pi 2 el modelo B.

- **Conector de vídeo (RCA)**

Este conector RCA para Raspberry Pi de 3.5mm permite conectar un cable para videocámara para transportar señales de audio y vídeo,

### 1.1.4. Funcionamiento de los pines GPIO

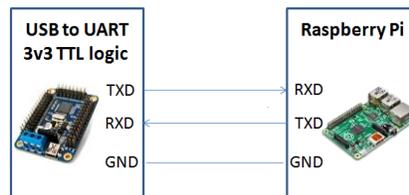
GPIO es un pin genérico y reutilizable que puede ser configurado con un valor lógico 1 (3.3V) y 0 (GND), para actuar como una entrada o salida de acuerdo al propósito del usuario, para evitar daños en el chip de un voltaje suministrada a los pines GPIO, la tensión debe estar en un rango aceptable en función de la especificación de cada chip.



**Figura 8** Raspberry Pi Diagrama de GPIO  
**Fuente:** <http://gadoinfouned.blogspot.com/>

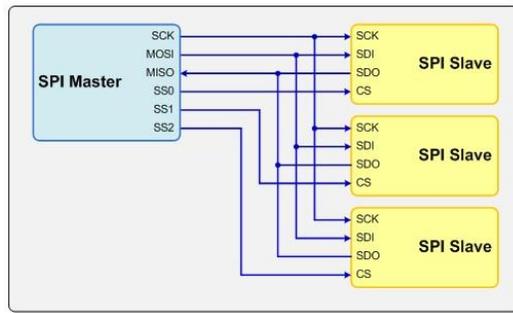
El diagrama de pines de la Raspberry Pi GPIO se muestra en la figura 7, los primeros 26 pines son comunes para todos los modelos y los 14 pines restantes sólo están disponibles en el modelo A+, B+ y Raspberry Pi 2 modelo B. También podemos encontrar en algunos GPIO otras funciones como (UART) Transmisión y Recepción Universal Asíncrona, (SPI) Interfaz Serial Periférico, (I2C) Circuito Inter-integrado.

- **Power.**- Los colores naranja y rojo de la figura 7, proporciona 4 pines para 2 niveles, 3.3V (pin 1) y 5V (pin 2), se considera como tensión de alimentación de la placa, los pines 2 y 4 permiten conectar con otros dispositivos que requieren fuente de alimentación 5V.
- **UART.**- Es una conexión en serie que permite a dos dispositivos para el intercambio de datos, proporciona los pines 8 (TxD) y 10 (RxD) para la transmisión y recepción de la figura 7 de color púrpura.



**Figura 9** Comunicación UART  
Fuente: Autores

- **I2C.**- Es un estándar de comunicación muy utilizado, diseñado para comunicarse con otros chips como ADC, DAC, memorias etc, cada uno con una dirección única, que puede a menudo configurarse mediante jumpers, los pines para la comunicación son 3 y 5 del color rosa de la figura 7.
- **SPI.**- Es un bus de interfaz utilizada para enviar datos entre los microcontroladores y pequeños periféricos tales como registros de desplazamiento, sensores y tarjetas SD. Utiliza líneas de reloj y datos separados, además de una línea de selección para elegir el dispositivo con el que desea comunicarse, los pines 19, 21, 23, 24 y 26 sirven para este propósito y están de color azul en la figura 7.



**Figura 10** Conexión de SPI  
**Fuente:** <http://comohacer.eu/gpio-raspberry-pi/>

Los pines necesitan ser configurados a través de un lenguaje de programación como Python, C o por medio de un software de paquetes de soporte tales como Matlab, para realizar aplicaciones

### 1.1.5. Cámara Pi

La Raspberry Pi Camera, podemos conectar vía USB, existen librerías para su control y funcionamiento, es recomendable instalar los sistemas operativos más recientes distribuidos por Raspberry Pi que permite realizar multitud de cambios como color, calidad o brillo. En la Tabla 2 podemos ver las principales características.

**Tabla 2** Características de Camera Pi V2  
**Fuente:** Autores

Características	Valor
Dimensiones	25x20x9mm
Peso	3g
Resolución	5 Megapíxeles
Modos Video	1080p30,720p60,VGA90
Resolución Imagen	2592x1944 pixel



**Figura 11** Cámara para Raspberry Pi 2  
**Fuente:** <https://www.raspberrypi.org/magpi/get-started-pi-camera/>

### 1.1.6. Adaptador USB Wi Fi

Para la conexión a internet en la tarjeta Raspberry Pi se puede conectar a través del conector RJ45 con un cable Ethernet, o darle mayor movilidad al sistema y ahorrar conexiones de más cables, con una conexión inalámbrica a la red. El adaptador TL-WN725N de TP-LINK con el estándar Wi-Fi de la IEEE 802.11 se pueden crear redes de área local inalámbricas de alta velocidad siempre y cuando el equipo que se vaya a conectar no esté muy alejado del punto de acceso.

**Tabla 3** Características de Adaptador USB WIFI  
**Fuente:** [www.raspberrypi.org/adaptadorwifi2345](http://www.raspberrypi.org/adaptadorwifi2345)

Especificaciones	Valor
Estándar	IEEE 802,11b/g/n
Frecuencia	2.4 – 2.4835 GHz
Velocidad	150 Mbps
Dimensiones	18.6x15x7.1mm
Seguridad	WEP, WPA/WPA2, WPA-PSK/WPA2-PSK



**Figura 12** Adaptador USB Wi-Fi  
**Fuente:** <https://www.raspberrypi.org/conexionwifi>

## 1.2. Controlador de servo Motores – Torobot

Permite el manejo de 32 servomotores al mismo tiempo, soporta 3 métodos de control, conexión directa por interfaz USB y el computador, TTL serial para usar con sistemas o tarjetas de control (por ejemplo Arduino y Microcontroladores) o usando un lenguaje scripting que trae interno. El circuito tiene un rango de trabajo de 180 grados con una salida PWM que va desde 0.5 ms hasta 2.5 ms.



**Figura 13** Controlador de 32 Servo Motores

**Fuente:** <https://hetpro-store.com/controlador-servomotores-32ch-torobot-usb/>

### 1.2.1. Especificaciones

- Servos fuente de alimentación: DC (depende de motores, generalmente 5V DC)
- Canales de control: 32
- Comunicación de entrada: USB o UART (TTL o USART)
- Salida de la señal: PWM (Precisión 1us)
- Servo resolución motor: 1us, 0,09 grados
- Baud Rate: 9600, 19200, 28400, 57600, 115200, 128000
- Dimensiones: 63,5 mm x 43,5 mm x 1,6 mm
- Modo de control: USB y UART (TTL) para aceptar comandos.
- Memoria Flash: 512K ROM

Los comandos para enviarle mensajes al controlador vienen definidos en la Tabla 4, donde se explica cómo controlar uno o varios servos al mismo tiempo, y el significado de cada parte del comando que se envía y sus modos de control son USB y UART (TTL) para

aceptar comandos, el software de control de PC con velocidades de transmisión: 9600,19200, 28400,57600, 115200,128000 (reconocimiento automático).

**Tabla 4** Sintaxis para Controlar Servo Motores  
**Fuente:** [http://letsmakerobots.com/files/32\\_Servo\\_Controller\\_Manual.pdf](http://letsmakerobots.com/files/32_Servo_Controller_Manual.pdf)

Nombre	Comando	Descripción
Controlar un único servo	#1P1500T100\r\n	<p>El dato 1 hace referencia al canal del servo.</p> <p>El dato 1500 hace referencia a la posición del servo, con un rango de 500-2500.</p> <p>El dato 100 hace referencia al tiempo de ejecución y representa la velocidad, con un rango de 100-9999</p>
Controlar Varios Servos	#1P600#2P900#8P2500T100\r\n	<p>Los datos 1, 2, y 8 hacen referencia a los canales de los servos.</p> <p>Los datos 600, 900, y 2500 hacen referencia a la posición de los servos y los sitúa a los tres.</p> <p>El dato 100 hace referencia al tiempo de ejecución y representa la velocidad de los tres servos. De acuerdo al número de los servos hay solo un tiempo o una T.</p> <p>El comando se ejecuta al mismo tiempo, es decir, todos los servos operan simultáneamente.</p>
Ejecutar una única sección	#1GC2\r\n	El dato 1 se refiere al número/nombre del grupo de

		acción. El dato <b>2</b> se refiere al número de ciclos.
Ejecutar varias acciones	<code>#1G#3G#1GC2\r\n</code>	Ejecuta el <b>primer</b> , <b>tercer</b> y <b>primer</b> grupo de acción. El número de ciclos es <b>2</b> . Un grupo en particular puede aparecer repetidamente. Solo puede haber un único número de ciclos o C. El comando es ejecutado en secuencia, esto significa que los grupos de acciones se ejecutan en secuencia.

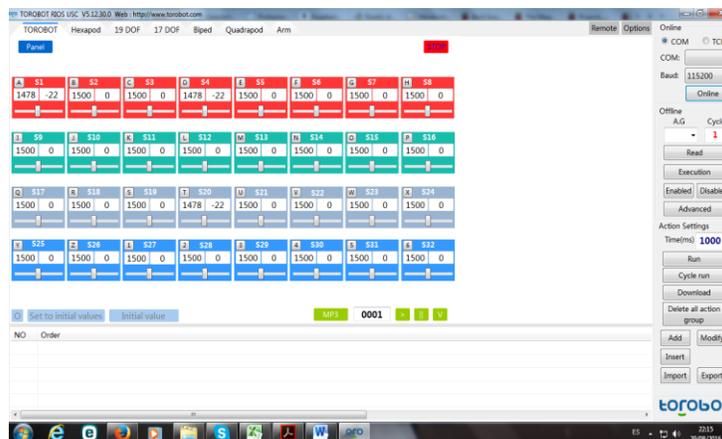
**Ejemplo:**

`# 8P600T1000 \ n \ r`

Mover servo 8 a pulso 600us anchura, usando el tiempo de 1000 microsegundos

**1.2.2. Software del controlador de Servos**

La tarjeta ofrece la posibilidad de usar un programa que permite controlar los valores que se envían a la tarjeta de manera rápida mediante una interfaz sencilla. Este programa es útil para procesos como la calibración.



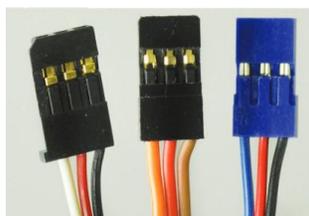
**Figura 14** Software Torobot  
**Fuente:** Autores

### 1.3.Servo Motor

Es un dispositivo capaz de ubicarse en una posición concreta, dentro de un rango de funcionamiento, manteniéndola constante, tienen un gran campo de aplicación dentro de los sistemas de radiocontrol y robótica.

#### 1.3.1. Características

Está conformado con un eje de rendimiento controlado que lleva a posiciones angulares específicas al enviar una señal codificada posee un conector con tres terminales como se observa en la Figura 14 en los estándares comerciales.



**Figura 15** Terminales de un Servo Motor

**Fuente:** <http://smartcubofiles.com/>

Sus características son:

- Alimentación positiva (cable rojo): entre 5V y 7V.
- Señal de control (cable blanco, naranja o azul): terminal de entrada por el que recibe la indicación de la posición que tiene que tener en cada momento.
- Alimentación negativa (cable negro o marrón): al negativo de nuestro sistema de alimentación.

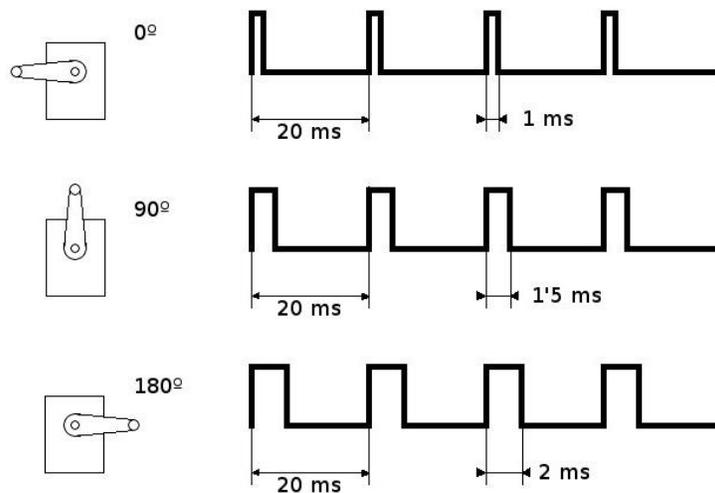
**Tabla 5** Características de Servo Motor

**Fuente:** <http://www.areatecnologia.com/electricidad/servomotor.html>

<b>Especificaciones</b>	<b>Valor</b>
<b>Dimensiones</b>	40x20x40.5 servo 64x55.6x20 Tamaño de soportes del metal
<b>Peso</b>	60g
<b>Longitud de cable</b>	320mm
<b>Velocidad</b>	0.16seg – 60° - 6V 0.14seg – 60° - 7.2V
<b>Torque</b>	15Kg.cm – 7.2V
<b>Voltaje de Operación</b>	4.8V – 8.2V
<b>Valor de zona muerta</b>	3 Microsegundos

### 1.3.2. Funcionamiento de un Servo Motor

El principio de funcionamiento de un servomotor está determinado por el motor interno que posee algunos circuitos de control y un potenciómetro, como se presenta en la Figura 15. Ambos elementos son conectados al eje central del servo motor. El circuito de control es capaz de supervisar el ángulo actual por medio del potenciómetro, de tal forma que si el eje está en el ángulo correcto, entonces el motor está apagado, si el circuito verifica que el ángulo no es el correcto, el motor gira en la dirección adecuada hasta llegar al ángulo correcto, el eje del servo es capaz de llegar alrededor de los 180 grados, sin embargo este rango de movimiento varía según el fabricante y en algunos casos el rango se amplía alrededor de los 210 grados, tiene un movimiento angular de entre 0 y 180 grados.



**Figura 16** Funcionamiento de un Servo Motor

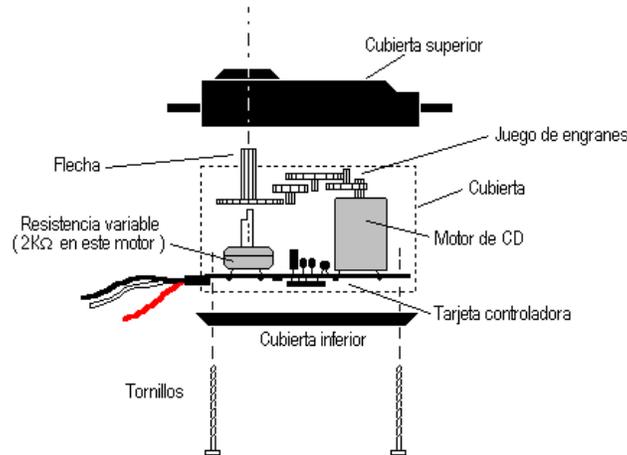
Fuente: <http://smartcubofiles.com/>

La señal de Control de un servo Motor establece la posición o ángulo de rotación de su eje de salida.

Para ello se utiliza un sistema conocido como PWM (Pulse Width Modulation) o Modulación por Anchura de Pulso, que sirve para establecer la posición del servo en función de la Anchura de un Pulso eléctrico aplicado en la entrada de control del Servo Motor. (Servo Motor, pág. 156).

### 1.3.3. Partes De Un Servo Motor

- Motor eléctrico: es el encargado de generar el movimiento, a través de su eje.
- Sistema de regulación: formado por engranajes, que actúa sobre el motor para regular su velocidad y el par. Mediante estos engranajes, normalmente ruedas dentadas, podemos aumentar la velocidad y el par o disminuirlas.
- Sistema de control o sensor: circuito electrónico que controla el movimiento del motor mediante el envío de pulsos eléctricos.
- Potenciómetro: conectado al eje central del motor que permite saber en todo momento el ángulo en el que se encuentra el eje del motor.

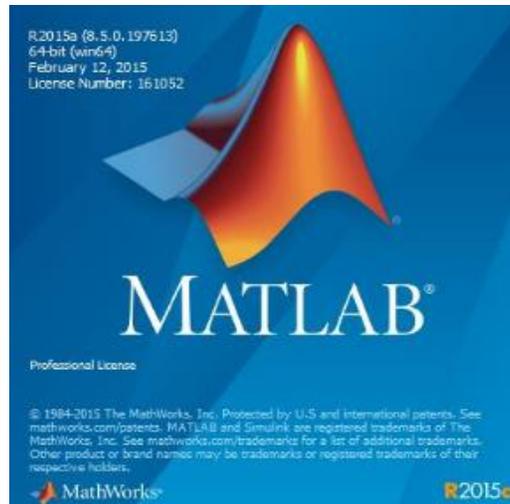


**Figura 17** Partes de un Servo Motor  
**Fuente:** <http://smarcubofiles.com/>

### 1.4. Matlab

Matlab es un entorno de trabajo para el cálculo científico, su nombre es la abreviatura de MATrix LABoratory dado que el tipo de dato básico que gestiona es una matriz. Puede ser utilizado en computación matemática, modelo y simulación, análisis y procesamiento de datos, visualización y representación de gráficos, así como el desarrollo de algoritmos.

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

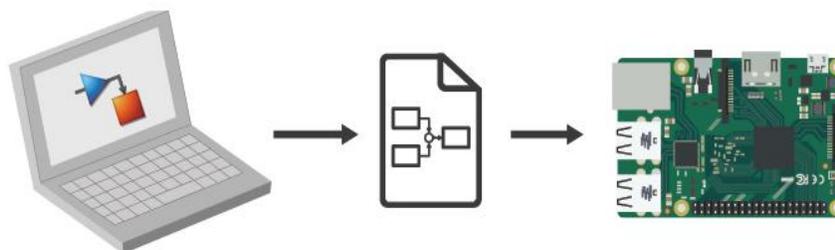


**Figura 18** Logo Matlab 2015  
**Fuente:** Mathworks

Sus prestaciones básicas encontramos: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario – GUI). Se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets).

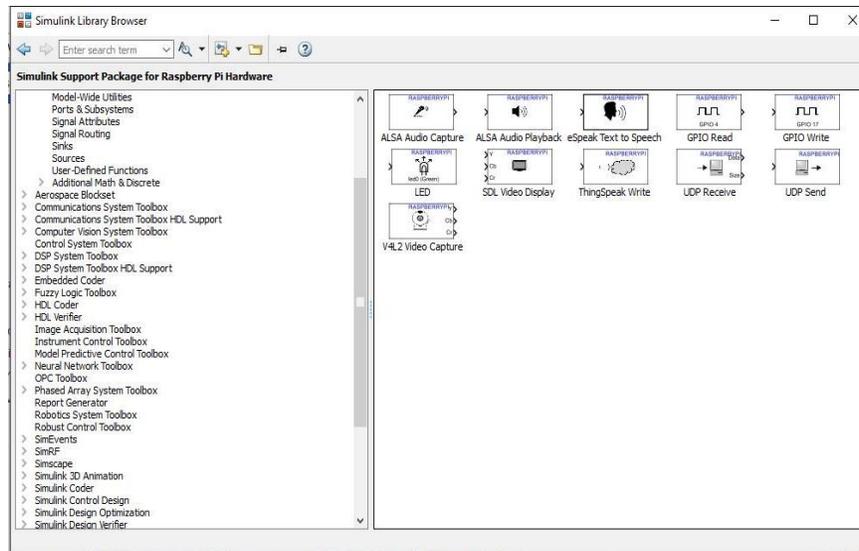
#### **1.4.1. Programación de Raspberry Pi con Matlab**

Para trabajar con imágenes, vídeos, audio y otros tipos de datos. Los Paquetes de Soporte Simulink para el hardware de Raspberry Pi le permiten crear y ejecutar modelos de Simulink. El paquete de soporte incluye una biblioteca de bloques de Simulink para configurar y acceder a los periféricos Raspberry Pi e interfaces de comunicación.



**Figura 19** Interacción Matlab - Raspberry Pi  
**Fuente:** Autores

Desarrollar algoritmos que ejecutan de forma autónoma en Raspberry Pi. Simulink le permite desarrollar algoritmos en un entorno, diagrama de bloques para el modelado de sistemas dinámicos y las ejecutan de forma autónoma, el envío y recepción de paquetes UDP, y la lectura o escritura de los sensores. Esto incluye la escritura de datos en el servicio de agregación de datos ThingSpeak libre para el Internet de las cosas.



**Figura 20** Herramientas de Raspberry Pi Matlab - Simulink  
**Fuente:** Autores

La prueba de conectividad con la Raspberry Pi, MATLAB ofrece un comando que es: `!ping ip.add.re.ss`. Este lo que hace es enviar y recibir datos a través de la Raspberry Pi, como lo muestra la figura 20.

```
>> !ping 169.254.0.31

Pinging 169.254.0.31 with 32 bytes of data:
Reply from 169.254.0.31: bytes=32 time=1ms TTL=64
Reply from 169.254.0.31: bytes=32 time<1ms TTL=64
Reply from 169.254.0.31: bytes=32 time<1ms TTL=64
Reply from 169.254.0.31: bytes=32 time<1ms TTL=64

Ping statistics for 169.254.0.31:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss)
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

fx >> |
```

**Figura 21** Datos de Raspberry Pi en Matlab  
**Fuente:** <http://colab-raspberry.blogspot.com/2013/11/raspberry-pi-matlab-simulink.html>

Las funciones para la compatibilidad de Matlab y Raspberry Pi se muestra en la Tabla 6 que permite desarrollar algoritmos.

**Tabla 6** Funciones de Matlab con Raspberry  
**Fuente:** Autores

Funciones	Descripción o Funcionamiento
<code>mypy</code>	Crea una conexión desde Matlab a la tarjeta Raspberry Pi
<code>mypi = raspi(ipaddress,username,password)</code>	Esta conexión para interactuar con el hardware Raspberry Pi y los dispositivos conectados (por ejemplo, la cámara del tablero, I2C, de serie, y SPI).
<code>configurePin</code>	Configurar pines GPIO como entrada digital, salida digital o salida PWM
<code>readDigitalPin</code>	Leer valor lógico del pin de entrada GPIO
<code>writeDigitalPin</code>	Escribir el valor lógico pin de salida GPIO
<code>showPins</code>	Mostrar diagrama de pines GPIO

El paquete de soporte de MATLAB para Raspberry Pi le permite escribir programas de MATLAB que se comunican con el Raspberry Pi y adquirir datos de los pines GPIO, cámaras y otros dispositivos conectados. Debido a que MATLAB es un lenguaje interpretado de alto nivel, es fácil para crear prototipos y ajustar los algoritmos para sus proyectos con Raspberry Pi, tales como el procesamiento de imágenes y de vídeo, optimización, estadística y procesamiento de señales.

A continuación presentamos el mensaje en la interacción de Matlab con Raspberry Pi.

```
mysecondpi =  
  
ROSSP con Propiedades:  
  
    DeviceAddress: '169.254.0.4'  
        Puerto: 18725  
        BoardName: 'Raspberry Pi Modelo B Rev 2'  
        AvailableLEDs: { 'LED0'  
AvailableDigitalPins: [4 7 8 9 10 11 14 15 17 18 22 23 24 25 27 30 31]  
AvailableSPIChannels: {}  
    AvailableI2Cbuses: { 'I2C-0' 'I2C-1'  
        I2CbusSpeed: 100000  
  
periféricos soportados
```

## 1.5.Python

Python es un lenguaje de programación fácil de aprender, muy popular y muy potente. Es un lenguaje de programación serio, usado en entornos profesionales, es desarrollado bajo una licencia open-source, por lo que puede ser usado en cualquier sistema con total libertad, incluso con fines comerciales.

