



**IMPLEMENTACIÓN DE UN SISTEMA DE TELECONTROL DE UN ROBOT LEGO
MINDSTORMS EV3 MEDIANTE EL SOFTWARE LIBRE DE LEGO EV3 PARA SU
APLICACIÓN EN COMPUTADORES Y DISPOSITIVOS ANDROID**

PAREDES CHILUISA ALEXANDER MIGUEL

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

**MONOGRAFÍA, PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA**

ING. CHUCHICO ARCOS, CRISTIAN PAÚL.

25 DE AGOSTO DEL 2020



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

CERTIFICACIÓN

Certifico que el trabajo de monografía, **“IMPLEMENTACIÓN DE UN SISTEMA DE TELECONTROL DE UN ROBOT LEGO MINDSTORMS EV3 MEDIANTE EL SOFTWARE LIBRE DE LEGO EV3 PARA SU APLICACIÓN EN COMPUTADORES Y DISPOSITIVOS ANDROID”**, fue realizado por el señor **PAREDES CHILUISA, ALEXANDER MIGUEL**, el mismo que ha sido revisado y analizada en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, Agosto del 2020

Ing. Chuchico Arcos, Cristian Paúl

DIRECTOR DEL TRABAJO DE GRADUACIÓN

C. C.: 0503062713

URKUND



Document Information

Analyzed document	Paredes Alexander Monografia.docx (D77884621)
Submitted	8/18/2020 5:42:00 AM
Submitted by	
Submitter email	amparedes6@espe.edu.ec
Similarity	7%
Analysis address	cpchuchico.espe@analysis.arkund.com

Sources included in the report

W	URL: https://riuma.uma.es/xmlui/bitstream/handle/10630/12980/Alberto%20Martin%20Domingu ... Fetched: 5/30/2020 5:37:16 PM	4
SA	J.C. R. TesisCompleta (2).pdf Document J.C. R. TesisCompleta (2).pdf (D11498227)	12
SA	Universidad de las Fuerzas Armadas ESPE / plagio tesis Cbop German.docx Document plagio tesis Cbop German.docx (D25916479) Submitted by: chris4324@hotmail.com Receiver: jjespinosa.espe@analysis.arkund.com	18
SA	Universidad de las Fuerzas Armadas ESPE / Trabajo#8_2423_GRUPO4_ManualdeusuarioGripper.pdf Document Trabajo#8_2423_GRUPO4_ManualdeusuarioGripper.pdf (D47427815) Submitted by: nejimenez@espe.edu.ec Receiver: phmejia.espe@analysis.arkund.com	6
SA	Universidad de las Fuerzas Armadas ESPE / TRABAJO#7_2423_GRUPO4_MANUALDEUSUARIOLEGO.pdf Document TRABAJO#7_2423_GRUPO4_MANUALDEUSUARIOLEGO.pdf (D46887051) Submitted by: nejimenez@espe.edu.ec Receiver: phmejia.espe@analysis.arkund.com	2

Ing. Chuchico Arcos, Cristian Paúl

DIRECTOR



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

RESPONSABILIDAD DE AUTORÍA

Yo, **Paredes Chiluisa, Alexander Miguel**, con cedula de ciudadanía N° **0550257166**, declaro que el contenido, ideas y criterios de la monografía: **“IMPLEMENTACIÓN DE UN SISTEMA DE TELECONTROL DE UN ROBOT LEGO MINDSTORMS EV3 MEDIANTE EL SOFTWARE LIBRE DE LEGO EV3 PARA SU APLICACIÓN EN COMPUTADORES Y DISPOSITIVOS ANDROID”**, es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, Agosto del 2020

Firma

Paredes Chiluisa, Alexander Miguel

C.C.: 0550257166



DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA
CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

AUTORIZACIÓN DE PUBLICACIÓN

Yo **PAREDES CHILUISA ALEXANDER MIGUEL**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar la monografía: **“IMPLEMENTACIÓN DE UN SISTEMA DE TELECONTROL DE UN ROBOT LEGO MINDSTORMS EV3 MEDIANTE EL SOFTWARE LIBRE DE LEGO EV3 PARA SU APLICACIÓN EN COMPUTADORES Y DISPOSITIVOS ANDROID”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, Agosto del 2020

Firma

Paredes Chiluisa, Alexander Miguel

C.C.: 0550257166

DEDICATORIA

Dedico este proyecto de titulación a mi madre (Alexandra)
que cada instante de mi trayectoria estudiantil estuvo
siempre pendiente de mí, apoyándome
moral y económicamente.