Raspberry Pi Foundation lo ha elegido como el lenguaje de programación, esto no significa que Raspberry Pi se programa en Python, porque es un ordenador corriendo un sistema operativo que puede ejecutar programas escritos en cualquier lenguaje.

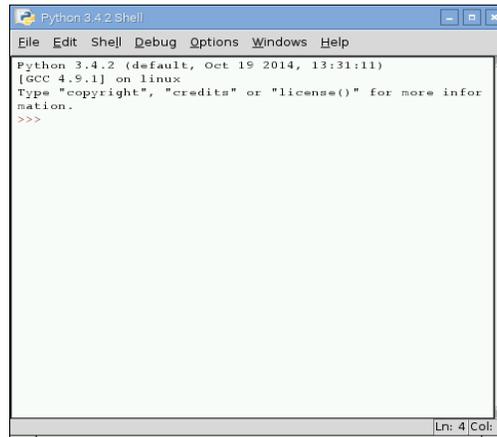
Python es un lenguaje interpretado, que quiere decir que un “intérprete” va leyendo las instrucciones y ejecutándolas en tiempo real. Un intérprete es un software encargado de convertir nuestras sentencias a código máquina. Puedes instalar cualquiera de los muchos intérpretes que existen para tu máquina, ya sea tu Raspberry Pi u ordenador personal. (Digital, 2016)



**Figura 22** Logo Python  
**Fuente:** <http://www.cristalab.com>

### 1.5.1. IDLE, el intérprete de Python de Raspberry Pi

El intérprete IDLE es el que viene preinstalado con Raspbian para Raspberry Pi y es muy sencillo de usar para aprender.



**Figura 23** New Model – Python  
**Fuente:** Autores

La interactiva de Python, es permitir introducir y ejecutar instrucciones una a una, mostrando los resultados de forma instantánea, para generar en la entrada un código que permita copiar a cada función ejecutado para por consiguiente cualquier programa sea ejecutable.

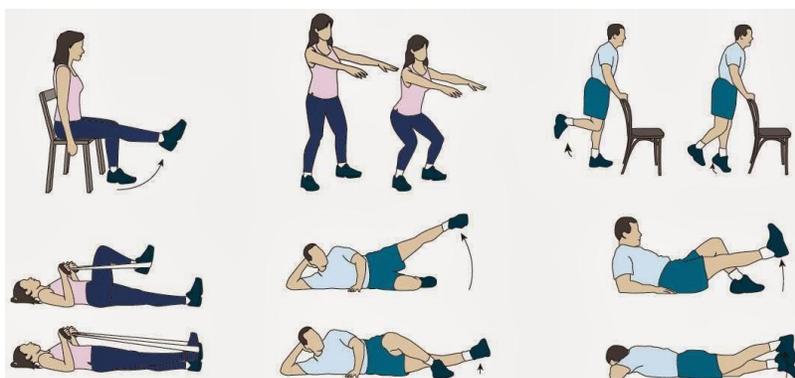
C Type	Python Type	ctypes Type
char	1-character string	c_char
wchar_t	1-character Unicode string	c_wchar
char	int/long	c_byte
char	int/long	c_ubyte
short	int/long	c_short
unsigned short	int/long	c_ushort
int	int/long	C_int
unsigned int	int/long	c_uint
long	int/long	c_long
unsigned long	int/long	c_ulong
long long	int/long	c_longlong
unsigned long long	int/long	c_ulonglong
float	float	c_float
double	float	c_double
char * (NULL terminated)	string or none	c_char_p
wchar_t * (NULL terminated)	unicode or none	c_wchar_p
void *	int/long or none	c_void_p

**Figura 24** Tipos de Datos – Python  
**Fuente:** Autores

## 1.6. Fisioterapia

La fisioterapia se centra en una precisa evaluación, diagnóstico y tratamiento de lesiones músculo-esqueléticas. Las lesiones músculo esqueléticas son las lesiones en articulaciones, músculos, huesos, tejidos blandos o de los nervios. Las patologías tratadas más comunes son el dolor de cuello y espalda, y lesiones de hombro, cadera y rodilla.

Los ejercicios de fisioterapia consisten básicamente en posturas estáticas, ejercicios de potenciación y estiramientos, tienen como objetivo normalizar el tono muscular y ayudarán a prevenir las complicaciones que derivan de la espasticidad, como el acortamiento del músculo, las rampas y el dolor.



**Figura 25** Ejercicios de Fisioterapia  
**Fuente:** <https://www.fisioterapia-online.com>

Las lesiones tratadas por el fisioterapeuta especializado en fisioterapia son:

- Lesiones de ligamentos (esguinces, roturas)
- Lesiones musculares
- Lesiones de cartílagos
- Luxaciones
- Fracturas
- Degeneración articular (artritis, artrosis)

Estas lesiones pueden requerir una evaluación y tratamiento de fisioterapia traumatológica y ortopédica para optimizar y acelerar el proceso de recuperación. Tras la evaluación, el

fisioterapeuta trabaja con el paciente para establecer objetivos y desarrollar un plan de fisioterapia adecuado. Algunos de los tratamientos que el fisioterapeuta puede utilizar son:

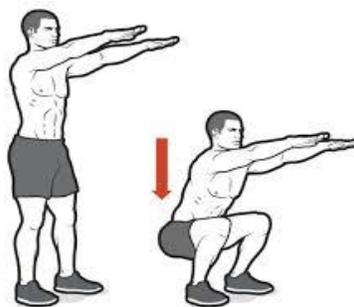
- Terapia manual
- Ejercicios terapéuticos
- Electroterapia
- Medidas de prevención y salud

El fisioterapeuta elegirá el tratamiento más adecuado basándose en la evaluación inicial del paciente.

- Optimizar el proceso de curación de la lesión
- Aumentar la fuerza en músculos.
- Disminuir el dolor
- Disminuir la inflamación
- Restaurar la movilidad
- Aumentar la fuerza
- Restaurar el movimiento normal
- Aumentar la autonomía del paciente
- Prevenir recaídas (Solofisio, 2012)

Los ejercicios más aplicados y fáciles de realizar son:

### Ejercicio 1: Sentadillas



**Figura 26** Ejercicio Sentadilla  
**Fuente:** <http://www.vitonica.com>

La sentadilla es un movimiento que se inicia de pie, mirando al frente y con la espalda recta, mientras los pies se separan del ancho de los hombros.

### **Técnica para Recuperación**

Pararse derecho con las piernas abiertas a las alturas de las caderas, estira los brazos, agacharse como si pareciera que se fuera a sentar en una silla y hasta lograr un ángulo de 90° con tus piernas, mantener la vista al frente sin encorvarte.

### **Errores Comunes**

- Perder la postura
- Levantar los talones
- Dejar las rodillas que se levanten o dejar que rebasen la punta de los pies al agacharte.

### **Ejercicio 2: Flexión de Pie**



**Figura 27** Ejercicio Flexión de Pie  
**Fuente:** <http://www.josemief.com>

Es un ejercicio para el trabajo de las piernas, más específicamente para la parte posterior de las mismas, es decir, para los isquiotibiales o femorales.

## Técnica de Recuperación

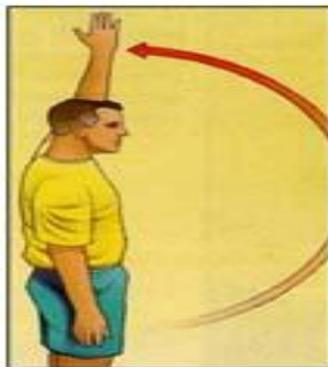
Primero pararse en forma firme con las piernas un poco abiertas y los brazos hacia el frente, luego pararse sobre una pierna (preferiblemente con un apoyo), después tiene que doblar la rodilla y moverlo de adentro hacia fuera hacerlo por 10 repeticiones

## Errores Comunes

- Perder la postura
- Perder estabilidad.

## Ejercicio 3: Flexión de Hombro

Las flexiones hindú son una variante de las habituales flexiones, solo que el movimiento en esta ocasión es más largo, por lo que el recorrido y la implicación muscular también lo es, debido a que tocaremos mucho más algunos grupos musculares como los hombros o los tríceps que con las flexiones convencionales no trabajamos tanto.



**Figura 28** Flexión de Hombro

**Fuente:** <http://ejerciciosencasa.es/tipos-de-flexiones-flexiones-para-hombros/>

## Técnica de Recuperación

Para su realización simplemente necesitaremos nuestro propio cuerpo que será el motor y el medio para la ejecución de este ejercicio, además de una esterilla que servirá para apoyar los movimientos de 90° de los brazos y hombros.

## Errores Comunes

- Levantar ligeramente la cadera
- Mirar hacia delante o hacia los pies.
- Serpentear con el tronco en la subida o bajada..

## Ejercicio 4: Abducción de Hombro

Es un movimiento\_ que aleja el miembro superior del tronco, se realiza en el plano frontal , en torno al eje anteroposterior. La amplitud de la abducción alcanza los 180°; el brazo queda vertical por arriba del tronco.



**Figura 29** Ejercicio de Abducción  
**Fuente:** [jeanaraaya.blogspot.com](http://jeanaraaya.blogspot.com)

## Técnica de Recuperación

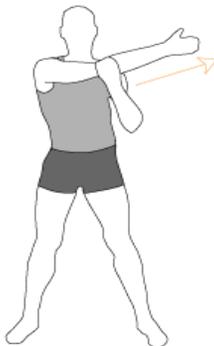
Levante los brazos lateralmente, con los codos derechos y las palmas hacia abajo. No encoja los hombros, ni incline el tronco. Mantenga esta posición 10 segundos; repitiéndolo 3 veces. Realizar este ejercicio 3 veces al día.

## Errores Comunes

- Levantar ligeramente la cadera
- Mirar hacia delante o hacia los pies.
- Serpentear con el tronco en la subida o bajada..

### **Ejercicio 5: Flexión de Brazos**

Es uno de los ejercicios físicos más populares. Son ideales para la construcción del músculo del pecho, hombros, brazos, y otros músculos del tronco, incluyendo los músculos abdominales.



**Figura 30** Ejercicio de Flexión de Brazos  
Fuente: <http://www.hagaestiramientos.com/>

### **Técnica de Recuperación**

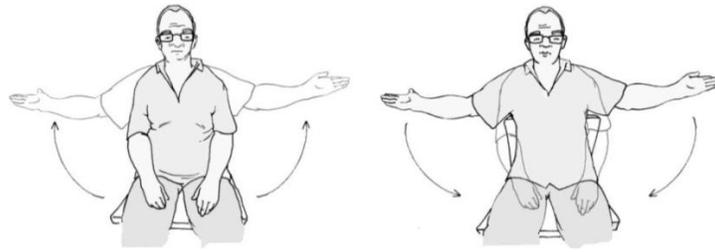
Para comenzar con las flexiones se debe colocar de pie en decúbito horizontal. las manos estiradas y se procede a subir y bajar los brazos hacia el pecho.

### **Errores Comunes**

- Movimiento de cadera
- Ubicación de Brazos
- Realizar rápidamente

### **Ejercicio 6: Extensión de Brazos**

El movimiento se puede realizar alternando los brazos o bien, con ambos brazos juntos, tomando una mancuerna en posición vertical, con ambas manos.



**Figura 31** Ejercicio Extensión de Brazos  
**Fuente:** <https://www.msn.com/es-us/salud/ejercicio>

### **Técnica de Recuperación**

El movimiento puede realizarse de pie o sentado. Para comenzar el ejercicio debemos tomar en una mano una mancuerna y elevar el brazo que posee la carga por el lateral de la cabeza, manteniendo el codo semiflexionado.

### **Errores Comunes**

- Posición de Brazos
- Parada Frontal
- Movimientos bruscos.

# CAPÍTULO 2

## **2. METODOLOGÍA**

### **2.1. Tipo de Estudio**

#### **2.1.1. Descriptivo**

El tipo de investigación es descriptiva, debido a que observar y describir los distintos tipos de ejercicios de Fisioterapia, para seleccionar los más adecuados para que el robot pueda ejecutarlos, así como el funcionamiento de la Tarjeta Raspberry Pi 2 que permite el monitoreo y evaluación del paciente, mediante el desarrollo del programa en Matlab.

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

#### **2.2.1. Métodos**

##### **2.2.1.1. Analítico/ Deductivo**

Se utilizó el método analítico-deductivo en el proyecto porque se inicia con un análisis particular de los componentes necesarios para el desarrollo del sistema y la forma de interactuar entre sí para que el sistema desempeñe su función de la manera más óptima.

#### **2.2.2. Técnicas**

##### **2.2.2.1. Observación**

Esta técnica consiste en la recolección de información que sea de apoyo para el desarrollo del proyecto, dando las pautas necesarias para el diseño y construcción de un robot para realizar rutinas de ejercicios de fisioterapia.

#### **2.2.3. Instrumentos**

Los instrumentos necesarios son libros, folletos, archivos, páginas web, blogs, cursos, datasheet, que son útiles para el diseño e implementación.

### **2.3. Población y Muestra**

La población interpreta los distintos tipos de ejercicios de fisioterapia.

La muestra son los ejercicios como sentadillas, abducción de hombros y brazo, flexión de pie, extensión de hombros y brazos.

## 2.4.Hipótesis

El diseño y construcción del robot permitirá la ejecución y monitoreo de las rutinas de ejercicios de fisioterapia.

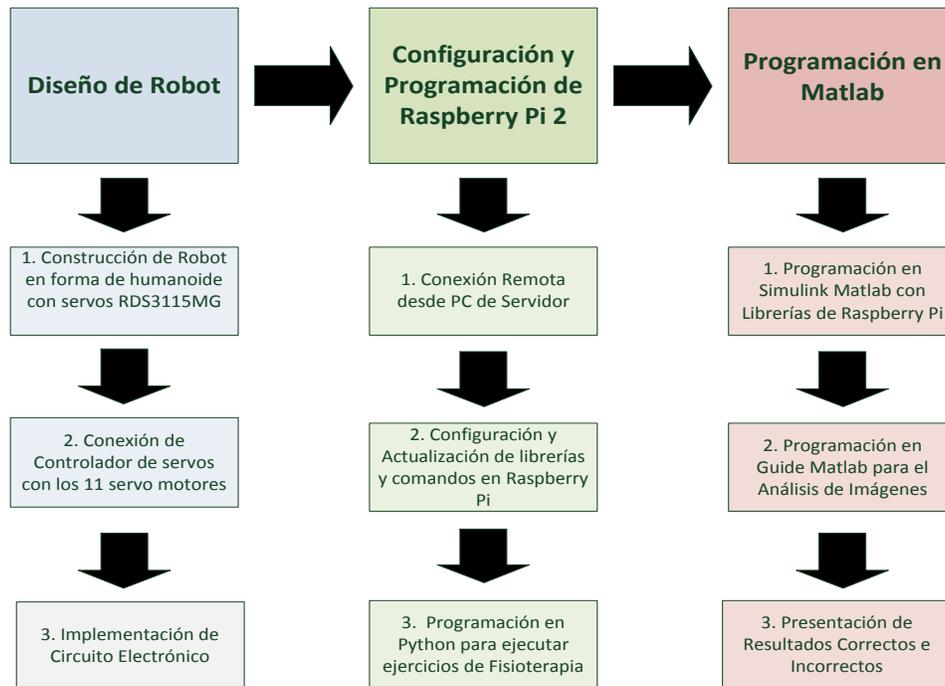
## 2.5.Operacionalización de Variables

**Tabla 7** Operacionalización de Variables  
Fuente.: Autores

Variable	Concepto	Categoría	Indicador	Técnicas e Instrumento
<b>Variable Independiente</b>  Robot	<b>EL Robot para ejercicios de Fisioterapia</b> El robot para ejercicios de Fisioterapia realiza ejercicios como sentadillas, abducción de brazos y hombro, flexión de pie, extensión de brazos y hombros.	Robótica	Grados de Libertad	Software de Control de Ejercicios
<b>Variable Dependiente:</b>  Las Rutinas de ejercicios de Fisioterapia	<b>Ejercicios de Fisioterapia</b> Método curativo de algunas enfermedades y lesiones físicas que se fundamenta en la aplicación de ejercicios físicos naturales o artificiales como la luz, el calor, el frío, el ejercicio físico, las radiaciones luminosas, los rayos X, etc	Ejercicios de Fisioterapia	<ul style="list-style-type: none"> <li>▪ Sentadillas</li> <li>▪ Aducción de brazos y Hombros</li> <li>▪ Flexión de brazos y Hombros</li> <li>▪ Extensión de Brazos</li> <li>▪ Flexión de Pie</li> </ul>	Ejercicios realizados correctos e incorrectos

## 2.6.Procedimientos

Para el diseño y construcción de un robot para realizar rutinas de ejercicios de fisioterapia, se implementó la siguiente serie de procesos a seguir, como se observa a continuación.

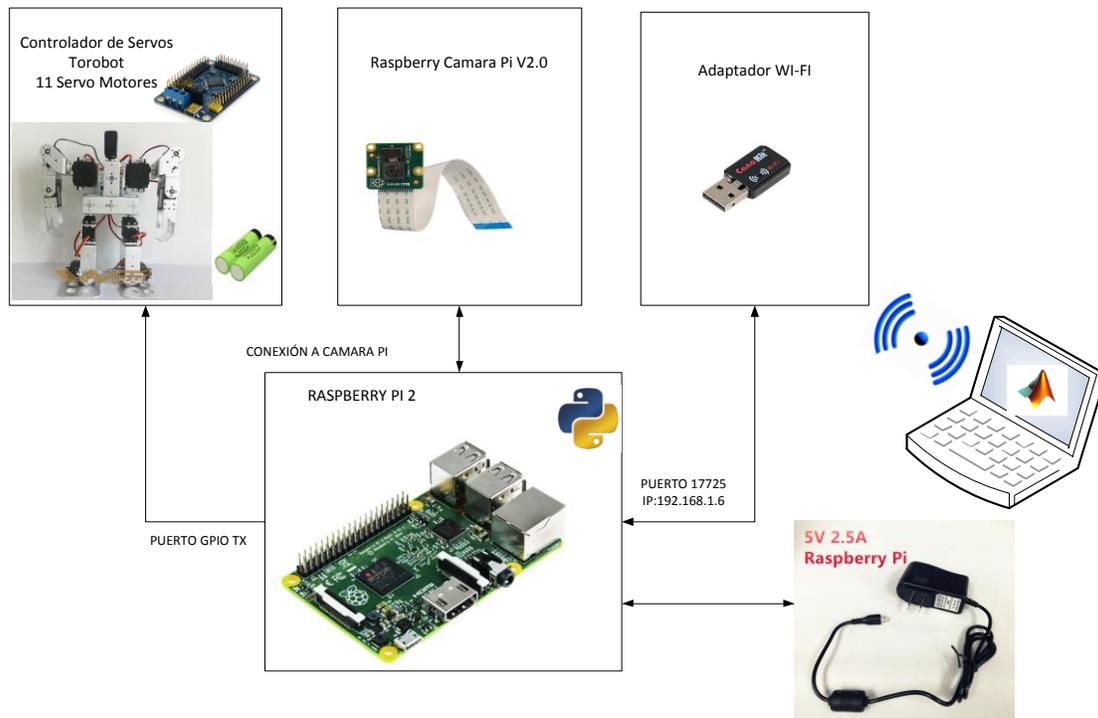


**Figura 32** Diagrama de Procedimientos

**Fuente:** Autores

### Diagrama de Funcionamiento

El sistema está conformado por un Robot de forma Humanoide con 10 servos de doble eje manejado mediante el controlador de servos Torobot y adaptados a dos pilas de tipo LifePo para su funcionamiento, la Tarjeta Raspberry Pi 2 está incorporado en la parte superior de la estructura del robot para tener una visión amplia en capturas de imágenes que son tomadas con la cámara, a su vez tiene una iunterfaz Wi- Fi para interactuar con Matlab, las imágenes capturadas se analizan a través del programa desarrollado en Matlab – Guide, los ejercicios de Fisioterapia serán verificados y evaluados diariamente o semanal para obtener un resultado satisfactorio.



**Figura 33** Diagrama de Conexión  
**Fuente:** Autores

## 2.6.1. Diseño de Robot

El diseño del robot está basado en prototipos de robots Humanoides de las marcas Bioloid, NAO y Kondo, adecuando los grados de libertad para los movimientos de los ejercicios de Fisioterapia, las piezas de ensamble se muestran en el **ANEXO 1**, los cuales fueron diseñados en AutoCAD, y elaboradas de aluminio.

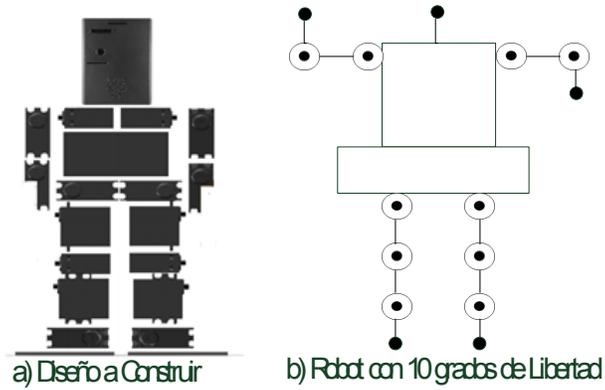
### 2.6.1.1. Diseño de Robot en forma de Humanoide

El robot debe tener dos brazos, dos piernas, una cabeza y todos deben estar unidos a un tronco, también debe de ser capaz de tener mayor grado de libertad al realizar movimientos, la estructura del robot se forma con piezas que están construidas en aluminio, que permite reducir el peso del conjunto y dota de rigidez a la estructura. Ver **ANEXO 1**.

### 2.6.1.2. Construcción de Robot

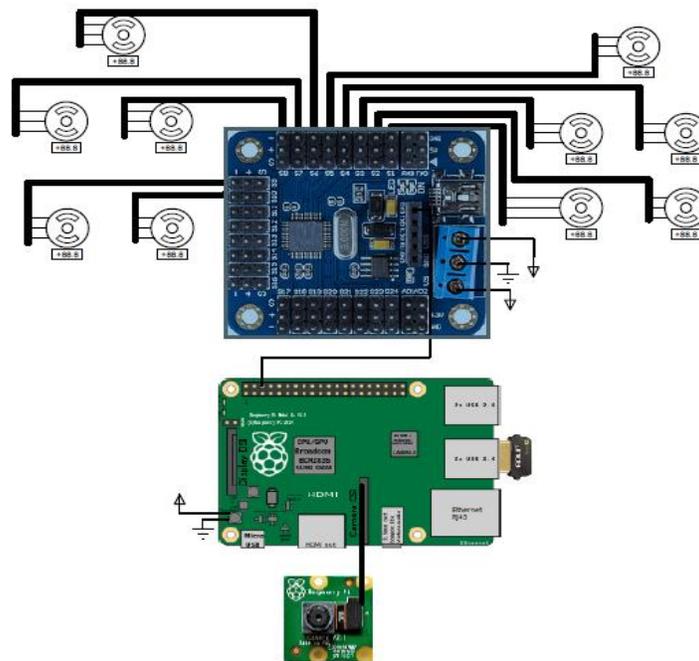
En el momento del diseño mecánico, se toma en cuenta el peso y la estabilidad que tendrá el robot que dependiendo del peso y del centro de gravedad permitirá realizar los

movimientos de Fisioterapia. La estructura mecánica que se desarrolló cuenta con 10 servos de marca RDS3115MG, controlados por la tarjeta de control de servos motores de 32 canales Torobot. Ver ANEXO 2.



**Figura 34** Modelo a Construir  
Fuente: Autores

### 2.6.1.3. Implementación de Circuito Electrónico

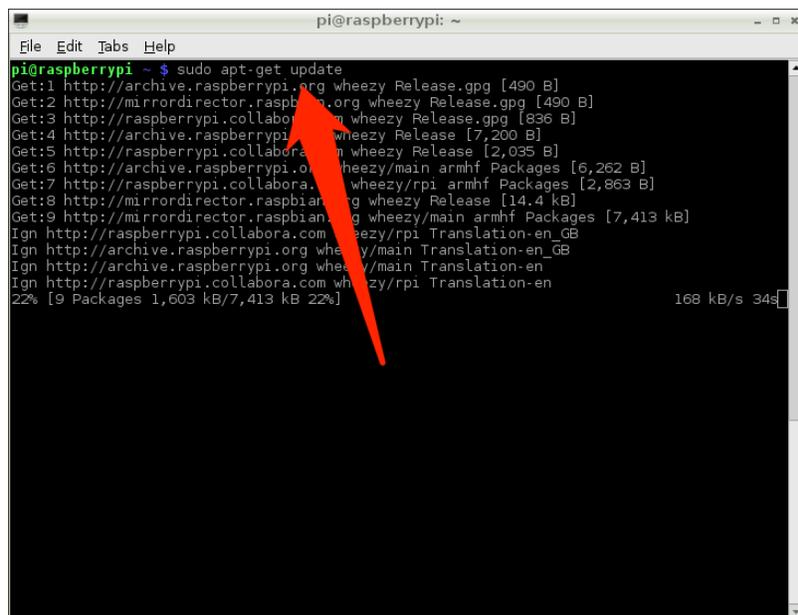


**Figura 35** Diagrama de Conexión  
Fuente: Autores

Para la implementación del circuito electrónico se realiza la conexión de Raspberry Pi alimentado con 5V y una corriente de 2000mA. El adaptador Wi-Fi se conecta en el puerto USB de la tarjeta, la cámara viene con un bus de datos que se conecta en la ranura de la tarjeta, el puerto GPIO TX pin 8 que es una comunicación UART se conecta al pin RX del controlador de servos, el PIN 6 del GPIO que es tierra del Raspberry pi se conecta con una tierra en común del controlador de servos, los terminales de los diez servo motores están conectados al controlador de 32 servos, y alimentados con dos pilas de 3.2V a 2300 mAh en serie teniendo voltaje total 6.4V.

### 2.6.2. Configuración y Programación de Raspberry Pi 2

La tarjeta Raspberry pi consta de una memoria SD card que contiene el sistema operativo Raspbian Wheezy descargado desde el software MATLAB, a su vez pueden conectarse algunos accesorios como: mouse, teclado, monitor, adaptador WI FI.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~$ sudo apt-get update  
Get:1 http://archive.raspberrypi.org wheezy Release.gpg [490 B]  
Get:2 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]  
Get:3 http://raspberrypi.collabora.com wheezy Release.gpg [836 B]  
Get:4 http://archive.raspberrypi.org wheezy Release [7,200 B]  
Get:5 http://raspberrypi.collabora.com wheezy Release [2,035 B]  
Get:6 http://archive.raspberrypi.org wheezy/main armhf Packages [6,262 B]  
Get:7 http://raspberrypi.collabora.com wheezy/rpi armhf Packages [2,863 B]  
Get:8 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]  
Get:9 http://mirrordirector.raspbian.org wheezy/main armhf Packages [7,413 kB]  
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB  
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB  
Ign http://archive.raspberrypi.org wheezy/main Translation-en  
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en  
22% [9 Packages 1,603 kB/7,413 kB 22%] 168 kB/s 34s
```

Figura 36 Actualización de Librerías y Paquetes

Fuente: Autores

Para realizar el proceso de descarga e instalación del sistema operativo para la tarjeta Raspberry Pi. Ver ANEXO 4

### 2.6.2.1. Conexión Remota PC – Raspberry Pi



**Figura 37** Conexión Remota de PC  
**Fuente:** Autores

Para la conexión remota de Raspberry Pi a una PC se utilizara la red inalámbrica WI FI de nuestro modem o router, se configura el dispositivo inalámbrico de la tarjeta Raspberry Pi, para realizar la comunicación a la PC, mediante SSH (Secure SHell, en español: intérprete de órdenes seguro) usando el programa PuTTY procedemos a ingresar la dirección IP del dispositivo Wi FI del Raspberry Pi para acceder a la conexión remota para esto ingrese a la terminal de su Raspberry Pi y verifique sus aplicaciones. Ver **ANEXO 5**

### 2.6.2.2. Actualización de Comandos y Librerías en Raspberry Pi

Raspbian “Wheezy” ha sido instalado, por consiguiente se configura los parámetros acorde a la necesidad del usuario. Se puede configurar el idioma, la expansión de tarjeta SD, el adaptador Wi Fi, la hora y fecha, la comunicación serial, el I2C habilitación de cámara.

A continuación se presentan los comandos para la actualización de las librerías de la tarjeta Raspberry pi.

```
sudo apt-get update
```

El comando sudo apt-get update comprueba si existen paquetes sin actualización desde la plataforma de Raspberry, y si ese fuera el caso de que los paquetes no están actualizados automáticamente los actualiza.

```
sudo apt-get upgrade
```

El comando `sudo apt-get upgrade` permite realizar las descargas de los instaladores de las distintas aplicaciones que posee Raspberry pi.

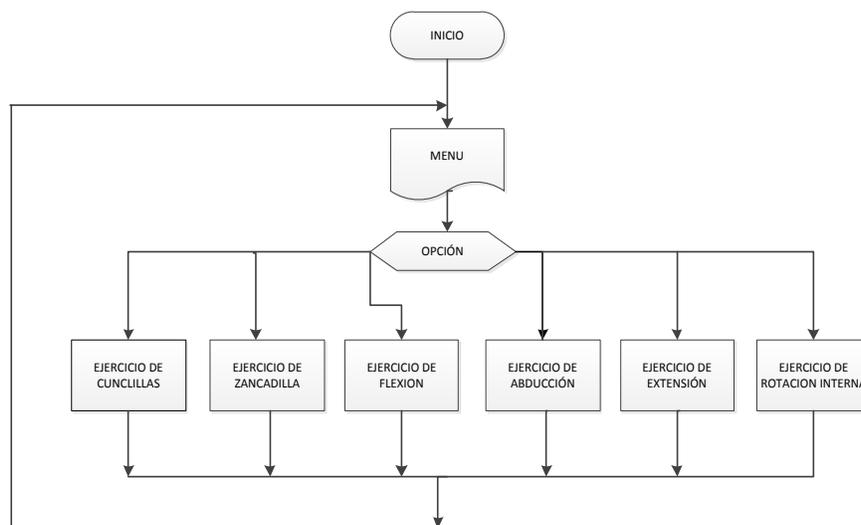
```
sudo rpi-update
```

El comando `sudo rpi-update` permite realizar la actualización del firmware a su última versión, después de terminar con la ejecución de dicho comando se procede a reiniciar la tarjeta Raspberry para que el firmware actualizado se mantenga grabado en la tarjeta con el siguiente comando: **sudo reboot**.

Para la configuración se ingresa al terminal del dispositivo el primer comando que se ejecuta es **raspi-config**, este programa solo se ejecuta para experimentar y analizar las distintas aplicaciones de la tarjeta.

### 2.6.2.3. Programación en Python para ejecutar ejercicios de Fisioterapia

La programación para la ejecución de los ejercicios de Fisioterapia, mediante el software Python instalada en nuestra tarjeta Raspberry se procede a realizar nuestro algoritmo. Es este programa consiste básicamente en imprimir en pantalla un mensaje de texto, con los seis ejercicios propuestos para las rutinas de Fisioterapia su programación la describimos en un diagrama de flujo que se presenta a continuación.



**Figura 38** Diagrama de Flujo - Programación en Python

**Fuente:** Autores

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 time
import serial
import subprocess
```

Una vez importados las librerías del puerto serial y las funciones se procede a configurar el puerto serial

```
ser=serial.Serial("/dev/ttyAMA0")
ser.baudrate=9600
```

Luego se procede a programar un menú con los seis ejercicios planteados, al ejecutar el programa se mostrara la opción de seleccionar un ejercicio.

```
EJERCICIOS DE REHABILITACION PARA LOS BRAZOS Y PIERNAS
```

```
>> Ejercicio 1 : A
>> Ejercicio 2 : B
>> Ejercicio 3 : C
>> Ejercicio 4 : D
>> Ejercicio 5 : E
>> Ejercicio 6 : F
>> Ejercicio 7 : G
```

```
Seleccione el ejercicio a realizar
```

Una vez seleccionado un ejercicio se procederá la comunicación serial de la tarjeta Raspberry con el controlador de servos, así podrá ejecutarse los códigos de movimientos del robot indicando el ejercicio de fisioterapia.

```
ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P2500#10P1211T500\r\n")      ##Parada del robot
```

Para realizar repeticiones se programara con la función **for** que ejecutaremos las veces que sean de acuerdo a las repeticiones de cada ejercicio.

```
for i in range(1,10)
```

```
else:
```

La función **time.sleep** () indica el tiempo de ejecución.

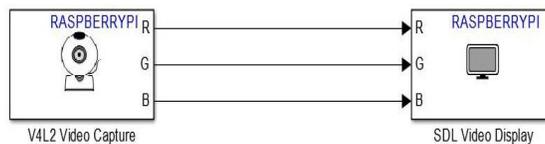
```
time.sleep(1:3)
```

### 2.6.3. Programación en Matlab

Para el funcionamiento de la transmisión de video, el análisis y procesamiento de imágenes capturadas a los pacientes que realizan ejercicios realizados correspondientes a la fisioterapia se programa en el software Matlab 2015 por su utilidad en la informática, su facilidad de uso de sus herramientas y comandos para ejecutar cualquier orden, operación matemática, o simulaciones de circuitos electrónicos.

#### 2.6.3.1. Programación en Matlab – Simulink

Para realizar el monitoreo del paciente se debe establecer la conexión entre la cámara de Raspberry Pi y Matlab-Simulink, con esta herramienta se establecen los parámetros de conexión para la transmisión de Video en tiempo Real, y diseñar el diagrama de bloques como muestra la figura 38 los parámetros de configuración y funcionamiento de cada bloque. (VER ANEXO 8)

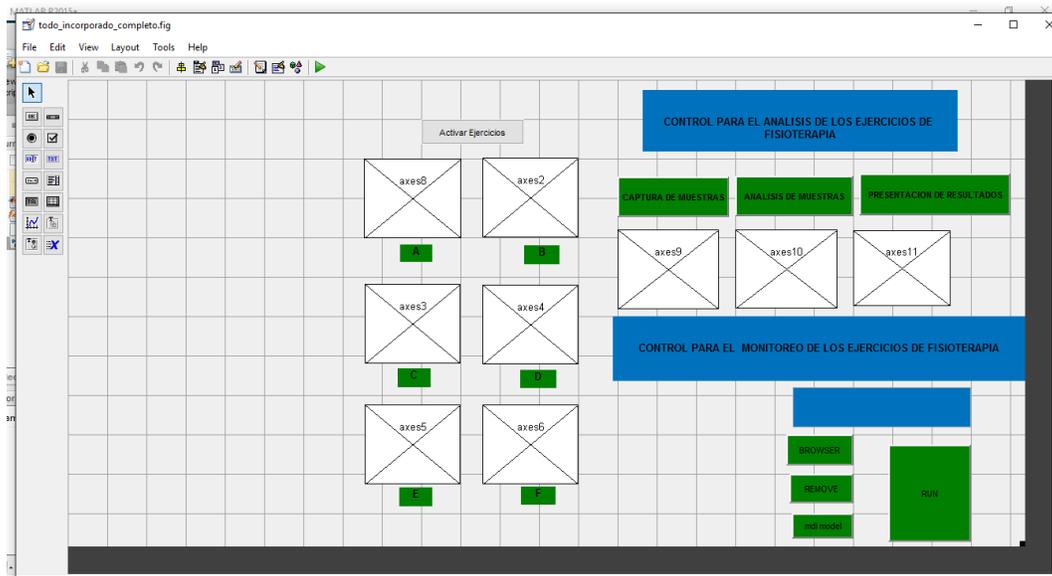


**Figura 39** Programación en Matlab - Simulink

**Fuente:** Autores

El bloque de V4L2 Video Capture indica la conexión de la cámara de Raspberry Pi, para la captura de imágenes en colores RGB permitiendo así imágenes de alta calidad, el bloque de SDL Video Display es la pantalla por donde se podrá visualizar las imágenes capturadas, se puede configurar según los pixeles deseados de nuestra imagen.

### 2.6.3.2. Programación en Matlab – Guide para Procesamiento de Imágenes.



**Figura 40** Programación Matlab – Guide  
.Fuente: Autores

Para el procesamiento de imágenes se tiende a tomar varias imágenes tomadas desde la cámara de Raspberry pi, para realizar este procedimiento hay que lograr que el Matlab interactúe con la tarjeta Raspberry pi a través de los siguientes comandos:

```
mypi=raspi();  
myCam = cameraboard(mypi,'Resolution', '640x480')
```

Con los comandos escritos en la ventana principal del software MATLAB nos permite crear una conexión con la tarjeta Raspberry Pi, y con ello se puede tomar varias imágenes para su procesamiento.

**Datos de forma entero:**

```
global mypi;  
global myCam;  
global y;  
global q;  
global t;  
global guardardat;
```

## Interacción de Matlab con Raspberry Pi

```
mypi=raspi();  
myCam = cameraboard(mypi,'Resolution', '640x480');
```

Para mostrar las imágenes captadas desde la cámara del Raspberry Pi y visualizarlas en el software MATLAB hay que tener en cuenta los siguientes comandos:

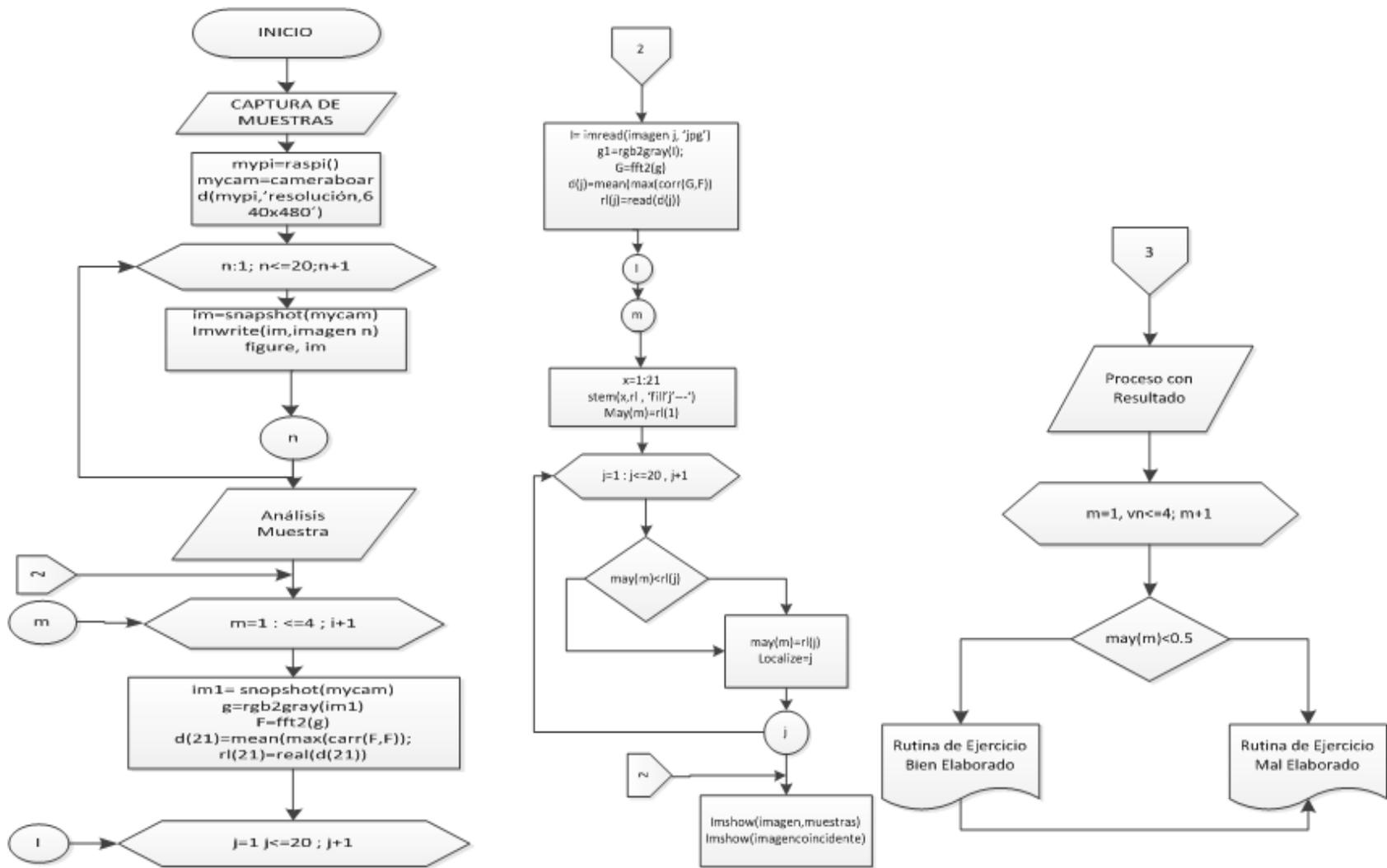
```
im = snapshot(myCam)
```

Con el comando **snapshot** se capta una imagen.

```
imwrite(im, nombre)
```

El comando **imwrite** permite la escritura de dicha imagen captada.

```
figure, imshow(im);
```



**Figura 41** Diagrama de Flujo - Procesamiento de Imágenes  
**Fuente:** Autores

El comando **imshow** permite la visualización de la imagen en la pantalla del matlab.

```
imshow(Imgcoincidente)
axes(handles.axes10)           %%%% mostrar imagen en el
axes 10
imshow(im1)
pause(15);
```

Con las imágenes captadas se puede hacer el análisis, tomando una imagen patrón para así lograr una comparación entre la imagen patrón y las primeras imágenes que fueron captadas con los comandos descritos.

Para captar la imagen patrón se vuelve a realizar el proceso anterior, ya con la imagen captada se debe transformar la imagen a blanco y negro con la siguiente instrucción:

```
g=rgb2gray(g1)
```

Una vez transformada la imagen a blanco y negro, con la instrucción **F= fft2(g)** se genera la transformada de Fourier.

### **Transformada de Fourier**

```
g1=imread('patron', 'jpeg');
g=rgb2gray(g1);
F= fft2(g);
d(13)=mean(max(corr(F,F)));
r1(13)=real(d(13));
```

La transformada de Fourier en las imágenes corresponde a los niveles de gris o a la intensidad de las diferentes filas o columnas de la matriz de la imagen.

Se aplica el método de correlación:

```
d(13)=mean(max(corr(F,F))),
```

Permite asimilar el grado de relación que existe entre dos variables, en el caso de las imágenes se va a correlacionar con la misma imagen para así alcanzar una ponderación de 1 la cual va hacer la máxima cuando estas sean las mismas.

Como resultado se obtiene un número complejo en el cual la parte imaginaria es nula el

cual se tiende a tomar únicamente la parte real con la siguiente instrucción.

```
r1(13)=real(d(13));
```

Una vez realizado el mencionado proceso con la imagen patrón se repite el proceso para cada una de las imágenes con las que van hacer comparadas, con una modificación en el método de correlación ya que la comparación ya no se realiza con sí misma, esta comparación se la realizara con la imagen patrón el cual presenta como resultado una imagen parecida con una ponderación que se encontrara en un rango de 0 a 1.

Dicho resultado se visualizara en un diagrama de barras con la siguiente instrucción:

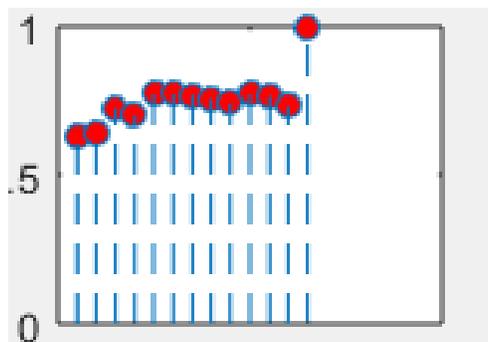
```
h=stem(x,r1,'fill','--')
```

### Almacenamiento de Resultados en un vector

```
guardardat(t)=10;  
save('C:\Users\Luis  
Fernando\Desktop\INTERFAZ\datosemana.mat','guardardat','t');
```

Con el diagrama de barras se visualizara cuál de las imágenes es la que más se asemeja a la imagen patrón, y para visualizar la imagen más parecida hay que localizarla en una variable y con el comando **inshow** se notara claramente la imagen.

Para detectar si el ejercicio de fisioterapia está bien realizada hay que verificar si la ponderación se encuentra arriba de 0.5, si dicha ponderación es más baja se dice que el ejercicio está mal realizado.



**Figura 42** Análisis de Resultados  
**Fuente:** Autores

Los Resultados Correctos podemos analizar en la imagen que Matlab visualizara en forma de barras, el cual describe el armónico más próximo a 0.5 es la que cumple con la ejecución del ejercicio correcto en caso del ejercicio Incorrecto se dará cuando son los armónicos bajos a 0.5 puedes ser incluso hasta negativos esto dice que no se parece el ejercicio al ejercicio correcto

## **2.7.Comprobación de Hipótesis**

Para la comprobación de la hipótesis se utiliza el método estadístico de Chi-Cuadrado mediante la Prueba de Bondad de Ajuste, en la que toman en cuenta el número de éxitos en las pruebas realizadas, en este caso será el monitoreo por medio del robot si realiza bien o mal los ejercicios de fisioterapia.

### **2.7.1. Técnicas de Procedimiento para el Análisis**

El análisis estadístico que se aplica es cualitativo y cuantitativo, para la parte cualitativo es comprobar la estabilidad y confiabilidad del monitoreo del robot que realizara al paciente por medio de la transmisión de video vía la conexión WIFI, para la parte cuantitativa se comprobará la eficiencia y la utilidad de este robot en la demostración de rutinas de ejercicios en centros de fisioterapia.

### **2.7.2. Comprobación de Hipótesis**

**Ho** Con el diseño y construcción del robot no permitirá la ejecución y monitoreo de las rutinas de ejercicios de fisioterapia.

**H1** Con el diseño y construcción del robot permitirá la ejecución y monitoreo de las rutinas de ejercicios de fisioterapia.

### **2.7.3. Elección de la Prueba de Hipótesis**

Es una prueba de muestras relacionadas a un mismo grupo se le aplica dos medidas en un tiempo distinto, es un estudio longitudinal, la variable fija que crea los grupos se muestra dos medidas una antes y una medida después, la variable aleatoria de comparación, es la variable tiempo es una variable numérica, por lo tanto se usa la prueba del chi cuadrado

para comprobar la hipótesis. En tabla 8 se muestra los diferentes ejercicios de Fisioterapia que realizaron 10 pacientes como prueba si realizo bien o mal.