A mi enamorada y queridos hermanos
que siempre me apoyaron,
me brindaron su
confianza, amor y cariño.

AGRADECIMIENTO

A Dios primordialmente que me ha regalado
salud y vida para cumplir
cada uno de mis propósitos.
También agradezco infinitamente a mis tías/os
que al igual que mi madre estuvieron ahí
en mis momentos más difíciles
con su amor, pusieron
toda su confianza motivándome
a ser una persona de bien.
A mis hermanos que siempre estuvieron
apoyándome de todas las
formas posibles.
Quiero agradecer al Ingeniero Cristian Chuchico
quién con paciencia me ilustro y ayudo
para el desarrollo y culminación
de este proyecto.

ÍNDICE DE CONTENIDOS

CARÁTULA	1
CERTIFICACIÓN	2
U R K U N D	3
RESPONSABILIDAD DE AUTORÍA	4
AUTORIZACIÓN DE PUBLICACIÓN	5
DEDICATORIA	6
AGRADECIMIENTO	7
ÍNDICE DE CONTENIDOS	8
ÍNDICE DE TABLAS	11
ÍNDICE DE FIGURAS	12
RESUMEN	21
ABSTRACT	22
CAPÍTULO I	
PLANTEAMIENTO DEL PROBLEMA	23
1.1 Antecedentes	23
1.2 Planteamiento del problema	24
1.3 Justificación.....	25
1.4 Objetivos	26
1.4.1 Objetivo General.....	26
1.4.2 Objetivos Específicos	26
1.5 Alcance.....	27
CAPÍTULO II	
MARCO TEÓRICO	28
2.1 ¿Qué es la robótica?.....	28

2.1.1. Fundamento teórico robot Lego Ev3	28
2.1.2. Descripción general Lego Mindstorms	29
2.2 Tecnología EV3.....	31
2.2.1 Descripción General.....	31
2.2.2 Compatibilidad entre NXT y EV3	32
2.2.3 Software de EV3	34
2.3 Bloque EV3.....	38
2.3.1 Descripción General.....	38
2.3.2 Instalación de pilas y Encendido del bloque EV3.....	40
2.4 Resto de Herramientas del EV3 con sus características	42
2.4.1 Motor grande EV3	42
2.4.2 Motor mediano EV3.....	43
2.4.3 Sensor de Color.....	44
2.4.4 Sensor de Contacto.....	47
2.4.5 Sensor Infrarrojo	49
2.4.6 Transmisor De Infrarrojo Remoto (Baliza)	50
2.5 Conexiones del Bloque EV3	51
2.5.1 Conexión de Sensores y Motores	51
2.5.2 Conexión Bloque EV3 a su equipo	53
2.5.3 Conexión del Bloque EV3 a un dispositivo android.....	57

CAPÍTULO III

DESARROLLO DEL TEMA	60
----------------------------------	-----------

3.1 Introducción	60
3.2 Ensamblaje del chasis del robot móvil.....	60
3.3 Ensamblaje del Bloque EV3 y cableado de sensores y motores.....	75
3.4 Programación en el software LEGO EV3.	78
3.5 Telecontrol del Robot Lego Mindstorms EV3	111

CAPÍTULO IV

CONCLUSIONES	159
RECOMENDACIONES.....	160
REFERENCIAS BIBLIOGRÁFICAS.....	161
ANEXOS	163

ÍNDICE DE TABLAS

Tabla 1. Diferencias entre los ladrillos de LEGO.....	33
Tabla 2. Requisitos mínimos de instalación	34
Tabla 3. Opciones y recursos de la página de inicio	35
Tabla 4. Partes que constan el editor de programas.....	37
Tabla 5. Explicación de las Luces de estado del Bloque EV3.	39
Tabla 6. Especificaciones del motor grande.....	43
Tabla 7. Modos de programar el sensor infrarrojo	50
Tabla 8. Especificaciones del transmisor infrarrojo remoto.....	50
Tabla 9. Especificación de uso predeterminado de los puertos de entrada	52
Tabla 10. Especificación de uso predeterminado de los puertos de salida	53
Tabla 11. Pasos de conexión del bloque hacia la PC	53
Tabla 12. Tipos de estado del sensor Táctil	86
Tabla 13. Tipos de estado del sensor Infrarrojo	91
Tabla 14. Combinaciones de botones de la baliza infrarroja remota.....	113
Tabla 15. División de desarrollo de la APP	115
Tabla 16. División de apartados.....	116