$H_o$  = No influye el tipo de ejercicio

$H_i$  = Si influye el tipo de ejercicio

**Tabla 8** Resultados de Ejercicios de Fisioterapia  
Fuente: Autores

Monitoreo de Ejercicios Correctos e Incorrectos							
	Ejercicio 1	Ejercicio 2	Ejercicio 3	Ejercicio 4	Ejercicio 5	Ejercicio 6	TOTAL
SI	5	4	6	7	7	7	36
NO	3	2	3	2	1	1	12
TOTAL	8	6	9	9	8	8	48

El nivel de significancia de alfa es  $\alpha= 0,05\%$  que representa el 5% y con un 95% de confianza.

**Tabla 9** Frecuencias Esperadas  
Fuente: Autores

Frecuencias Esperadas						
SI	6	4,5	6,75	6,75	6	6
NO	2	1,5	2,25	2,25	2	2

Grado de Libertad

$$v = n^{\circ} \text{ filas} - 1 \times (n^{\circ} \text{ columnas} - 1)$$

**Tabla 10** Grado de Libertad  
Fuente: Autores

Grado de Libertad	5
-------------------	---

**Formula de Chi cuadrado**

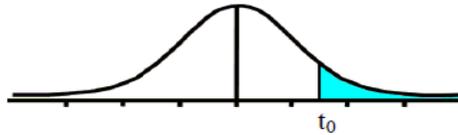
$$\chi^2 = \sum \frac{(o_i - e_i)^2}{e_i}$$

**Tabla 11** Cálculos de Chi Cuadrado

**Fuente:** Autores

CHI CUADRADO						
0,16666667	0,05555556	0,08333333	0,00925926	0,16666667	0,16666667	0,64814815
0,5	0,16666667	0,25	0,02777778	0,5	0,5	1,94444444
<b>TOTAL</b>						<b>2,59259259</b>

Tabla t-Student



Grados de libertad	0.25	0.1	0.05	0.025	0.01	0.005
1	1.0000	3.0777	6.3137	12.7062	31.8210	63.6559
2	0.8165	1.8856	2.9200	4.3027	6.9645	9.9250
3	0.7649	1.6377	2.3534	3.1824	4.5407	5.8408
4	0.7407	1.5332	2.1318	2.7765	3.7469	4.6041
5	0.7267	1.4759	2.0150	2.5706	3.3649	4.0321
6	0.7176	1.4398	1.9432	2.4469	3.1427	3.7074
7	0.7111	1.4149	1.8946	2.3646	2.9979	3.4995
8	0.7064	1.3968	1.8595	2.3060	2.8965	3.3554
9	0.7027	1.3830	1.8331	2.2622	2.8214	3.2498
10	0.6998	1.3722	1.8125	2.2281	2.7638	3.1693
11	0.6974	1.3634	1.7959	2.2010	2.7181	3.1058
12	0.6955	1.3562	1.7823	2.1788	2.6810	3.0545
13	0.6938	1.3502	1.7709	2.1604	2.6503	3.0123
14	0.6924	1.3450	1.7613	2.1448	2.6245	2.9768
15	0.6912	1.3406	1.7531	2.1315	2.6025	2.9467
16	0.6901	1.3368	1.7459	2.1199	2.5835	2.9208
17	0.6892	1.3334	1.7396	2.1098	2.5669	2.8982
18	0.6884	1.3304	1.7341	2.1009	2.5524	2.8784
19	0.6876	1.3277	1.7291	2.0930	2.5395	2.8609
20	0.6870	1.3253	1.7247	2.0860	2.5280	2.8453

**Figura 43** Tabla T- Student

**Fuente:** Autores

<b><math>X^2_{\text{tabla}} = 2.0150</math></b>	<b><math>X^2_{\text{calculado}} = 2.5925</math></b>
2.5925 > 2.0150 se rechaza la hipótesis nula	

Por lo tanto los ejercicios si realizan correctamente.

#### **2.7.4. Muestra**

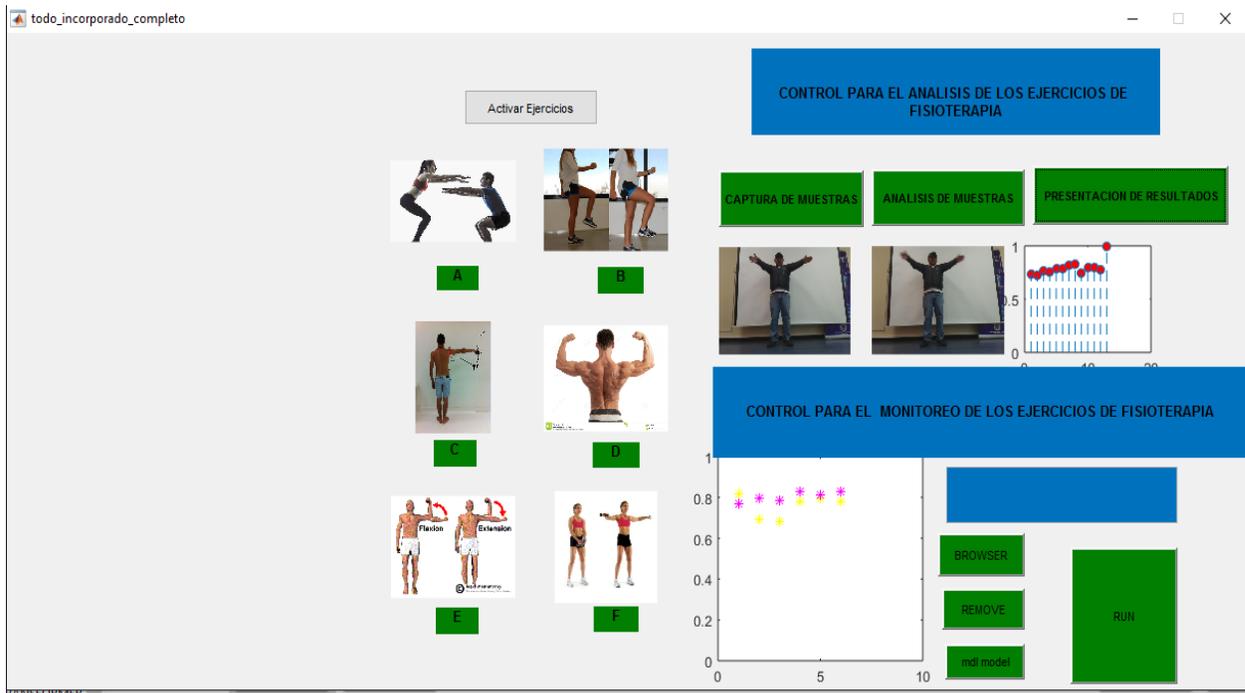
La muestra se realizó de 50 corridas de software en distintos días del ejercicio de fisioterapia extensión de brazos, se tomó solamente 15 muestras para la comprobación de hipótesis de la ejecución del programa desarrollado para verificar si realizó bien o mal el ejercicio.

## CAPÍTULO III

### 3. RESULTADOS

#### 3.1.Resultados de la funcionalidad del sistema y software desarrollado

Las imágenes capturadas desde el menú desarrollado en Matlab, obtenemos una secuencia de Imágenes realizando ejercicios de Fisioterapia, ejercicios que serán ejecutados por el robot al seleccionar los diferentes tipos de Ejercicios, el robot aplicara su movimiento adecuado para que el paciente siga el ejemplo, se obtuvo datos de dos días para analizar la recuperación del paciente si es satisfactoria o necesita seguir realizando rutinas de ejercicios para una pronta y excelente recuperación.

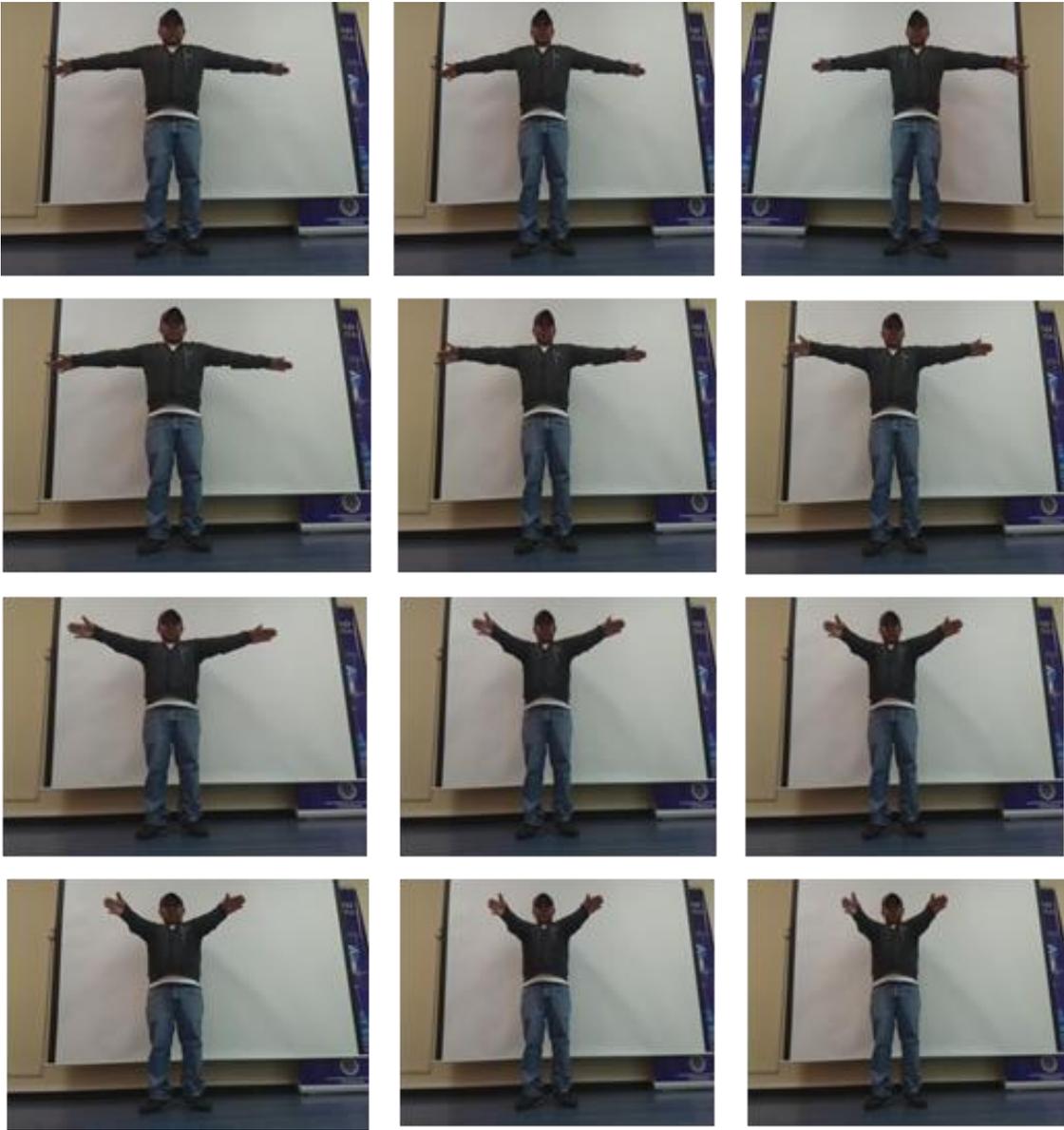


**Figura 44** Sistema Desarrollado para Aplicaciones de Ejecución

**Fuente:** Autores

La secuencia de imágenes realizando Ejercicios del primer día, se muestra que el paciente tuvo resultados altos de 0,5 esto indica que los ejercicios fueron bien elaborados, estos datos se guardan para la evaluación semanal.

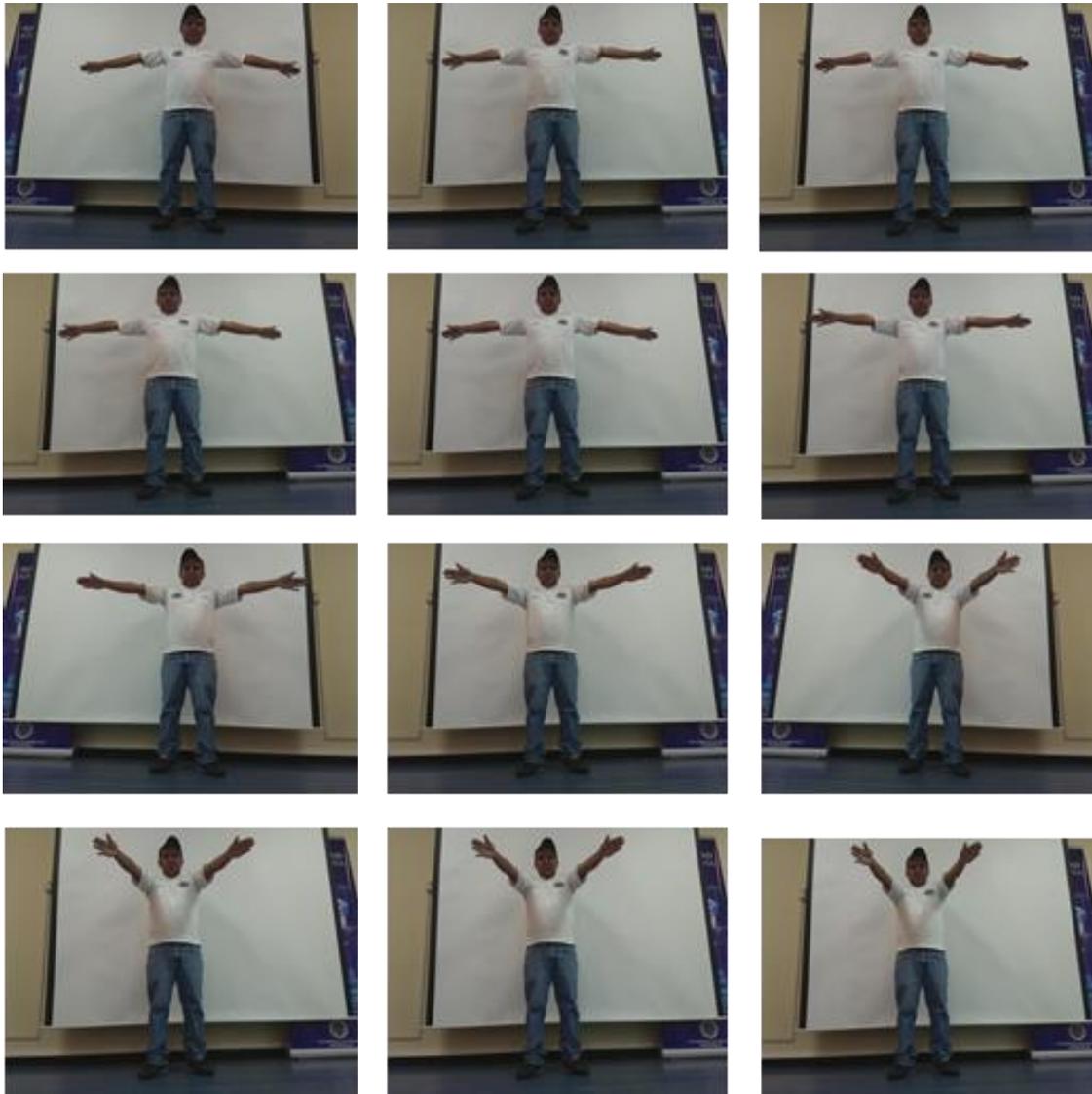
Ejercicios Realizados el Primer día



**Figura 45** Resultados del Primer día  
**Fuente:** Autores

Los resultados del segundo día son realizados correctamente, por lo que la ponderación fue mayor de 0,5 esto permite que el paciente analice como está el proceso de recuperación, finalmente los resultados se muestran en la gráfica de coordenadas x, y.

## Ejercicios Realizados el Segundo Día



**Figura 46** Resultados del segundo día  
**Fuente:** Autores

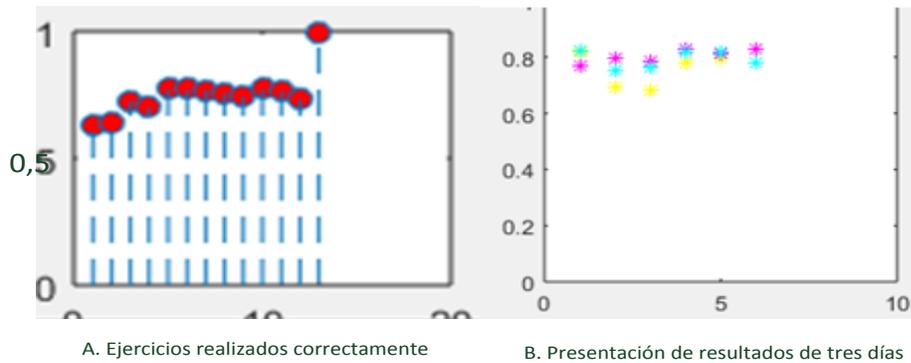
El proceso de recuperación al realizar los ejercicios, emplea movimientos firmes y lentos, esto se debe realizar por un tiempo aproximado de 30 segundos y repítelos de 3 a 5 veces al día.

Ejercicios Realizados el Tercer Día



**Figura 47** Resultados del tercer día  
**Fuente:** Autores

Para poder analizar los ejercicios realizados correctos e incorrectos de una paciente se debe verificar con el grafico de puntos que se muestra en la figura 47.



La Evaluación de la recuperación del paciente se muestra a través de una gráfica de puntos, así se podrá ordenar al paciente que continúe o repita el ejercicio. En la figura 47 b. se analiza la evaluación del paciente de tres días teniendo una recuperación satisfactoria.

### 3.2. Análisis Financiero

**Tabla 12 Análisis Financiero**

Fuente: Autores

<b>Costos</b>	<b>Descripción</b>	<b>Cantidad</b>	<b>Valor Usd.</b>
<b>Hardware</b>	Computador	1	1220
	Raspberry pi 2	1	77.45
	Adaptador para RPI	1	10
	Convertidor HDMI-VGA	1	15
	Amplificador WIFI	1	25
	Cámara Pi V2.0	1	35.40
	Caja de Raspberry	1	10.15
	Monitor	1	40
	Teclado	1	6
	Mouse	1	6
	Servo Motor 15Kg	12	250
	Reproductor de Voz	1	50
	Protoboard	1	25
	Multímetro	1	60
	Cables	20	3
	Trabajo de Aluminio	1	20
<b>Software</b>	Raspbian	1	0
	Matlab	1	0
	Python	1	0
	Torobot Rio USB	1	0
	Advanced Port Scanner	1	0
	VNC	1	0
<b>Proyecto</b>	Transporte		50
<b>Varios</b>	Cables, silicona, papel tapiz el etc.		50
<b>Imprevistos</b>			50
<b>Total</b>			<b>2002.00</b>

## CAPÍTULO IV

### 4. CONCLUSIONES Y RECOMENDACIONES

#### 4.1. Conclusiones

El robot posee dos celdas LifePo de 3.2V que permite tener un voltaje de 6.4V, un rango de autonomía de una hora de funcionamiento constante y soportar una carga de 40 Amperios por corto periodo de tiempo.

El robot humanoide podrá incrementar grados de libertad, añadiendo servos de doble eje, en cabeza, tronco, muñecas que le permitirá tener movilidad y ejecutar mayor número de ejercicios de Fisioterapia.

Las imágenes capturas mediante el programa desarrollado en Matlab - Guide poseen un retardo aproximadamente de 15 segundos debido a la latencia en la tarjeta Raspberry Pi, conexión Wi-Fi y procesamiento de la cámara.

La correlación de imágenes utilizada entre la imagen patrón y la capturada, permitió determinar el nivel de similitud entre imágenes, de tal manera que se pueda determinar la eficiencia del ejercicio de rehabilitación ejecutada en una escala de 0 a 1, considerando como aceptable mayor a 0.5.

## **4.2.Recomendaciones**

El sistema operativo cargado a la tarjeta Raspberry Pi debe descargarse vía cable Ethernet ya que al realizar la descarga vía Wi-Fi puede existir un corte en la conexión y el archivo obtenido genera fallas en la instalación.

Para no perder la conexión Wi-Fi con la tarjeta Raspberry Pi se debe procurar no trabajar en áreas en las que concurren múltiples redes inalámbricas, debido a que el robot tiende a desconectarse de la red asignada.

Tener una iluminación óptima en el área donde se realizará los ejercicios de fisioterapia, para que la captura de imágenes sea adecuada para la operación del procesamiento de imágenes de tal manera que permita obtener resultados con una eficiencia mayor al 50%.

Para realizar las pruebas en la tarjeta Raspberry pi se debe realizar una conexión remota debido que al conectar los accesorios como teclado, mouse, monitor produce un calentamiento excesivo en el procesador provocando que la tarjeta sea inestable

## CAPÍTULO V

### 5. PROPUESTA

#### 5.1. Título de la Propuesta

Diseño y construcción de un robot para realizar rutinas de ejercicios de fisioterapia.

#### 5.2. Introducción

El Diseño y construcción de un robot para realizar rutinas de ejercicios de fisioterapia sirve para la aplicación en el campo de la medicina particularmente en la fisioterapia, para ello se monitoreara a través del robot a los pacientes que ejercicios realizaran para una recuperación rápida y adecuada, en las extremidades de brazos y piernas, se desarrolla por medio de la tarjeta Raspberry Pi 2 para una transmisión inalámbrica de video, imagen y datos, se puede utilizar el software Matlab 2015 para realizar procesos de transmisión vía UDP Y TCP de la Raspberry Pi 2 al controlador de servos Torobot para ejecutar movimientos en nuestro robot.

La Raspberry Pi es un ordenador básico con un sistema operativo, encargado de realizar aplicaciones de un computador, a su vez aplicar sistemas de robótica y de telecomunicaciones, con una eficiencia confiable y segura de utilización.

#### 5.3. Discusión

La investigación se enfoca en el Diseño y construcción de un robot para realizar rutinas de ejercicios de fisioterapia usando la tarjeta Raspberry Pi 2, el mismo que permite realizar varias aplicaciones en el desarrollo de las comunicaciones inalámbricas.

Se conocerá este micro computadora mostrándole qué se puede hacer, qué distribuciones existen, cómo desarrollar aplicaciones simples utilizando Python y cómo controlar dispositivos externos utilizando una interfaz GPIO.

Este proyecto se basa en el monitoreo y la transmisión de video en tiempo real del robot con los pacientes quienes realizaran ejercicios de fisioterapia mediante la utilización de del puerto serial TX de la Raspberry Pi 2 a la RX del Controlador de Servos Torobot que se encuentra conectado a la minicomputadora Raspberry Pi, para lo cual se ha elaborado un software que permite el procesamiento de datos para su presentación en una selección de

los ejercicios como Sentadillas, flexión de pie. Abducción de brazos y hombros , extensión de brazos y hombros implementado en el lenguaje Python.

La principal potencialidad de este minicomputador es su capacidad de comunicación con periféricos a través de sus diferentes puertos y en especial por el puerto de propósitos generales GPIO, que permite realizar aplicaciones para el control y manejo de dispositivos.

El procesamiento y análisis de Imágenes se a través de la captura de imágenes que realiza el robot por medio de la cámara Raspberry Pi y el programa desarrollado en Matlab - Guide el cual realiza la verificación del paciente cómo evoluciona en su recuperación ya sea correcta o incorrecta y al final se presenta los resultados realizados en varios días de la semana.

## CAPÍTULO VI

### 6. OBJETIVOS

#### 6.1. Objetivo General

Diseñar y construir un robot para la ejecución y monitoreo de rutinas de ejercicios de fisioterapia.

#### 6.2. Objetivos Específicos

- ✓ Programar el robot para monitorear la rutina de ejercicios de fisioterapia y realizar pruebas de las rutinas de ejercicios de fisioterapia de acuerdo al tipo de lesión que será ejecutado por el robot.
- ✓ Diseñar el interfaz de la PC para las diferentes rutinas de ejercicios de fisioterapia que realizara el robot.
- ✓ Evaluar mediante el software MATLAB el desarrollo de las rutinas de ejercicios de fisioterapia.

#### 6.3. Fundamentación Científico – Técnica

Contendrá una versión resumida y actualizada del estado del conocimiento en que se encuentra el tema específico de la propuesta.

El robot realiza los movimientos de fisioterapia e indica, si estos se están haciendo en forma correcta o incorrecta. A su vez, realiza el monitoreo instantáneo durante las terapias con los pacientes, de manera que el fisioterapeuta pueda tener un monitoreo continuo de la evolución de su paciente. El objetivo es que los pacientes tengan una gran facilidad de realizar rehabilitaciones con dispositivos inalámbricos como Tablet, teléfonos con tecnología Android el etc. De esta manera, el robot enviaría, en forma periódica y regular, a través de Internet los datos y la información recogida, al Hospital o centro de fisioterapia donde los especialistas irían tomando las decisiones pertinentes.

#### 6.4. Descripción de la Propuesta

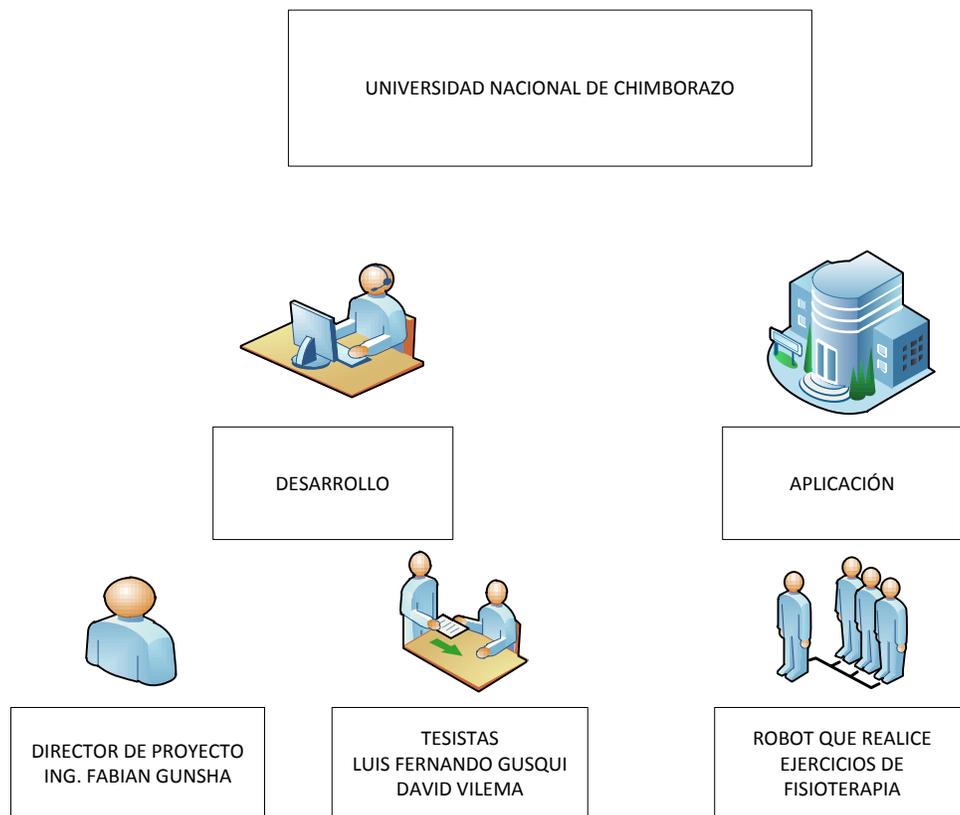
El Diseño y construcción de un robot para realizar rutinas de ejercicios de fisioterapia se encontrara dentro de las instalaciones que conforman un Hospital o centros de fisioterapia en la ciudad de Riobamba, este proyecto es un robot bípedo que posee una cámara de 5

Megapíxeles y una tarjeta Raspberry Pi 2 para realizar ejercicios mediante una comunicación serial, fabricado para esta finalidad concreta. Utiliza una red inalámbrica creada para ejecutar funciones de demostraciones y ejecuciones para la reacción de un paciente y la recuperación de un músculo o tendón que haya sufrido por alguna lesión.

El diagnóstico detallado es enviado a “un archivo que compare los ejercicios realizados por el paciente” para que los fisioterapeutas, sin importar dónde estén, puedan supervisar la recuperación del paciente.

### 6.5.Diseño Organizacional

Se establecera la estructura organica y funcional de la unidad administrativa que ejecutara la propuesta como se muestra en la figura



## CAPITULO VII

### 7. BIBLIOGRAFÍA

ALBERTO, S. G. (Noviembre de 2008). *Blogger*. Obtenido de Blogger: [www.blogger.com](http://www.blogger.com)

BAÑÓ, C. I. (2014). *DISEÑO, MONITORIZACIÓN Y CONTROL DE UN HEXÁPODO CON ROS*. Valencia.

Castillo, L. (2013). *Redes Inalambricas*. EE. UU.: AMNTEL GROUP.

Cristalab. (20 de Agosto de 2010). *Tutorial básico de Python*. Obtenido de <http://www.cristalab.com/tutoriales/tutorial-basico-de-python-c90356l/>

Defee, I. (2015). *SIGNAL AND IMAGE PROCESSING WITH MATLAB ON*. SIRIPHAT POMYEN.

Domínguez, C. G. (2015). *Aplicaciones orientadas a la domótica con Raspberry Pi*. Sevilla.

Integrativa, P. (2015). *Salud Terapia*. Obtenido de <http://www.saludterapia.com/glosario/d/32-fisioterapia.html#ixzz4lslhUImc>

Martínez., J. M. (2013). *Microcontroladores PIC*. Mc GrawHill.

Servo Motor. (s.f.).

Solofisio. (31 de Mayo de 2012). *Fisioterapia Traumatológica y Ortopédica*. Obtenido de <http://www.solofisio.com/especialidades/articulo/fisioterapia-traumatologica-y-ortopedica-en-que-consiste-26>

<https://unpocodejava.wordpress.com/2013/08/01/wireless-video-de-raspberry-pi-a-pc/>

<http://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/>

<http://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/>

[http://www.academia.edu/15306688/Manejo\\_de\\_una\\_Mano\\_Rob%C3%B3tica\\_desde\\_un\\_M%C3%B3dulo\\_Electr%C3%B3nico\\_Raspberry\\_Pi\\_B\\_con\\_Sistema\\_Operativo\\_Raspbian](http://www.academia.edu/15306688/Manejo_de_una_Mano_Rob%C3%B3tica_desde_un_M%C3%B3dulo_Electr%C3%B3nico_Raspberry_Pi_B_con_Sistema_Operativo_Raspbian)

<http://www.mathworks.com/help/supportpkg/raspberrypiio/ug/get-video-from-the-raspberry-pi-camera-board.html>

<http://www.mathworks.com/help/supportpkg/raspberrypiio/examples/working-with-raspberry-pi-camera-board.html>

<http://www.raspberryshop.es/guia-completa-raspberry-pi.php>

<http://www.hagaestiramientos.com/>

<http://www.electronicaestudio.com/raspberrypi.htm>

<https://unpocodejava.wordpress.com/2013/08/01/wireless-video-de-raspberry-pi-a-pc/>

<http://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie>

<http://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/>

<https://electrosome.com/uart-raspberry-pi-python/>

<http://www.instructables.com/id/Read-and-write-from-serial-port-with-Raspberry-Pi/>

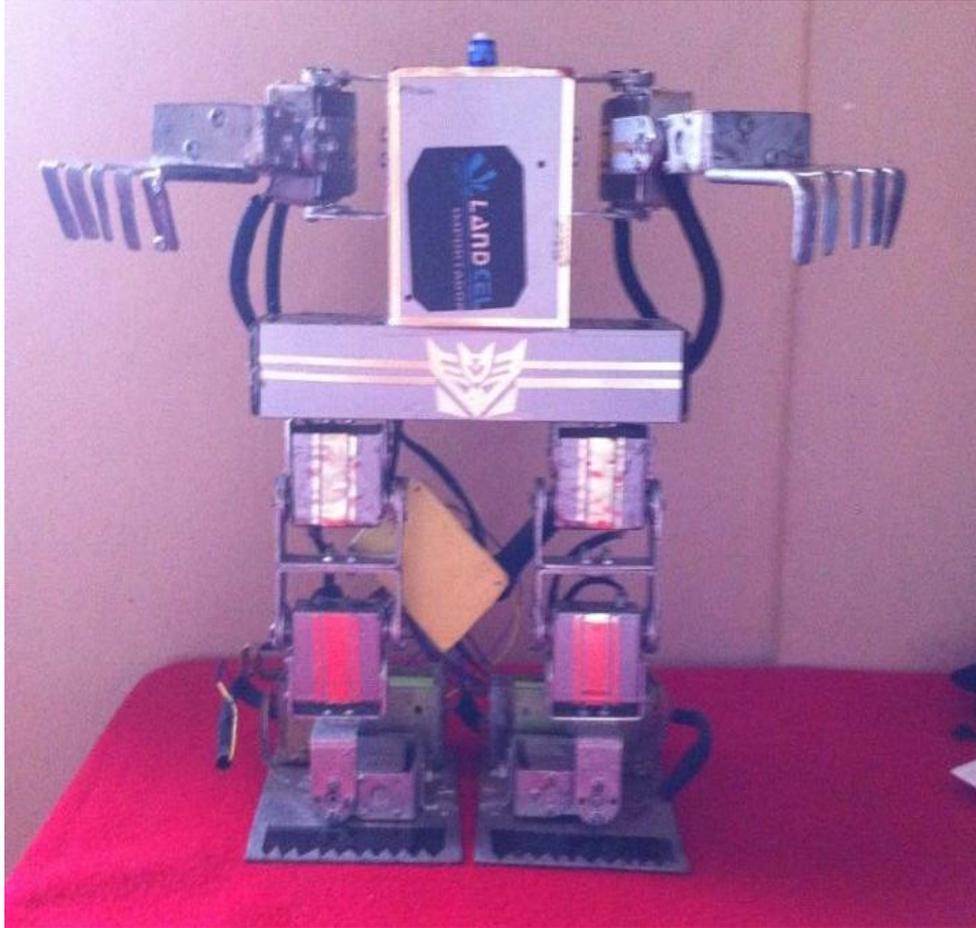
<http://www.mathworks.com/help/supportpkg/raspberrypiio/ug/get-video-from-the-raspberry-pi-camera-board.html>

<http://www.mathworks.com/help/supportpkg/raspberrypiio/examples/working-with-raspberry-pi-camera-board.html>



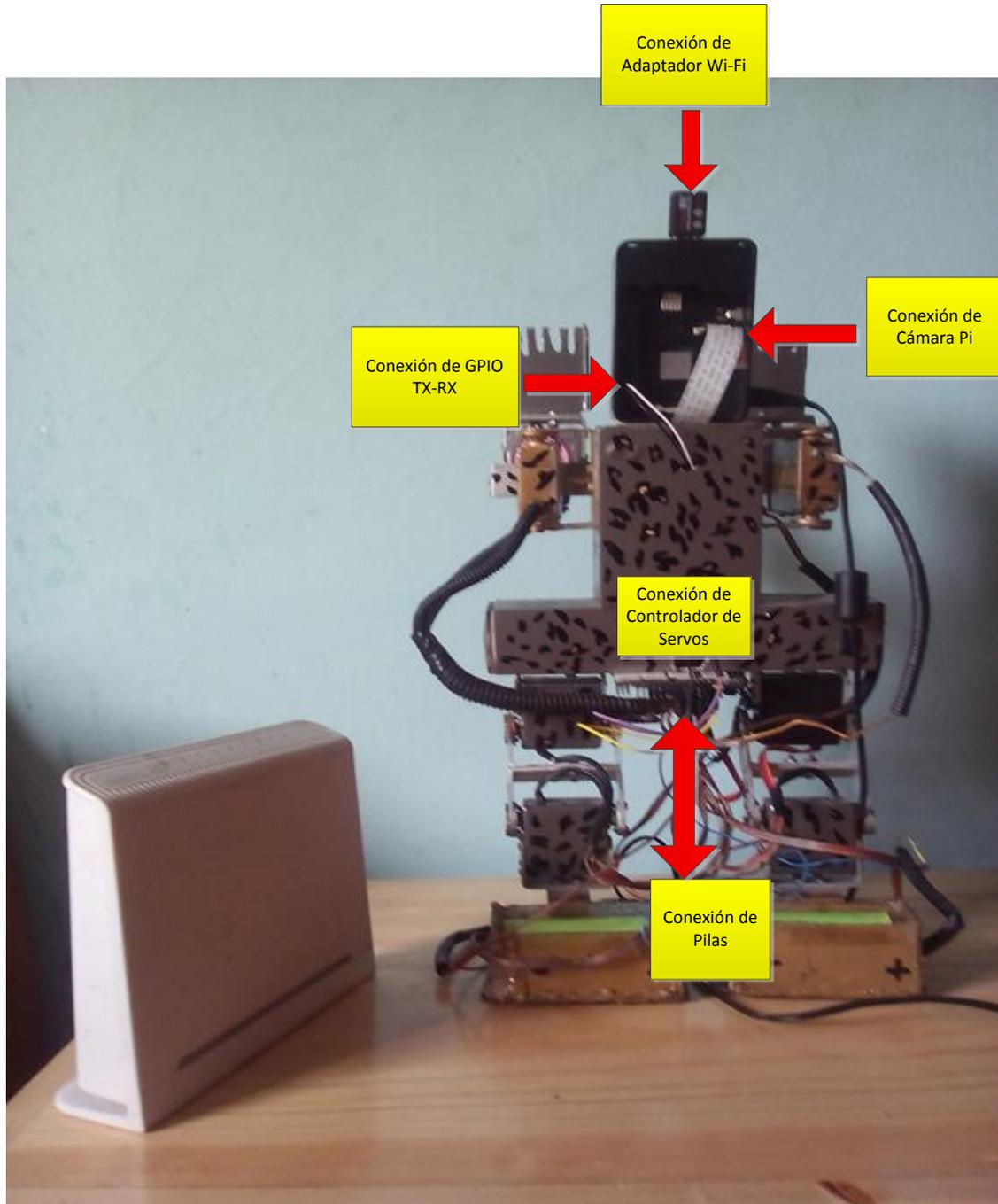
## ANEXO 2

### Construcción de robot



## ANEXO 3

### Implementación y Conexión de Servos, Raspberry y Baterías

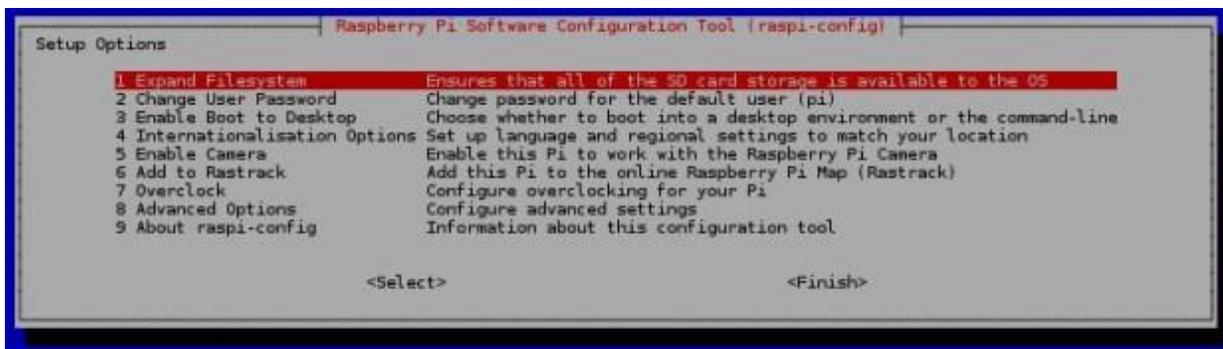


## ANEXO 4

### Configuración y Programación de Raspberry Pi 2

#### Menú principal

El primer menú que muestra contiene 9 diferentes opciones disponibles, a continuación revisaremos cada una de las opciones disponibles:



#### Opción 1 – Expandir el sistema de archivos (Expand Filesystem)

Esta opción permite expandir el sistema operativo para que utilice todo el espacio disponible en la tarjeta. Cuando se instala Raspbian “Wheezy” la imagen copiada en la tarjeta solo ocupa 2 GB, por lo tanto es necesario ejecutar esta opción para que todo el espacio de la tarjeta SD sea utilizado.

#### Opción 2 – Cambiar la contraseña del usuario Pi (Change User Password)

En el Raspberry Pi y en general en sistemas Linux existen diferentes tipos de usuario, los dos que vienen predeterminados por el sistema son los usuarios “root” y “pi”

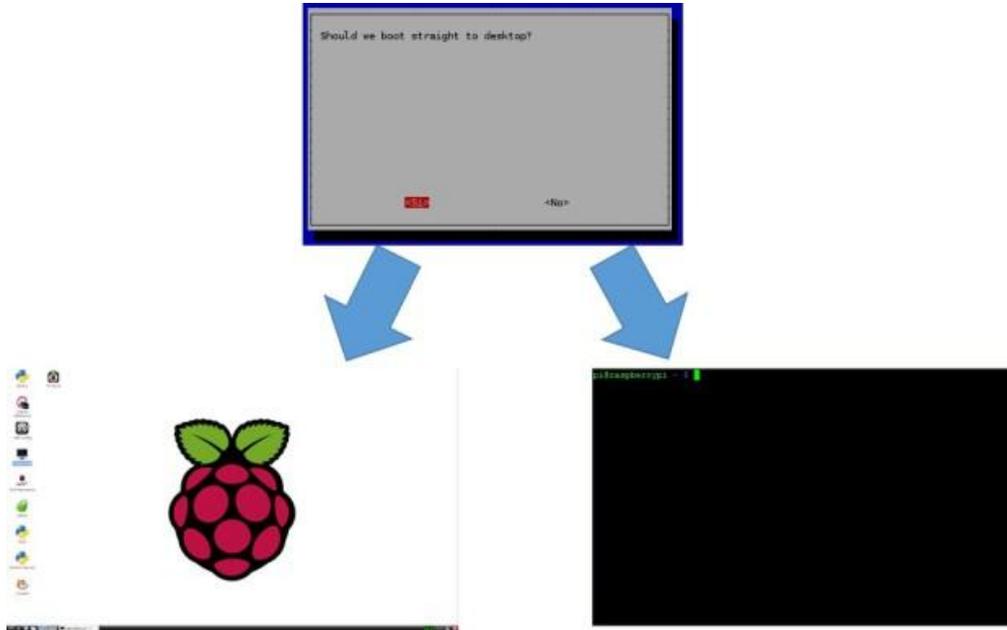
El más importante que es el administrador del sistema que se llama “root”, este tiene acceso privilegiado a todos los archivos, configuraciones y carpetas del sistema. El otro tipo de usuario son los comunes como lo es “pi”, este viene predeterminado con la contraseña “raspberrypi” por lo tanto cualquier persona podría acceder su sistema. Por eso, es recomendable cambiar la contraseña en esta opción. El sistema le solicitará que ingrese la nueva contraseña y que la repita nuevamente, al finalizar espere un mensaje como el siguiente:



### Opción 3. Activar el escritorio al iniciar (Enable Boot to Desktop)

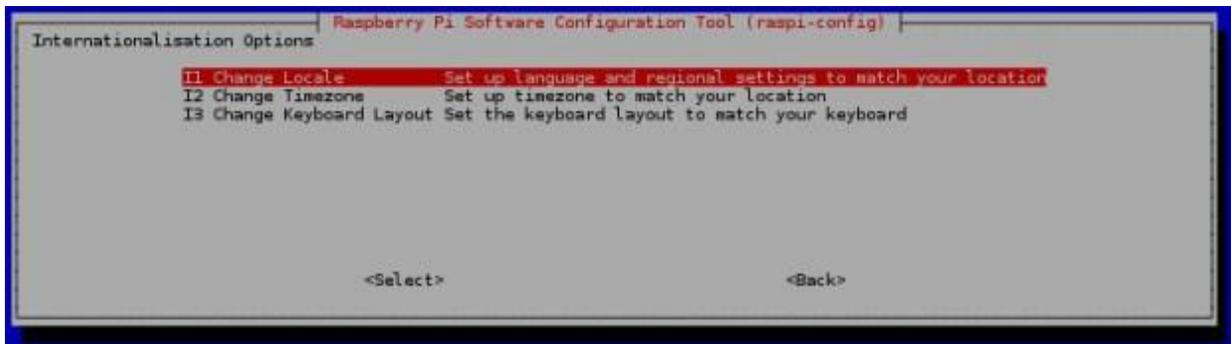
Esta opción permite que el Raspberry Pi después de iniciar el sistema, comience inmediatamente el escritorio modo gráfico o en línea de comando. En caso que inicie en modo de comando y después desee ingresar al modo gráfico solo ingrese el siguiente código:

1 Startx



### Opción 4. Opciones de internacionalización (Internationalisation Options)

Esta opción permite modificar el lenguaje del sistema operativo, la zona horaria y la distribución de su teclado. Para este ejemplo vamos a considerar que se encuentra en Colombia y tiene teclado en modo latinoamericano.