ÍNDICE DE FIGURAS

Figura 1. Descripción gráfica de la robótica	28
Figura 2. Uno de los modelos de Robots.....	29
Figura 3. Panorama gráfico de la tecnología EV3	31
Figura 4. Ladrillo NXT	32
Figura 5. Entorno del Programa	36
Figura 6. Entorno del editor de programas.....	37
Figura 7. Bloque EV3 con especificaciones generales.....	38
Figura 8. Luces de estado del Bloque EV3	39
Figura 9. Colocación de batería recargable al Bloque EV3.....	40
Figura 10. Colocación de pilas recargables al Bloque EV3.....	41
Figura 11. Pantalla de encendido y apagado del bloque.....	42
Figura 12. Motor Grande.....	43
Figura 13. Motor Mediano.....	44
Figura 14. Sensor de color.....	44
Figura 15. Sensor en Modo color	45
Figura 16. Sensor en Modo Intensidad de Luz Refleja.....	46
Figura 17. Sensor Modo Intensidad de Luz Ambiental.	46
Figura 18. Sensor Táctil.....	47
Figura 19. Sensor Táctil Presionado	48
Figura 20. Sensor Táctil Lanzado.....	48

Figura 21. Sensor Táctil en Contacto	49
Figura 22. Sensor Infrarrojo.....	49
Figura 23. Transmisor de IR remoto.....	51
Figura 24. Conexión de Sensores.	51
Figura 25. Conexión de Motores	52
Figura 26. Bloque del panel conectado por USB.	54
Figura 27. Pestaña de bloque EV3 disponibles.	54
Figura 28. Muestra de la conexión de bluetooth	55
Figura 29. USB introducido en la ranura brick de LEGO	56
Figura 30. Pestaña de bloque EV3 disponibles	56
Figura 31. Muestra de la conexión WIFI.....	57
Figura 32. Pantalla muestra configuración de bluetooth.....	58
Figura 33. Icono de no hay dispositivos conectados.	58
Figura 34. Pantalla mostrando la conexión de forma manual	59
Figura 35. Comienzo del ensamblaje.....	61
Figura 36. Pasos y piezas para el acople de la rueda	61
Figura 37. Piezas y montaje de la rueda derecha delantera.	62
Figura 38. Motor mediano derecho acoplado.....	62
Figura 39. Chasis montaje con rueda delantera.	63
Figura 40. Rueda trasera montada al chasis	63
Figura 41. Acoplamiento del chasis de abajo del robot.	64

Figura 42. Montaje de las primeras piezas del soporte.	65
Figura 43. Piezas faltantes para el montaje del soporte del chasis central.	65
Figura 44. Montaje del soporte del motor grande al chasis general.	66
Figura 45. Colocación de la parte izquierda del motor grande con sus respectivas ruedas	67
Figura 46. Acople del motor mediano.....	67
Figura 47. Colocación del motor mediano con sus aspas angulares y acople al chasis.....	68
Figura 48. Colocación de dos aspas rectas atrás del motor mediano.....	69
Figura 49. Montaje de las piezas de la parte frontal del robot.....	69
Figura 50. Ensamblaje del sensor de color y luminosidad en el robot	70
Figura 51. Acoplamiento del tronco del sensor infrarrojo para sujetarlo	71
Figura 52. Sujeción del sensor infrarrojo con el tronco armado.	71
Figura 53. Montaje del sensor infrarrojo en el robot móvil.....	72
Figura 54. Acoplamiento primario del brazo del robot.	73
Figura 55. Acople del antebrazo.....	73
Figura 56. Ensamblaje de las pinzas del robot móvil.....	74
Figura 57. Acoplamiento de los brazos robóticos al EV3MEG.	74
Figura 58. Acople de aspas rectas en el bloque EV3.	75
Figura 59. Demostración de los sensores con sus respectivos cableados	76
Figura 60. Cable USB conectado del Bloque EV3 a la PC.....	77
Figura 61. Conexión del Puerto A al motor mediano	77