La opción I1 sirve para indicar donde se encuentra ubicado, esta opción configura el lenguaje del sistema operativo, los caracteres, la denominación de la moneda, etc. Busque y seleccione la el modo de codificación:

es\_CO. UTF-8 UTF-8

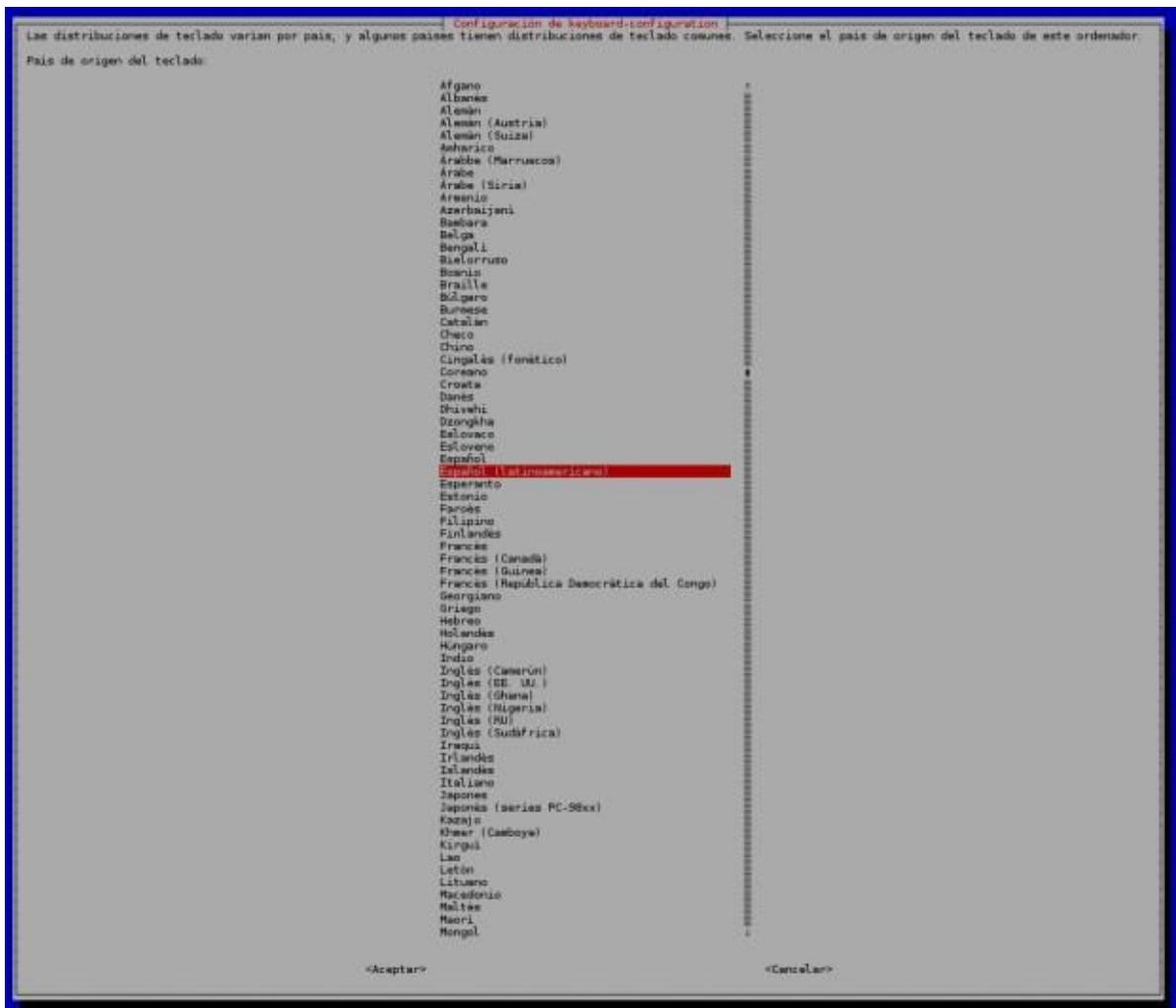




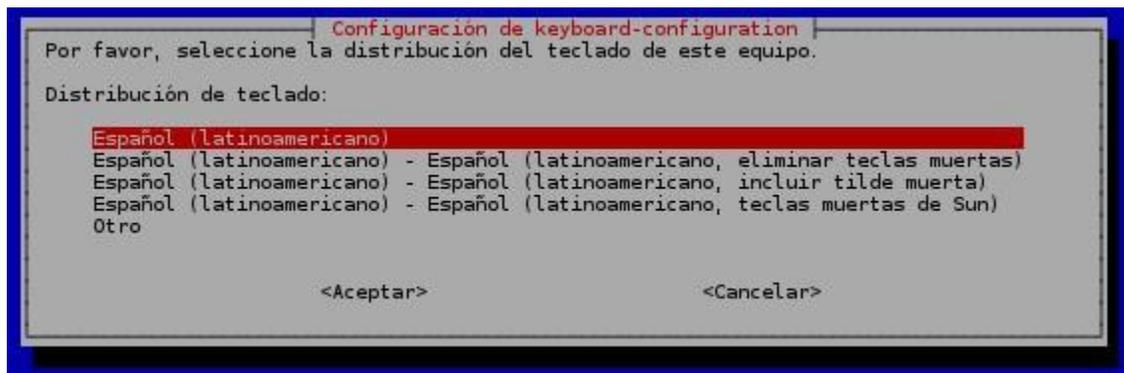
La opción A3 – le permitirá cambiar la configuración de su teclado, si la marca y tipo de su teclado no aparece, seleccione el predeterminado “PC genérico 105 teclas (intl)”.



Luego seleccione el idioma del teclado, si en la primera pantalla no aparece Español seleccione otro. En la siguiente ventana tendrá una lista de opciones seleccione “Español” o “Español (Latinoamericano)”



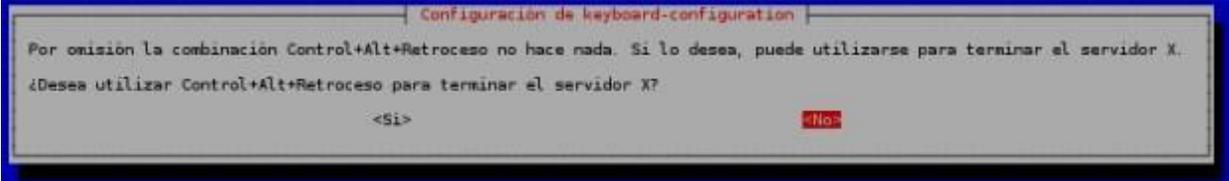
Luego seleccione la distribución de su teclado, se recomienda que utilice la primera opción solo “Español Latinoamericano”.



Las siguientes dos ventanas le permiten configurar la tecla AltGr izquierdo y derecho para funciones especiales, si su teclado posee este botón seleccione la primera opción para que ejecute las opciones predeterminadas.

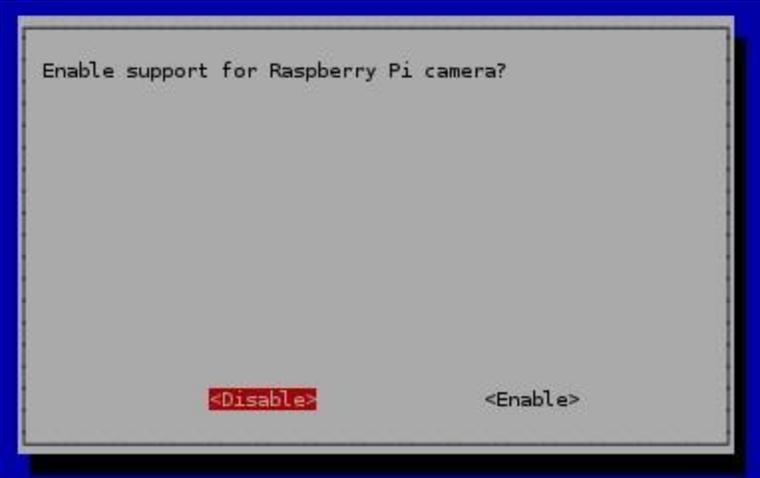


Por último el sistema solicitará que si desea activar Control+Alt+Retroceso para terminar el Servidor X (x server). Esto significa que si se encuentra en el modo gráfico y oprime esta combinación de teclas, el modo gráfico se cerrará y entrará a modo de comandos. Es recomendable dejar la opción predeterminada “No”.



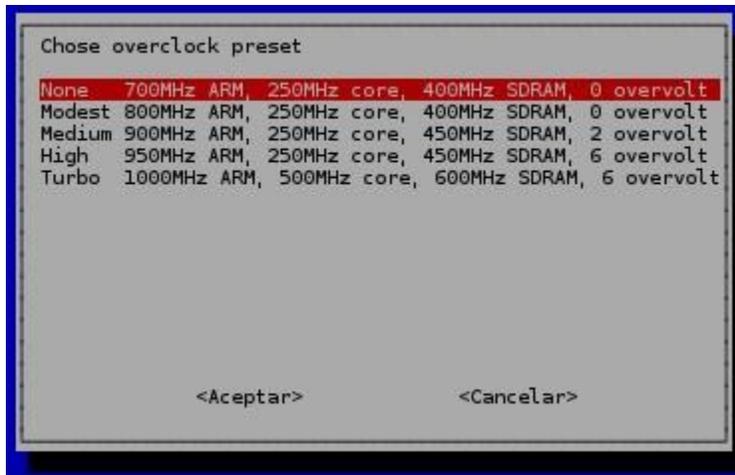
**Opción 5 – Activar la cámara (Enable camera)**

Esta opción sirve para dar soporte a la cámara de Raspberry Pi, esta opción permite activar el puerto para que haya comunicación entre la CPU y el controlador de la cámara.



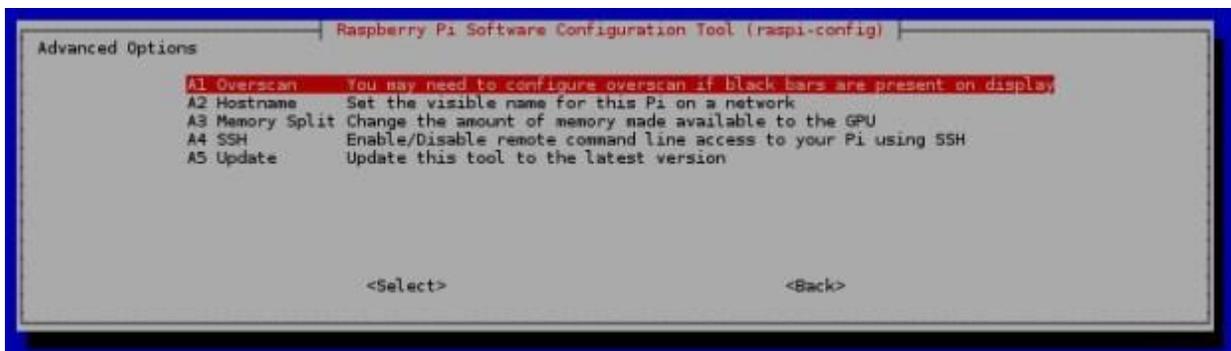
**Opción 6 – Overclockig**

Esta opción permite aumentar la velocidad del procesador. Tenga en cuenta lo siguiente al modificar la velocidad. Primero, la vida del dispositivo se puede disminuir considerablemente. Segundo, el dispositivo generará más calor, por lo tanto es recomendable tener disipadores en el procesador, en circuito de Ethernet y en regulador de energía. Tercero, va a necesitar una fuente de poder de mayor capacidad para que pueda compensar la nueva velocidad seleccionada. Se recomienda dejar la predeterminada, pero si quiere tener más poder de cómputo siéntase libre de escoger la que más le convenga.



### Opción 7. Opciones avanzadas (Advanced Options)

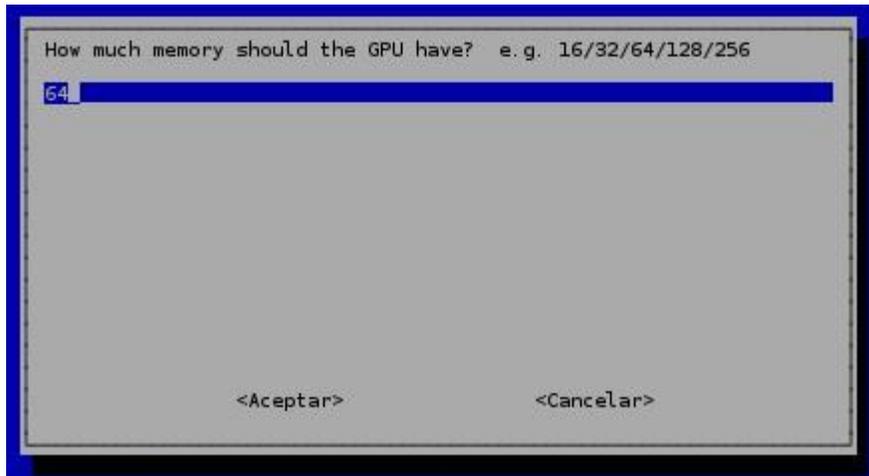
Esta opción presenta un otro submenú con las siguientes opciones.



La opción A1 overcan sirve para borrar las líneas negras en algunos monitores o televisores. La opción A2 Hostname, sirve para identificar su Raspberry Pi en su red local, solamente utilice letras y números. Tenga en cuenta que el sistema diferencia mayúsculas y minúsculas, por lo tanto recuerde como escribe este nombre.



La opción A3 – Distribución de la memoria (Memory Split) le permite seleccionar la cantidad de memoria compartida entre la CPU y la unidad de gráficos (GPU), el modelo B versión 2.0 cuenta con 512 MB en total. El predeterminado es 64 MB para la memoria de vídeo, si piensa ejecutar aplicaciones que requieren alto procesamiento gráfico como ver películas o ver imágenes aumente el valor. Tenga en cuenta que al aumentar la memoria de vídeo y disminuir la del procesador, este se vuelve un poco más ‘lento’ para ejecutar otras tareas, es el precio que toca pagar por ejecutar mejor los gráficos.



La opción A4 – Activar SSH (Enable SSH) se utiliza para acceder el Raspberry Pi remotamente desde un cliente SSH. SSH significa “Secure SHell” el cual es una forma segura de conectarse al Raspberry Pi a través de la red, es recomendable activar esta opción, ya que con esto no necesitará utilizar ni un monitor, ni teclado, ni mouse adicionales para poder controlar su dispositivo. En futuros tutoriales les enseñaremos como utilizar esta opción.

Y la última opción A5 – Actualizar (update) se utiliza para que descargue una actualización del sistema, si ya se encuentra conectado a la res lo puede ejecutar inmediatamente. Si hay nuevas versiones de las librerías o programas se descargarán e instalarán las últimas versiones. O si prefiere lo puede hacer más adelante con el siguiente comando:

```
1 sudo apt-get update
```

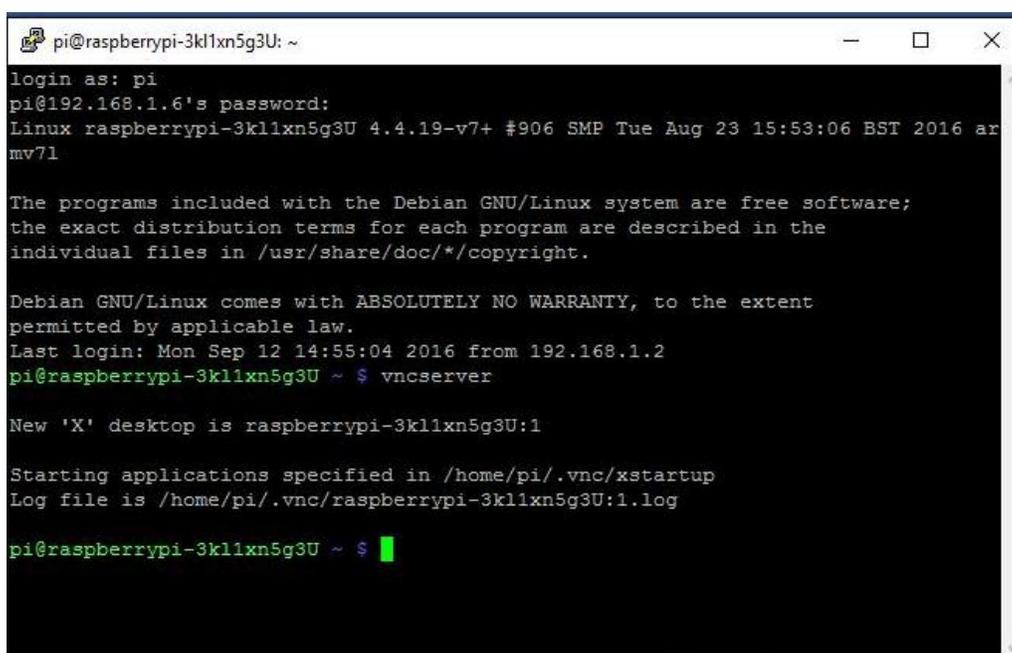
## ANEXO 5

### Conexión Remota PC – Raspberry Pi

#### Instalación VNC en el Raspberry Pi

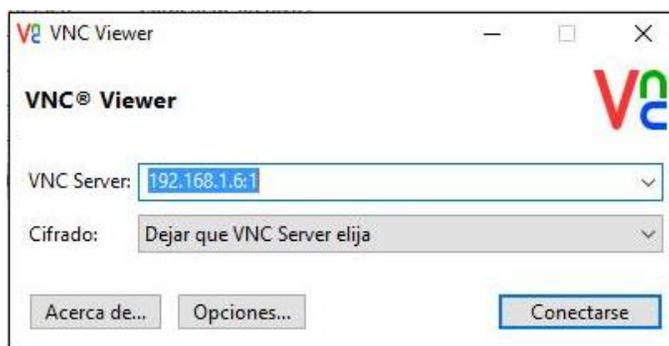
Lo primero que se necesita es instalar el servidor VNC en el Raspberry Pi, este se encargara de cargar el escritorio remoto. Para esto ingrese a la terminal de su Raspberry Pi a través de SSH usando Putty o en la terminal local si no tiene configurado el acceso remoto todavía.

Se procede a ejecutar el comando vncserver

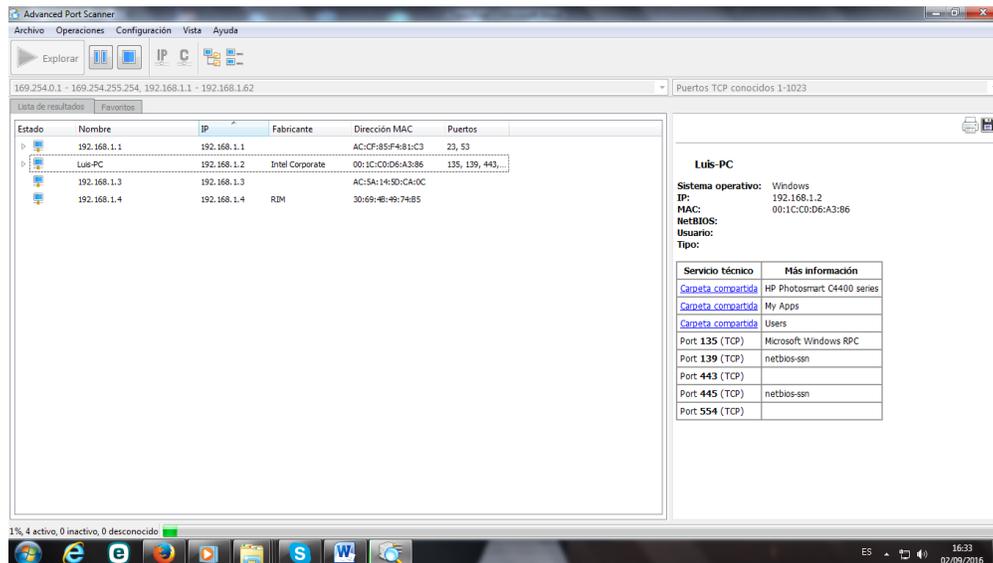


```
pi@raspberrypi-3kl1xn5g3U: ~  
login as: pi  
pi@192.168.1.6's password:  
Linux raspberrypi-3kl1xn5g3U 4.4.19-v7+ #906 SMP Tue Aug 23 15:53:06 BST 2016 ar  
mv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Mon Sep 12 14:55:04 2016 from 192.168.1.2  
pi@raspberrypi-3kl1xn5g3U ~ $ vncserver  
  
New 'X' desktop is raspberrypi-3kl1xn5g3U:1  
  
Starting applications specified in /home/pi/.vnc/xstartup  
Log file is /home/pi/.vnc/raspberrypi-3kl1xn5g3U:1.log  
  
pi@raspberrypi-3kl1xn5g3U ~ $ █
```

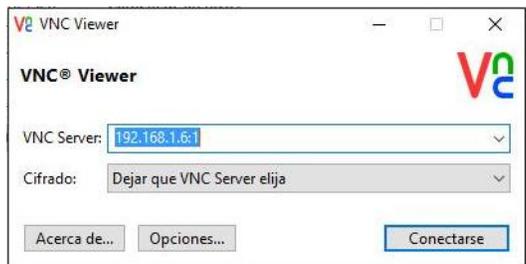
Una vez realizado el comando anterior, es necesario iniciar la aplicación VNC



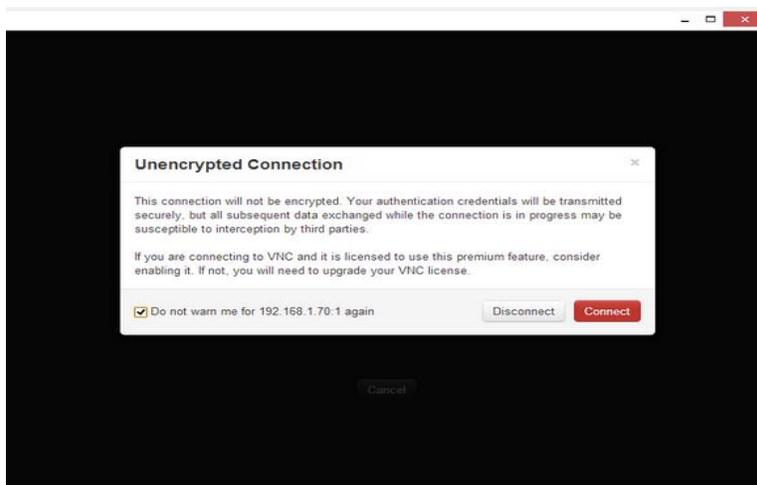
Para obtener la dirección IP de tu servidor procedemos a ejecutar el software Advanced Port Scanner y obtenemos la dirección IP de tu Servidor:



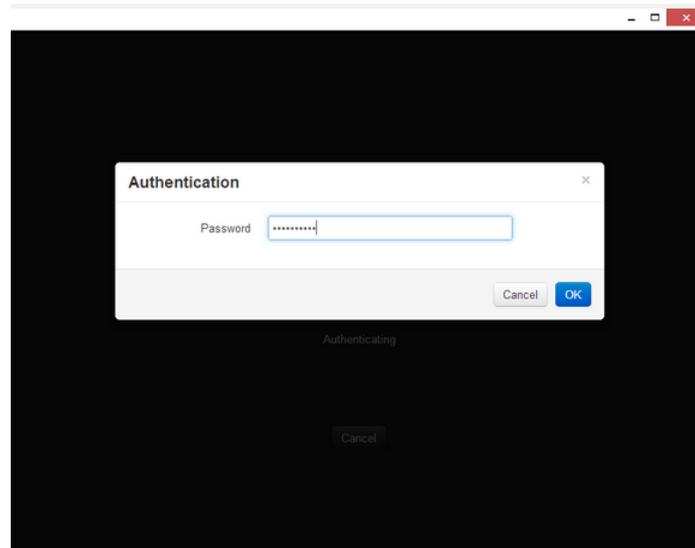
En la primera pantalla en ingresar la dirección IP de su Raspberry Pi (aquí se ha usado la conexión Wi-Fi, entonces la dirección IP es 192.168.1.6) seguido del número de la pantalla que se ha creado.



Luego aparecerá otra ventana acerca de encriptar la conexión, ignore este mensaje selección no volver a mostrar.

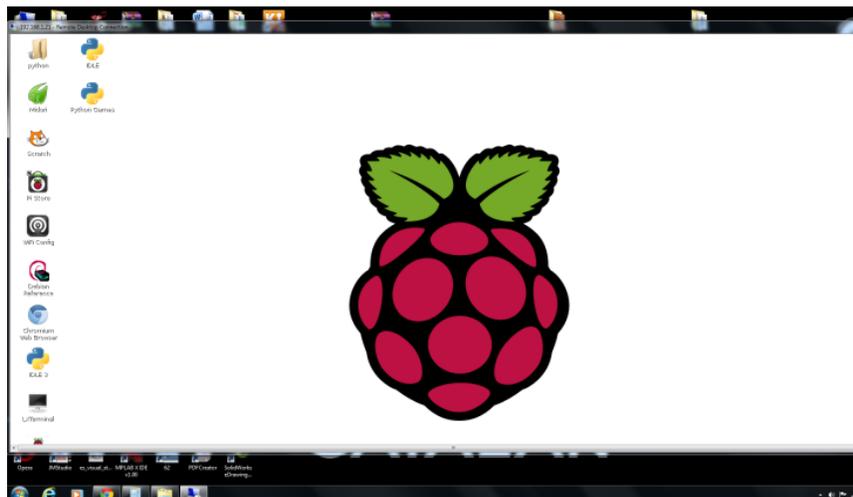


Por último ingrese la contraseña creada para el VNC server.



De este modo tendrá ahora acceso al escritorio remoto de Raspbian.

Una vez la contraseña ha sido verificada, ahora tendrá acceso remoto al escritorio.



## ANEXO 6

### Programación en Python para ejecutar ejercicios de Fisioterapia

```
'''
EJERCICIOS DE REHABILITACION PARA LOS BRAZOS Y PIERNAS

>> Ejercicio 1 : A
>> Ejercicio 2 : B
>> Ejercicio 3 : C
>> Ejercicio 4 : D
>> Ejercicio 5 : E
>> Ejercicio 6 : F
>> Ejercicio 7 : G

Seleccione el ejercicio a realizar
'''
print __doc__

import time
import serial
i=0
ser=serial.Serial("/dev/ttyAMA0")
ser.baudrate=9600

import subprocess
#Inicio de funciones
def A():
    print "Ejecutando ejercicio 1"

    ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")          ##Parada del robot
    time.sleep(1)

    for i in range(1,10):
##Primer ejercicio cunclillas

    ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")
        time.sleep(1)

    ser.write("#1P2478#2P2144#3P2011#4P833#5P856#6P2456#7P2456#8P1544#9P2500#
10P1211#11P1544T1000\r\n")
        time.sleep(1.1)
    else:

    ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")          ##Parada del robot
        print "FIN DEL EJERCICIO 1"

def B():
    print "Ejecutando ejercicio 2"
##Cuadrada para el segundo ejercicio
```

```

ser.write("#1P2322#2P1500#3P1411T1000\r\n")
time.sleep(1.2)

ser.write("#6P2223T1000\r\n")
time.sleep(1.2)

ser.write("#1P2500T1000\r\n")
time.sleep(1)

ser.write("#2P856#3P1122T1000\r\n")
time.sleep(1.2)

    for i in range(1,10):
##Repeticion del segundo ejercicio
        ser.write("#1P2500#2P1344#3P767T1000\r\n")
        time.sleep(1.3)

        ser.write("#2P833T1000\r\n")
        time.sleep(1)
    else:

ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")    ##Parada del robot
    print "FIN DEL EJERCICIO 2"

def C():
    print "Ejecutando ejercicio 3"

    for i in range(1,10):
##Repeticion del tercer ejercicio
        ser.write("#7P2433T1000\r\n")
        time.sleep(1.3)

        ser.write("#7P811#9P1500#10P500T1000\r\n")
        time.sleep(1.3)
    else:

ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")    ##Parada del robot
    print "FIN DEL EJERCICIO 3"

def D():
    print "Ejecutando ejercicio 4"

    for i in range(1,10):
##Repeticion del cuarto ejercicio
        ser.write("#7P1400#10P2300T1000\r\n")
        time.sleep(1.3)

        ser.write("#7P1400#10P100T1000\r\n")
        time.sleep(1.3)
    else:

ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")    ##Parada del robot

```

```

        print "FIN DEL EJERCICIO 4"

def E():
    print "Ejecutando ejercicio 5"

    for i in range(1,10):
        ser.write("#8P2500#10P2300T1000\r\n")
        time.sleep(1.3)

        ser.write("#8P500#10P544T1000\r\n")
        time.sleep(1.3)
    else:

ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")    ##Parada del robot
    print "FIN DEL EJERCICIO 5"

def F():
    print "Ejecutando ejercicio 6"

    for i in range(1,10):
        ser.write("#7P1433#9P1522T1000\r\n")
        time.sleep(1.5)

        ser.write("#7P2433#9P2500T1000\r\n")
        time.sleep(1.5)
    else:

ser.write("#1P2478#2P1500#3P1411#4P1456#5P1500#6P2456#7P2478#8P1544#9P250
0#10P1211T500\r\n")    ##Parada del robot
    print "FIN DEL EJERCICIO 6"

def G():
    print "Salir"
    ser.write("#6P800T1000\r\n")
    print "FIN DE LA SESION"
#Inicio de programa

while True:
    '''
EJERCICIOS DE REHABILITACION PARA LOS BRAZOS Y PIERNAS

>> Ejercicio 1 : A
>> Ejercicio 2 : B
>> Ejercicio 3 : C
>> Ejercicio 4 : D
>> Ejercicio 5 : E
>> Ejercicio 6 : F
>> Ejercicio 7 : G

Seleccione el ejercicio a realizar
'''
    x = input()
    y=x()

print "Adios"

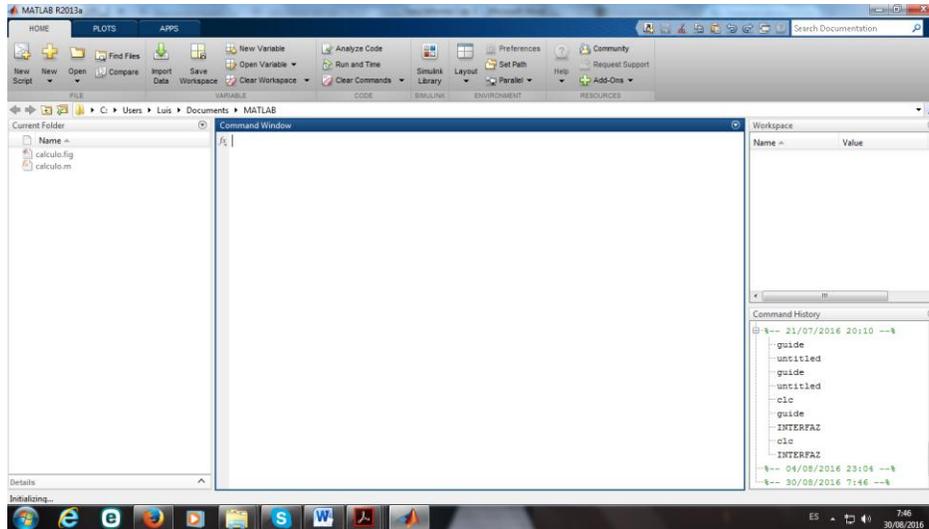
```

## ANEXO 7

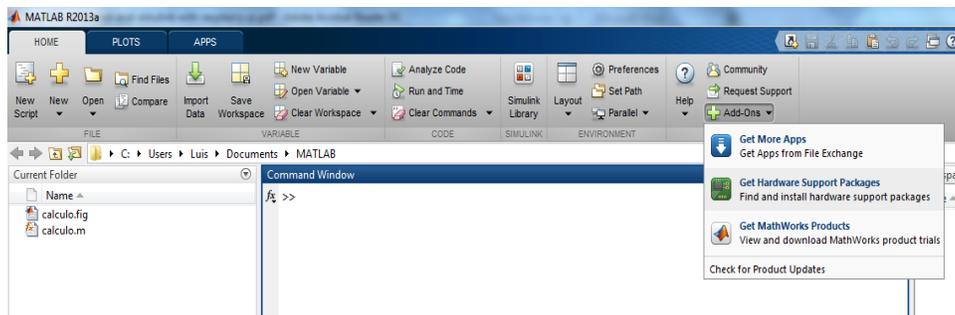
### Programación en Matlab – Simulink

#### Configuración en MATLAB y Simulink paquetes de apoyo para Raspberry Pi

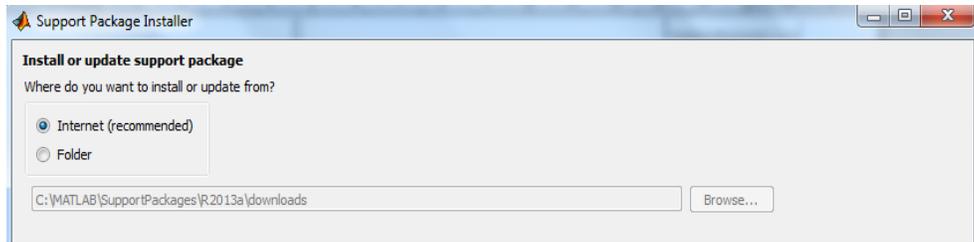
##### Iniciar Matlab



Iniciar el instalador para el soporte de paquetes  
Hacer click en Get Hardware Support Packages

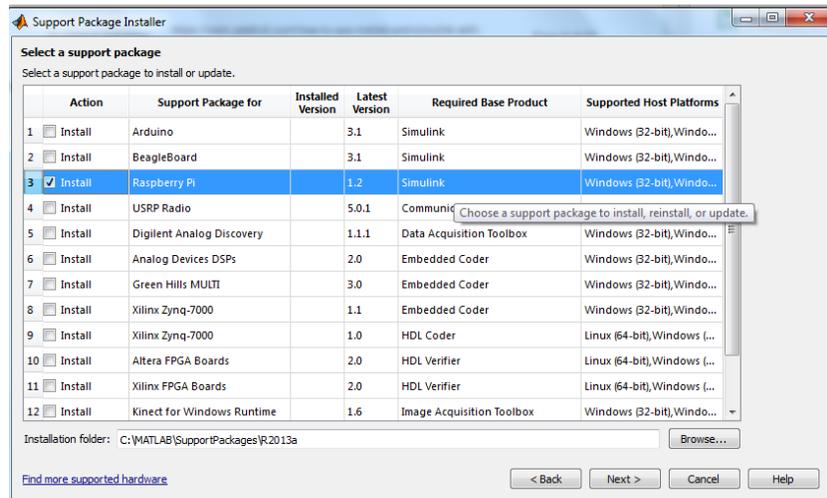


Seleccionar  
"Install Internet Recommended" como fuente para la instalación de los módulos de apoyo

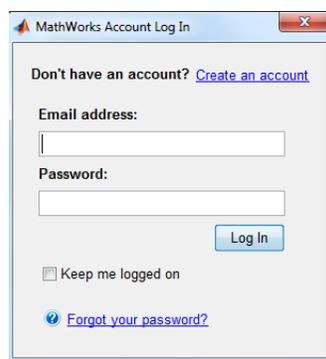


Selecciona Raspberry pi para el soporte de paquetes

Hacer click en Siguiente para ver una lista de los paquetes de soporte y seleccione Raspberry Pi de la lista para instalar tanto los paquetes de soporte a la vez.

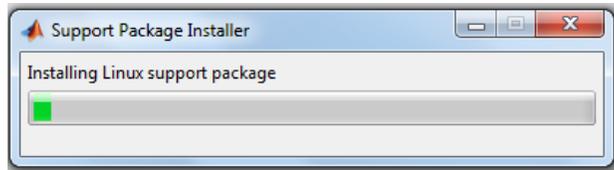


Iniciar una cuenta en MathWorks to Account



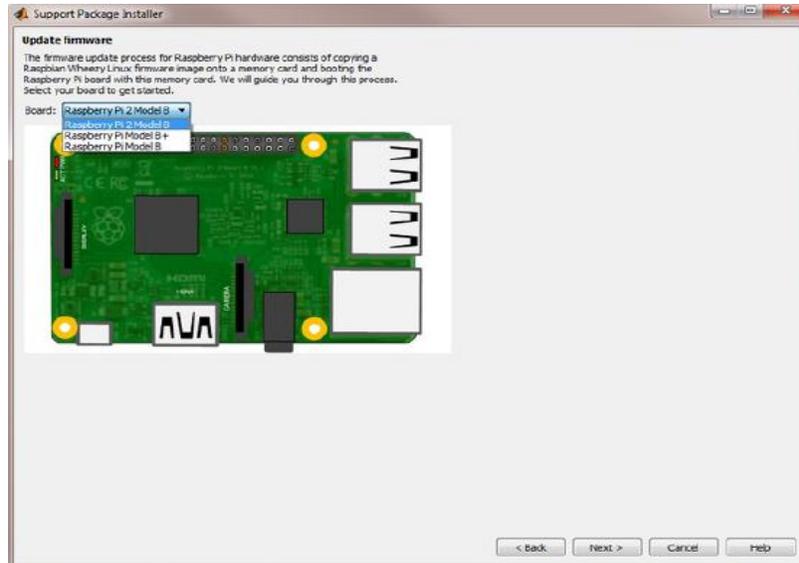
Continuar y completar la instalación.

Aceptar el contrato de licencia en la siguiente pantalla y haga clic en Siguiente en las pantallas siguientes a finalizar la instalación tanto de MATLAB y Simulink para el paquete de compatibilidad con Raspberry Pi.

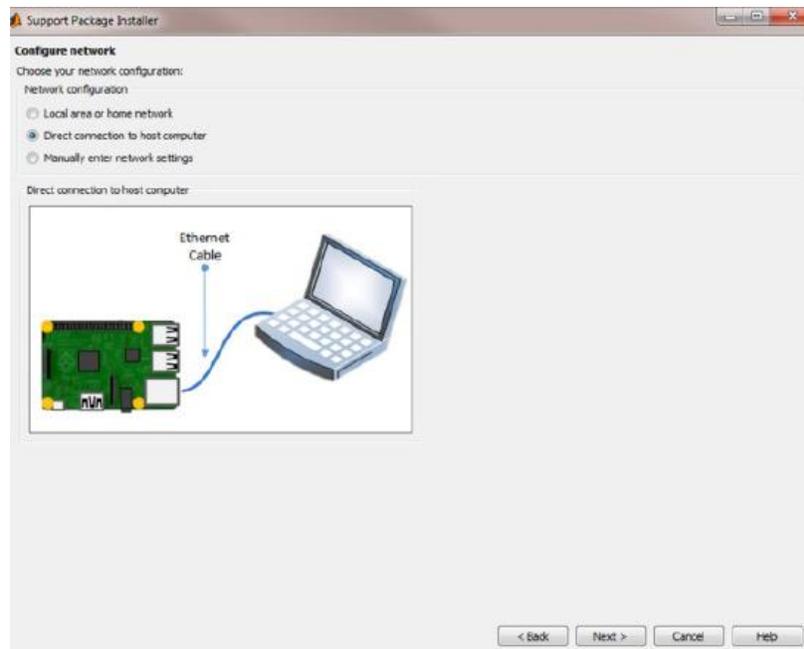


Actualización del firmware.

Haga clic en Siguiente y en la página Actualización del firmware seleccionar la junta adecuada (para este tutorial, se elige Frambuesa Pi Modelo B 2).



Haga clic en Siguiente y en la pantalla de configuración de red, seleccione la conexión directa al ordenador central (para seguir junto con este tutorial).



## Seleccione la unidad

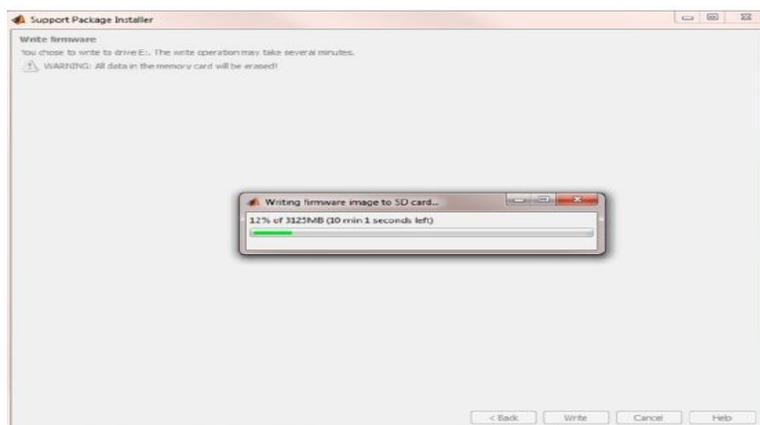
En la siguiente ventana "Seleccione una unidad", las tarjetas MicroSD que son detectados por MATLAB se mostrarán en un formato de lista.

Si la tarjeta de memoria MicroSD no consigue detectado por MATLAB, pero es detectado por el sistema operativo – cerca MATLAB y reiniciar MATLAB como administrador. Para continuar con el proceso, el targetupdater comando puede utilizarse en MATLAB.



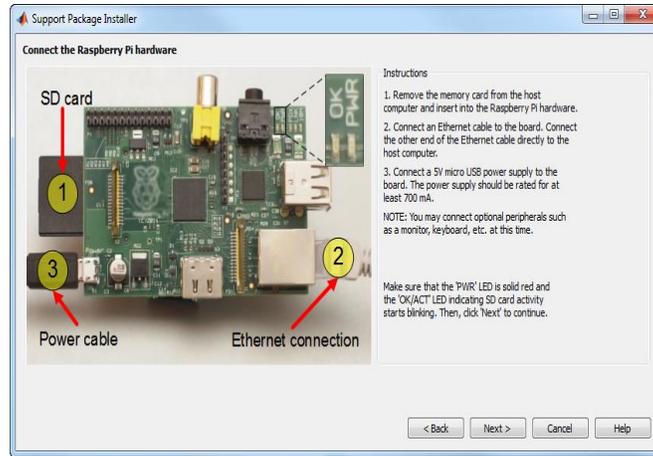
## Escritura de firmware

En la siguiente ventana, seleccione la opción de escritura para borrar los elementos existentes en la tarjeta de memoria y parpadea la última versión del firmware que se necesita por el paquete de soporte.

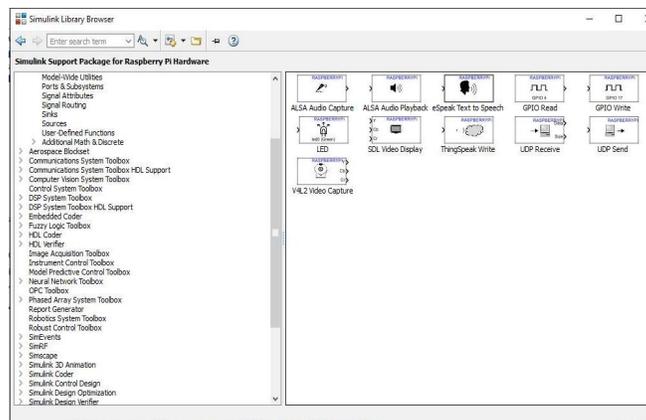


Continuar y completar la instalación

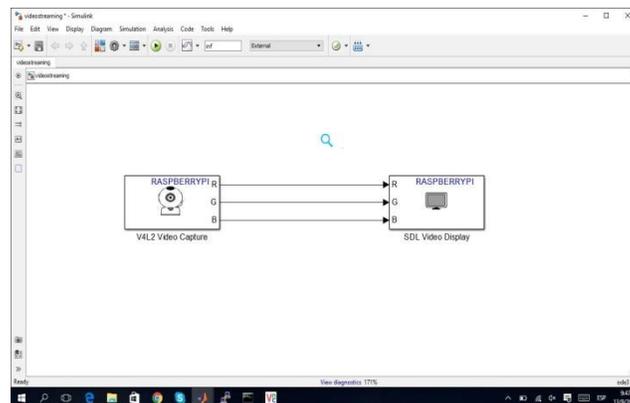
Haga clic en Siguiente en las pantallas siguientes para finalizar la instalación, tanto para MATLAB y Simulink Paquete de apoyo para Raspberry Pi.



Luego se obtiene las herramientas de Raspberry Pi en las librerías de Matlab 2015.



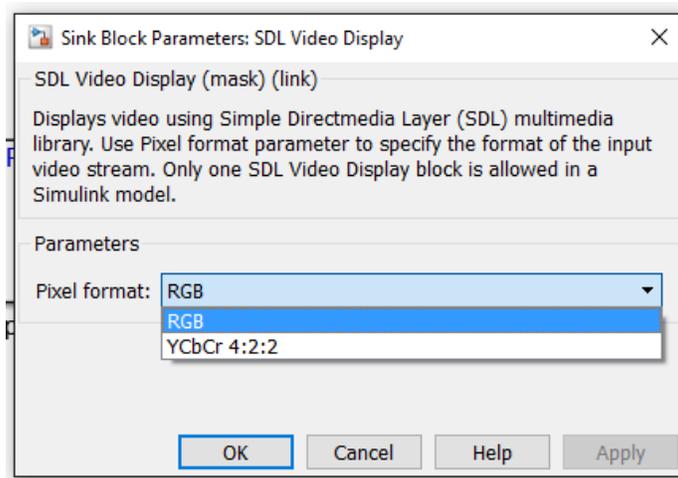
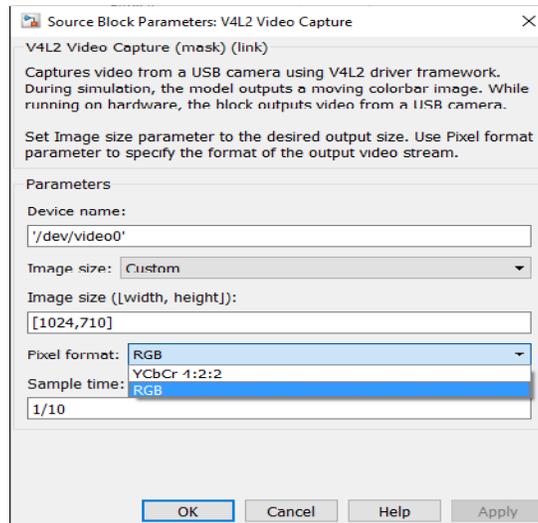
Luego se procede a realizar a la ejecución de la Transmisión de Video con las Librerías de Raspberry Pi en Simulink Matlab



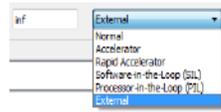
Finalmente se procede a configurar los parámetros de Transmisión

V4L2 Video Capture se encarga de activar la cámara del Raspberry Pi para la captura de video e imágenes, su configuración se realiza por medio de pixeles o la calidad de imagen que se desea obtener.

SDL Video Display es por donde se mostrara el tamaño y calidad de imagen ya sea en pixeles o en colores de RGB.



Para la ejecución de un modelo de Simulink en el modo externo, se selecciona el modo externo de un menú desplegable como se muestra en la Figura (a). Introducir una ejecución deseada tiempo en un espacio junto a la opción de modo. El valor predeterminado es 10. En este caso, es establecer como inf para indicar el tiempo infinito en funcionamiento. El proceso de simulación se inicia haciendo clic en un botón de ejecución se muestra en la Figura (b). El software preparará un modelo para ejecutar y mostrar el resultado en la pantalla de un ordenador si está conectado.

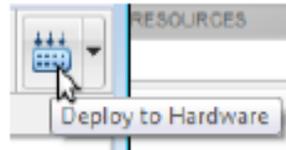


(a) External mode option



(b) Run button

Para la implementación de un modelo de Simulink en Raspberry Pi, después de confirmar la información sobre una ventana de configuración, un modelo puede ser desplegado en el tablero haciendo clic en un botón de hardware para implementar muestra en la Figura los modelo de Simulink se incluirán en la carpeta por defecto de la junta.