Figura 62. Cable USB conectado del Bloque EV3 a la PC.....	78
Figura 63. Entorno del software para el nuevo proyecto	79
Figura 64. Proyecto nuevo abierto con el bloque de flujo predeterminado.	79
Figura 65. Selección de la paleta de colores el sensor de color	80
Figura 66. Pasos a seguir para iniciar calibración	80
Figura 67. Colocación del bloque de pantalla en la programación.....	81
Figura 68. Configuración del bloque de pantalla en la programación	82
Figura 69. Configuración del bloque de espera.	82
Figura 70. Configuración del conjunto ID y Estado del bloque de espera.....	83
Figura 71. Colocación de bloques de sensor para calibrar y medir la cantidad de luz reflejada	83
Figura 72. Copiar y pegar los 4 últimos bloques de programación	84
Figura 73. Configuración para visualizar y calibrar la luz reflejada en la superficie negra.....	84
Figura 74. Últimos pasos para programar la calibración del sensor de color	85
Figura 75. Panel de Hardware listo para enviar a descargar	85
Figura 76. Valores visualizados en el bloque físico.	86
Figura 77. Colocación del bucle en la programación.	87
Figura 78. Colocación y configuración del sensor táctil dentro del bucle.....	88
Figura 79. Colocación del bloque de pantalla dentro del bucle.	88
Figura 80. Configuración del bloque de pantalla.	89
Figura 81. Configuración para la visualizar los valores lógicos de medida	89

Figura 82. Vista del panel del Hardware para enviar la descarga.....	89
Figura 83. Comprobación de estados más utilizados por el sensor táctil.	90
Figura 84. Colocación del bucle en la programación.....	92
Figura 85. Colocación y configuración del sensor infrarrojo dentro del bucle....	93
Figura 86. Colocación del bloque de pantalla dentro del bucle	93
Figura 87. Configuración del bloque de pantalla.....	94
Figura 88. Configuración para visualizar los valores lógicos de modo medida (Proximidad).....	94
Figura 89. Vista del panel del Hardware para enviar la descarga.....	95
Figura 90. Comprobación de estado en modo proximidad por el sensor infrarrojo.....	95
Figura 91. Primeros pasos de programación de los motores grandes.....	96
Figura 92. Configuración del bloque de rotación del motor.	97
Figura 93. Colocación de un bucle en la programación.....	97
Figura 94. Configuración dentro de bucle del bloque de rotación del motor.	98
Figura 95. Colocación y configuración del bloque de matemática.	99
Figura 96. Conexión desde el bloque rotación del motor hacia el bloque matemático	99
Figura 97. Colocación y configuración del bloque motor grande.....	100
Figura 98. Añadir dos bloques más de rotación de motor	100
Figura 99. Configuración de los dos nuevos bloques.	100
Figura 100. Conexión del bloque de rotación hacia el bloque de matemáticas	101

Figura 101. Añadir el bloque matemático y configurar para multiplicar.....	101
Figura 102. Configuración del bloque verde Mover la Dirección	102
Figura 103. Configuración del bucle para rotación de los motores grandes.	103
Figura 104. Asignación de parámetros en el bucle.	104
Figura 105. Funcionamiento de secuencia óptima de rotación de los motores	104
Figura 106. Primeros pasos de Programación del motor mediano.	105
Figura 107. Configuración del bloque de rotación del motor	105
Figura 108. Colocación del bucle y bloque de rotación con su configuración..	106
Figura 109. Añadir y configurar el bloque matemáticas.....	107
Figura 110. Conexión desde el bloque de rotación del motor hacia el bloque de texto.....	107
Figura 111. Añadir y configurar el bloque de pantalla.....	108
Figura 112. Unión de salida con la entrada de bloques programados y añadir un nuevo bucle.	108
Figura 113. Unión del nuevo bucle al programa y añadir un bloque de motor mediano.....	109
Figura 114. Configuración de la potencia del Bloque motor mediano	109
Figura 115. Funcionamiento óptimo de la velocidad del motor mediano	110
Figura 116. Inicio de configuración del brick.....	111
Figura 117. Selección y Entorno del IR remoto.....	112
Figura 118. Baliza Infrarroja Remota.....	113
Figura 119. <i>Activar modo baliza y Colocar canal adecuado</i>	114

Figura 120. Control del robot móvil mediante la baliza infrarroja remota	115
Figura 121. Entorno de pantalla para la creación del nuevo proyecto.	117
Figura 122. Entorno y Configuración de pantalla.	117
Figura 123. Inicio de creación del diseño de la aplicación.....	118
Figura 124. Colocación de dos Etiquetados.....	119
Figura 125. <i>Plantilla de componentes para el control</i>	119
Figura 126. <i>Componentes no visibles</i>	122
Figura 127. <i>Explicación breve de los elementos no visibles</i>	122
Figura 128. Inicialización del Bloque.	124
Figura 129. Bloque de Adelante y Retroceso.....	125
Figura 130. Bloque de Izquierda, Derecha y Paro.	126
Figura 131. Bloque de uso del deslizador1.	126
Figura 132. Bloque de uso del deslizador2.	127
Figura 133. Generar Código APK.....	128
Figura 134. Instalación Setup-leJOS.....	129
Figura 135. Instalación del JDK.....	130
Figura 136. Selección de Destinatario y Componentes	130
Figura 137. Selección Carpeta de Inicio e Instalación.	131
Figura 138. Aviso de Setup y proceso de Instalación.	132
Figura 139. Creación del EV3 SD CARD.	133
Figura 140. Selección de JRE en el directorio de descargas	134

Figura 141. Inserción de tarjeta SD y encendido del bloque	135
Figura 142. Secuencia de inicio de leJOS	135
Figura 143. <i>Página Principal de Download de Eclipse</i>	136
Figura 144. Pantalla Generada para descargar	137
Figura 145. Selección del Complemento del Eclipse	137
Figura 146. Instalación del nuevo software de leJOS.....	138
Figura 147. <i>Configuración para añadir el nuevo software.</i>	139
Figura 148. Visualización del nuevo software.	140
Figura 149. Configuración del EV3_HOME.....	140
Figura 150. Inicio de Creación de Nuevo Proyecto.....	141
Figura 151. <i>Asignar Nombre del Proyecto y uso de JRE</i>	142
Figura 152. Vista de Contenido del Paquete Explorador.....	143
Figura 153. Creación de una nueva carpeta en src.....	143
Figura 154. Creación de una Clase Nueva de Java.	144
Figura 155. Entorno de vista de un Proyecto Básico en Eclipse.	145
Figura 156. Creación de 2 New Java Class en Eclipse.	146
Figura 157. Inicio de Programación del "Servidor".	147
Figura 158. Líneas de programación para cancelar el programa.....	147
Figura 159. Líneas de programación con main.....	148
Figura 160. Líneas de programación para ejecución de datos "Servidor".	149

Figura 161. Programación para ejecución de datos que controlan a los motores del robot.....	149
Figura 162. Programación para ejecución de datos que controlan a los motores del robot.....	150
Figura 163. Verificador de problemas y advertencias.....	150
Figura 164. Inicio de programación con definición de variables.....	151
Figura 165. Uso de la extensión para la acción del Key.....	152
Figura 166. Uso de private Socket para la comunicación entre máquinas.	152
Figura 167. Uso del String para la dirección ip de vinculación.	153
Figura 168. Definición del main con la dirección ip.	153
Figura 169. Creación de instancia del objeto Socket.....	154
Figura 170. Método realizado para enviar Comando.....	155
Figura 171. Actualización del estado en la interfaz.....	155
Figura 172. Actualización del estado en la interfaz.....	156
Figura 173. Métodos para la interpretación del teclado.....	156
Figura 174. Explicación para el Envío del programa “Servidor”.	157
Figura 175. Explicación para el Envío del programa “Cliente”.	158

RESUMEN

El objetivo de este proyecto es llevar a cabo el telecontrol del Robot LEGO Mindstorms EV3, utilizando el software libre para LEGO MINDSTORMS del equipo de piezas en su versión doméstica (#31313) de la misma marca, además de implementar la guía de uso para la manipulación y programación de sus sensores. Esta tecnología (LEGO) permite optimizar recursos y a la vez facilita el manejo del desenvolvimiento mental de los estudiantes donde pueden crear su propio robot con sus elementos motores y sensores inteligentes que se incluyan en el set. El proceso empieza con la programación del Bloque EV3 en el software LEGO MINDSTORMS EV3 Home Edition, basado en el lenguaje de programación LabVIEW, esto facilita el manejo de los bloques de configuración presentes en el software, también existe una interfaz de comunicación inalámbrica para controlar el robot ya sea vía wifi o bluetooth. Para lograr el telecontrol se empleará un entorno de desarrollo de software (App Inventor) y un API (Interfaces de Programación de Aplicaciones), para la elaboración de una aplicación destinada al control del robot por medio de una HMI donde el usuario pueda manipular los motores y sensores que posee el equipo de LEGO en esta versión. El telecontrol ejecutará las instrucciones necesarias para manipular los puertos de entradas y salidas que se accionan para el funcionamiento del robot móvil.