## ANEXO 8

### Programación en Matlab – Guide para el Análisis de Imágenes

```
function varargout = todo_incorporado_completo(varargin)

% TODO_INCORPORADO_COMPLETO MATLAB code for todo_incorporado_completo.fig

%     TODO_INCORPORADO_COMPLETO, by itself, creates a new
%     TODO_INCORPORADO_COMPLETO or raises the existing

%     singleton*.

%     H = TODO_INCORPORADO_COMPLETO returns the handle to a new
%     TODO_INCORPORADO_COMPLETO or the handle to

%     the existing singleton*.

%
%     TODO_INCORPORADO_COMPLETO('CALLBACK', hObject,eventData,handles,...) calls
%     the local

%     function named CALLBACK in TODO_INCORPORADO_COMPLETO.M with the
%     given input arguments.

%     TODO_INCORPORADO_COMPLETO('Property','Value',...) creates a new
%     TODO_INCORPORADO_COMPLETO or raises the

%     existing singleton*. Starting from the left, property value pairs
%     are

%     applied to the GUI before todo_incorporado_completo_OpeningFcn
%     gets called. An

%     unrecognized property name or invalid value makes property
%     application

%     stop. All inputs are passed to
%     todo_incorporado_completo_OpeningFcn via varargin.

%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only
%     one

%     instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
% todo_incorporado_completo

% Last Modified by GUIDE v2.5 15-Sep-2016 22:50:26

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;
```

```

gui_State = struct('gui_Name',      mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn',
@todo_incorporado_completo_OpeningFcn, ...
                  'gui_OutputFcn',
@todo_incorporado_completo_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',  []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT

% --- Executes just before todo_incorporado_completo is made visible.

function todo_incorporado_completo_OpeningFcn(hObject, eventdata,
handles, varargin)

% This function has no output args, see OutputFcn.

% hObject    handle to figure

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% varargin   command line arguments to todo_incorporado_completo (see
VARARGIN)

% Choose default command line output for todo_incorporado_completo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes todo_incorporado_completo wait for user response (see
UIRESUME)

% uiwait(handles.figure1);

```

```

% --- Outputs from this function are returned to the command line.

function varargout = todo_incorporado_completo_OutputFcn(hObject,
eventdata, handles)

% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

function edit_simfile_Callback(hObject, eventdata, handles)
% hObject     handle to edit_simfile (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit_simfile as text
%           str2double(get(hObject,'String')) returns contents of
edit_simfile as a double

% --- Executes during object creation, after setting all properties.
function edit_simfile_CreateFcn(hObject, eventdata, handles)
% hObject     handle to edit_simfile (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject     handle to pushbutton1 (see GCBO)
% eventdata   reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)

[input_file,pathname] = uigetfile( ...
    {'*.mdl','mdl Files (*.mdl)';...
    '*.*', 'All Files (*.*)'}, ...
    'Select files', ...
    'MultiSelect', 'on');

if pathname == 0
    return
end

%gets the current data file names inside the listbox
inputfile= fullfile(pathname,input_file);
current_folder=strcat(cd,'\');
mdlname=strrep(inputfile,current_folder,'');
mdlname=strrep(mdlname,'.mdl','');

%updates the gui to display all filenames in the listbox
set(handles.edit_simfile,'String',mdlname);
guidata(hObject, handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
handles=guidata(hObject);
set(handles.edit_simfile,'String','');
guidata(hObject, handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton3 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

```

```

flag=get(hObject,'string');
if strcmp(flag,'RUN')==1
set_param(handles.modelname,'SimulationCommand','Start');
set(hObject,'string','STOP');
else
set_param(handles.modelname,'SimulationCommand','Stop');
set(hObject,'string','RUN');
end
guidata(hObject,handles);
% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
modelname=get(handles.edit_simfile,'string');
if isempty(modelname)
    errordlg('You have not input the model name');
end
checkload=~isempty(find_system('type','block_diagram','name',modelname));
if checkload==0
    try
        load_system(modelname);
    catch
    end
end
block_V4L2videocapture=sprintf('%s/V4L2videocapture',modelname);
%Create handles so that they can be accessed across callback.
handles.modelname=modelname;
handles.block_V4L2videocapture=block_V4L2videocapture;
guidata(hObject,handles)

```

```

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global mypi;
global myCam;
global y;
global q;
global t;
global guardardat;

q=1;
y=1;
mypi=raspi();
myCam = cameraboard(mypi,'Resolution', '640x480');
% for i=1:4
% im3 = snapshot(myCam);           %%%captura
para mejorar el brillo de las siguientes imagenes
% pause(5);
% end

load('C:\Users\Luis Fernando\Desktop\INTERFAZ\datosemana.mat')
guardardat;
t=t+1;
%%no se visualiza
%%I = snapshot(myCam);
%%pause(5);
for n=1:5
nom=['imagen',num2str(1)];
ext='.jpg';
nombre=strcat(nom,ext);
pause(5);

```

```

im = snapshot(myCam);
imwrite(im, nombre);
end
for n=1:12
nom=['imagen',num2str(n)];
ext='.jpg';
nombre=strcat(nom,ext);
pause(5);
im = snapshot(myCam);
imwrite(im, nombre);
figure, imshow(im);
end
% clear mypi
% clear myCam
% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global myCam;
global may;
global q;
global maytotal;
global guardardat;
global t;
localize=0;
z=1;
for m=1:2
%%%%%%%% Analisis de la imagen patron con las imagenes tomadas al momento
%%%%%%%% del ejercicio

```

```

if z==1
%
    mypi=raspi();
%
    myCam = cameraboard(mypi,'Resolution', '640x480');
im1 = snapshot(myCam);
imwrite(im1, 'patron.jpg');
g1=imread('patron', 'jpeg');
g=rgb2gray(g1);
F= fft2(g);
d(13)=mean(max(corr(F,F)));
rl(13)=real(d(13));
for j=1:12
    nom=['imagen',num2str(j)];
    I=imread(nom,'jpeg');
    g1=rgb2gray(I);
    G= fft2(g1);
    d(j)=mean(max(corr(G,F)));
    rl(j)=real(d(j));
    %%figure, imshow(I);
end
end
%%% grafico tipo stem
x=1:13;
axes(handles.axes11),h=stem(x,rl,'fill','--');
%%figure,h=stem(x,rl,'fill','--');
set(get(h,'BaseLine'),'LineStyle',':')
set(h,'MarkerFaceColor','red')
%%% final de la grafica tipo stem
may(m)=rl(1);
for j=1:12
    if may(m)<rl(j)

```

```

        may(m)=rl(j);
        localize=j;
    end
end
maytotal(q)=rl(localize);
guardardat(t)=maytotal(q);
q=q+1;

        t=t+1;

nom=['imagen',num2str(localize)];
Imgcoincidente=imread(nom,'jpeg');
axes(handles.axes9)                                %%%% mostrar imagen
en el axes 9
        imshow(Imgcoincidente)
        %%figure, imshow(Imgcoincidente)
                                axes(handles.axes10)                                %%%%
mostrar imagen en el axes 10
        imshow(im1)
        %%figure, imshow(im1)
        pause(15);

%%%%%%%%%% Fin del analisis con la imagen patron
end;

%%guardardat=maytotal;
guardardat(t)=10;

save ('C:\Users\Luis
Fernando\Desktop\INTERFAZ\datosemana.mat','guardardat','t');

q=q-1;
% %%for p=1:q
%     maytotal(p)
%     %%%axes(handles.axes12),plot(p,maytotal(p),'.')
%     xlim([0 10])
%     ylim([0 1])
%     hold on

```

```

% end;

% q=q+1;

clear mypi

clear myCam

% --- Executes on button press in pushbutton7.

function pushbutton7_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%%%%%%Evaluacion de ejercicios realizados correctos o incorrectos

global may;

global guardardat;

global t;

color=['y' 'm' 'c' 'r' 'b' 'g'];

aux=0;

cont=1;

colo=1;

t=t-1;

for m=1:t

    if guardardat(m)~=10

        y(cont)=guardardat(m)

axes(handles.axes12),plot(cont,guardardat(m),'*','Color',color(colo));

        xlim([0 10])

        ylim([0 1])

        hold on

        cont=cont+1;

    end

    if guardardat(m)==10

        cont=1;

        colo=colo+1

```

```

        end

    end

t=t+1;

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)

axes(handles.axes8)
imshow('sentadilla.jpg')

axes(handles.axes2)
imshow('flexionpie.jpg')

axes(handles.axes3)
imshow('aduccion.jpg')

axes(handles.axes4)
imshow('hombro.jpg')

axes(handles.axes5)
imshow('flexion.gif')

axes(handles.axes6)
imshow('brazos.png')

% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

## ANEXO 9

### Datasheet Raspberry Pi 2



Raspberry Pi



### Raspberry Pi 2, Model B

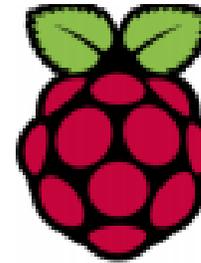
<b>Product Name</b>	Raspberry Pi 2, Model B
<b>Product Description</b>	The Raspberry Pi 2 delivers 5 times the processing capacity of previous models. This second generation Raspberry Pi has an upgraded Broadcom BCM2836 processor, which is a powerful ARM Cortex-A7 based quad-core processor that runs at 900MHz. The board also features an increase in memory capacity to 1Gbyte.

<b>Specifications</b>	
<b>Chip</b>	Broadcom BCM2836 SoC
<b>Core architecture</b>	Quad-core ARM Cortex-A7
<b>CPU</b>	900 MHz
<b>GPU</b>	Dual Core VideoCore IVB Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1Gpixel/s, 1.5Gpixel/s or 3MFLOPs with texture filtering and DMA infrastructure
<b>Memory</b>	1GB LPDDR2
<b>Operating System</b>	Boots from Micro SD card, running a version of the Linux operating system
<b>Dimensions</b>	85 x 56 x 17mm
<b>Power</b>	Micro USB socket 5V, 2A

<b>Connectors:</b>	
<b>Ethernet</b>	10/100 BaseT Ethernet socket
<b>Video Output</b>	HDMI (rev 1.3 & 1.4)
<b>Audio Output</b>	3.5mm jack, HDMI
<b>USB</b>	4 x USB 2.0 Connector
<b>GPIO Connector</b>	40-pin 2.54 mm (100 mil) expansion header: 2x20 ship Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
<b>Camera Connector</b>	15-pin MIPI Camera Serial Interface (CSI-2)
<b>JTAG</b>	Not populated
<b>Display Connector</b>	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
<b>Memory Card Slot</b>	Micro SDIO

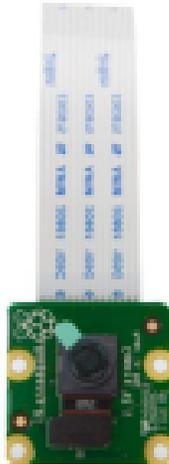
## ANEXO 10

### Datasheet Raspberry Pi Camera v2.0



#### Raspberry Pi Camera v2

Part number: RPI SMP CAMERA BOARD



- 8 megapixel camera capable of taking photographs of 3280 x 2464 pixels
- Capture video at 1080p30, 720p60 and 640x480p90 resolutions
- All software is supported within the latest version of Raspbian Operating System

The Camera v2 is the new official camera board released by the Raspberry Pi foundation.

The Raspberry Pi Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSI interface, designed especially for interfacing to cameras.

- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable.

The high quality Sony IMX219 image sensor itself has a native resolution of 8 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

#### Applications

- CCTV security camera
- motion detection
- time lapse photography

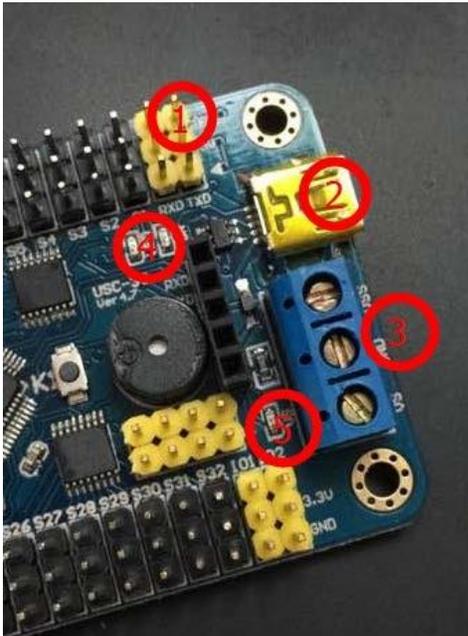
## ANEXO 11

### Datashet de Controlador de Servo Motor Torobot

Servo Controller Instructions for Use

Power Supply

The servo controller needs two power supplies: servo power supply and chip power supply.



Servo power supply (+): VS (left of the blue connecting terminal at Position 3 in the figure) Servo power supply (-): GND (middle of the blue connecting terminal at Position 3 in the figure)

Servo power supply's parameters depend on the parameters of the attached servo. For example, if the TR213 servo has a power supply of 4.8-7.2V, the servo power supply can use the power source of 4.8-7.2V.

Chip power supply (+): VSS (right of the blue connecting terminal at Position 3 in the figure) Chip power supply (-): GND (middle of the blue connecting terminal at Position 3 in the figure)

There is a VSS requirement of 6.5-12V. If the chip power is input through the VSS port, the power supply has to

range from 6.5 to 12V.

**Notes:**

The USB port at Position 2 in the figure can supply power to the chip. So it is adequate to choose the USB port or alternatively the VSS port.

Position 1 in the figure can supply power to the chip as well, marked 5V and GND, where 5V is the anode and GND is the cathode. The power supply has to be 5V.

Positions 1, 2 and 3 can supply power to the chip. It is adequate to choose any of them.

The green LED light at Position 4 in the figure is the chip power indicator. If the green light is on, it indicates the chip power works correctly; if the light is off, it indicates the chip power malfunctions.

The green LED light at Position 5 in the figure is the servo power indicator. If the green light is on, it indicates the servo power works correctly; if the light is off, it indicates the servo power malfunctions.

**It is necessary for both green LED lights to be on to control the servo.**

Install the Driver

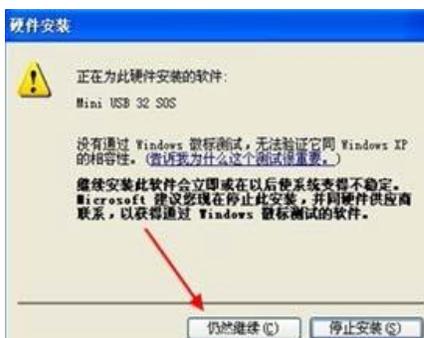
The driver is available at [http://dl.torobot.com/download/usc\\_driver.exe](http://dl.torobot.com/download/usc_driver.exe) (case-sensitive)

Directly double click on USC\_driver.exe; click on Next and the driver will be installed automatically.

If the prompt below occurs during installation, please choose “always install this driver”.



If the prompt below occurs during installation, please choose “continue”.

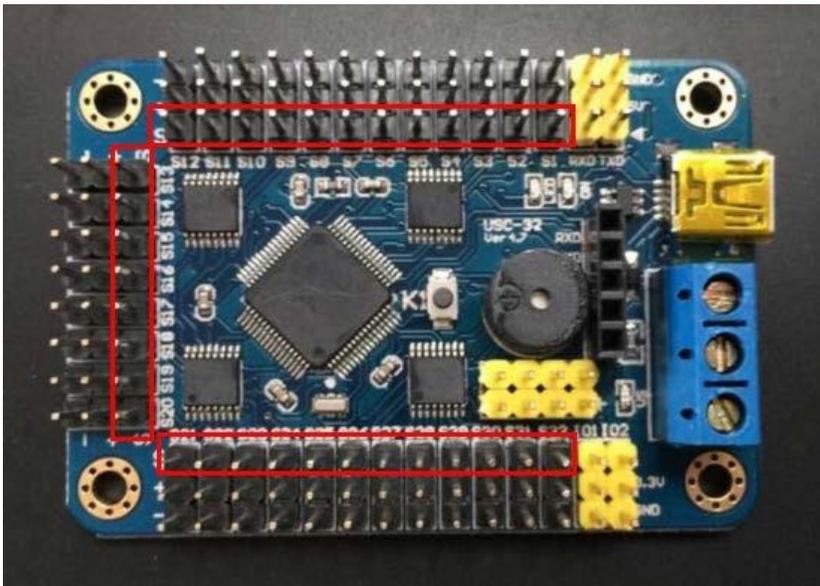


After the driver is installed successfully, enter the computer’s device manager and you will see the hardware device of the servo controller. For example, the mini USB servo control in the figure

below is the device name, and the COM1 is the port number. The device's port number is needed when the computer software control is exercised on the servo.



Connect the servo to the servo controller



What is marked red in the figure are the servo's connectors for signal wires (be careful about the direction when

connecting to the servo).

What is marked yellow in the figure are not the servo's connectors.

Pay attention to the white textual symbols aside when connecting to the servo. For example, S1, S2,..., S32 refers to the servo channels that correspond to the computer software.

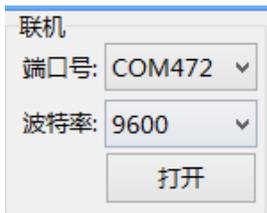
Download the Software

The software can be downloaded at

[http://dl.torobot.com/download/rios\\_usc.exe](http://dl.torobot.com/download/rios_usc.exe) (case-sensitive).

Control one single servo

Run RIOS\_USC.exe, choose the right port number, and then click the button “open”.



Use the mouse to drag the slider in the servo panel (drag the servo panel corresponding to the channel with which the servo is connected; at the upper part of the panel is the serial number, such as the S1 in the figure below).



### Control Multiple Servos Simultaneously

After multiple servos are controlled in sequence by following the steps above, set the time (e.g. in the figure below, the setting, referred to as the rotary speed, is 1000ms; it has to fall in the range 100-9999; the higher the value, the slower the speed). Then click on the button “add” at the lower part of the software. The software will produce a command at the lower part of the software which can exercise simultaneous control over all the servos that are controlled earlier (if 10 servos are controlled earlier, the command can control these 10 servos simultaneously).



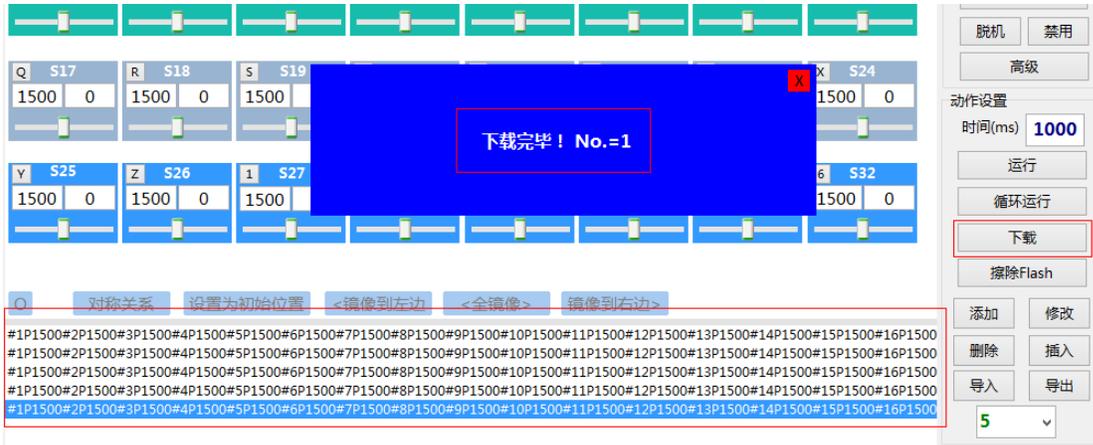
### Download the Action Group

If several or dozens of commands are produced by following the steps above, you can click on the button “run” at the right of the software to test these commands.

If the test result is acceptable, you can click on the button “download” at the right of the software to download the action group.

On completion of the download, the software will prompt “download is complete! No.=1”, where the number refers to the serial number of this action group.

Afterwards, all commands in the group can be executed by executing the action group.



Run the Action Group

First click on the button “read” to get all serial numbers of groups, then input the number of times of executions,

click the button “run”, and the selected action group will be executed.



Use the Off-Line Working

First click on the button “read” to get all serial numbers of groups, input the number of times of executions, click the button “off-line”, and then the selected action group will be executed off-line (off-line means that the group of actions will not be executed until the controller power is turned on).

脱机工作

动作组      次数

1      5

读取

执行

脱机      禁用

高级

If off-line working of the controller is no longer needed, you can click the button “disable” to turn off the function.

#### Erase Flash

Erase all action group already downloaded to the controller.

Name	Command	Description
Control one single servo	#1P1500T100\r\n	Data 1 refers to the servo's channel Data 1500 refers to the servo' location, in the range 500-2500 Data 100 refers to the time of execution and represents the speed, in the range 100-9999
Control multiple servos	#1P600#2P900#8P2500T100\r\n	Data 1, 2, and 8 refer to the servo's channels Data 600, 900, and 2500 refer to the locations of the servos that correspond to three channels Data 100 refers to the time of execution and represents the speed of three servos. Regardless of the number of servos, there is only one time, or one T. The command is executed at the same time; that is, all servos operate simultaneously.
Execute one single action group	#1GC2\r\n	Data 1 refers to the serial number of the action group Data 2 refers to the number of cycles
		Execute the first, third and first action group,The number of cycles is 2. One particular group of action

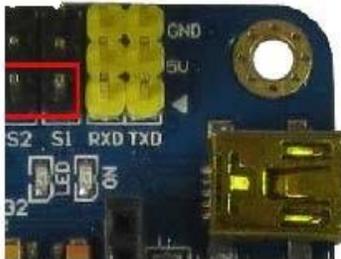
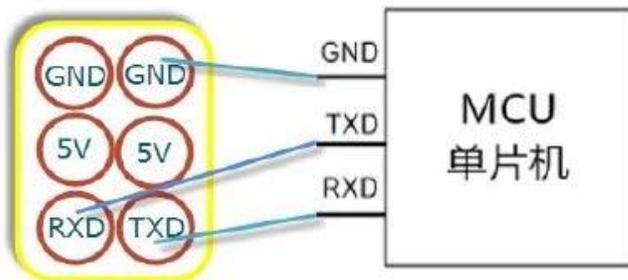
Execute multiple action groups	#1G#3G#1GC2\r\n	can appear repeatedly. There can be only one number of cycles or C. The command is executed in sequence; that is, the action groups are executed in sequence.
--------------------------------	-----------------	---

All commands above contain \r\n. It is the end mark of the command and is mandatory. All commands are no spaces.

\r\n represents two characters of carrier return and linefeed, and are the hexadecimal 0x0D and 0x0A, or Chr(13)

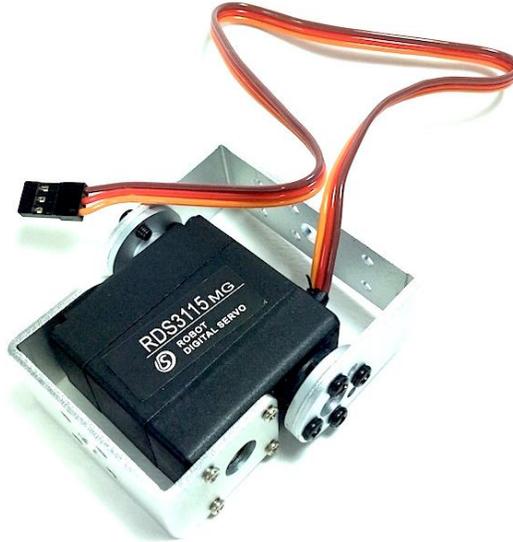
and Chr(10).

Connect to the MCU



## ANEXO 12

### Servo Motor RDS 3115MG



#### General Specification

Storage temperature range: -20°C ~ 60°C

Operating temperature range: -10°C ~ 50°C

Operating voltage: 5V ~ 7.2V

#### Electrical Specification

Description	5V	7.2V
Operating speed (at no load)	0.16sec/ 60°	0.14sec/ 60°
Running current (at no load)	80mA	100mA
Stall torque (at lock)	13.5kg.cm	15kg.cm
Stall current (at lock)	1.3 <sup>a</sup>	1.5A
Idle current (at stopped)	4mA	5mA

#### Mechanical Specification

Description	Specification
Overall dimension	40 x 20 x 40.5mm
Limit Angle	360°±10°
Weight	64±1g
Connector wire gauge	#28 PVC
Connector wire length	320±5mm
Horn gear spline	25T/ψ5.80
Reduction ratio/td>	310:1

**Control Specification:**

Description	Specification
Operating frequency	50-330Hz
Operating angle	90° (from 1000 to 2000 usec)
Neutral position	1500 usec
Dead band with	3 usec
Rotating direction	Counter clockwise (from 1000 to 2000 usec)
Pulse width range	From 500 to 2500 usec

**Package includes:**

1x Servo Motor RDS3115

Brackets + screws