PALABRAS CLAVE:

- ✓ **ROBOT LEGO**
- ✓ **TELECONTROL DEL ROBOT**
- ✓ **SENSORES (DE COLOR, DE CONTACTO, INFRARROJO)**
- ✓ **ELEMENTOS MOTORES**
- ✓ **CONEXIÓN INALÁMBRICA**

ABSTRACT

The objective of this project is to carry out the remote control of the LEGO Mindstorm EV3 Robot, using the free software LEGO MINDSTORM of the parts kit in its domestic version (#31313) of the same brand as well as implementing the user guide for the manipulation and programming of its sensors. This technology (LEGO) allows to optimize resources and at the same time facilitates the handling of the mental development of the students where they can create their own robot with its motor elements and intelligent sensors that are included in the set. The process begins with the programming of the EV3 Block in the software LEGO MINDSTORMS EV3 Home Edition, based on the programming language LabVIEW, this facilitates the handling of the configuration blocks present in the software, there is also a wireless communication interface to remote control the robot either via wifi or bluetooth. To achieve the telecontrol, a software development environment (App Inventor) and an API (Application Programming Interfaces) will be used to perform an application to control the robot through an HMI where the user can manipulate the motors and sensors that the LEGO kit has in this version. The telecontrol will execute the necessary instructions to manipulate the input and output ports that are activated for the operation of the mobile robot.

KEYWORDS:

- ✓ **LEGO ROBOT**
- ✓ **ROBOT TELECONTROL**
- ✓ **SENSORS (COLOR, CONTACT, INFRARED)**
- ✓ **MOTOR ELEMENTS**
- ✓ **WIRELESS CONNECTION**

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

LEGO lleva desarrollando plataformas para que los niños y adultos aprendan robótica desde el siglo pasado. Es por tal motivo que algunos investigadores han realizado la mayoría de sus proyectos de titulación en base a las diferentes aplicaciones que se pueden realizar con las nuevas tecnologías de LEGO.

En el proyecto de titulación del señor Á. Vásquez de la Universidad Politécnica de Madrid, a través de su Proyecto final de la carrera: “APLICACIÓN CLIENTE SERVIDOR PARA EL CONTROL DE LEGO MINDSTORMS EV3”, en donde especifica las diferentes alternativas de desarrollo inicial del robot enuncia lo siguiente. El robot se controla remotamente desde un dispositivo móvil, desde el cual se configuran las reglas y se utilizan como control remoto (Presilla, 2017).

En el trabajo E. Gallegos y E. Gallegos Mayorga de la Universidad Católica Santiago de Guayaquil previa a la obtención del título de ingenieros en telecomunicaciones en base a su tesis: “EVALUACIÓN DE LA PLATAFORMA DIDÁCTICA LEGO MINDSTORMS EV3 PARA EL CONCURSO ECUATORIANO DE ROBÓTICA EN LA CATEGORÍA CREATIVIDAD LEGO”, especifican como realizar el diseño y programación de un Robot LEGO EV3 (Gallegos & Mayorga, 2015).

Mediante el trabajo escrito de investigación de D. Zaldívar, E. Valdemar, M. Pérez, S. Esquivel y D. Oliva conjuntamente con la editorial RA-MA basados en la plataforma de LEGO añaden las diferentes posibilidades de programación de robots LEGO donde argumentan lo siguiente:

“No se trata de simples cubos con los que podemos hacer obras de arte, sino que con sus productos podemos también aprender a crear y programar robots, o al menos en su versión básica” (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018).

Mediante el trabajo de titulación del señor S. Solórzano de la Universidad de las Fuerzas Armadas “ESPE”, a través de su proyecto de fin de carrera: “Implementación de un robot móvil con el kit LEGO MINDSTORMS para la adquisición de señales y visualización en LabVIEW”, en donde especifica cómo adquirir señales por medio de un toolkit llamado NXT robotic este puede enviar los datos al modelo de su proyecto y visualizarlos en LabVIEW (Solórzano, 2012).

1.2 Planteamiento del problema

Con el avance de la tecnología se hace más común el uso de la robótica educativa en los colegios, universidades, centros de capacitación entre otros. Las actividades centradas en el uso de la robótica están basadas en la resolución de problemas cotidianos, por lo que diferentes fabricantes ofertan robots tanto para el área industrial como la educativa.

La mayoría de productos de LEGO están orientados al área de investigación y educación. Cuentan con material de construcción eficaz para potenciar habilidades y conocimientos tecnológicos en el área de la robótica, programación y automática, que son base en la formación de profesionales en electrónica.

Es importante recalcar que el laboratorio de Instrumentación Virtual de la Sección Gestión de Tecnologías de la Universidad de las Fuerzas Armadas ESPE, no cuenta con un prototipo de esta versión y así tener la oportunidad de manipular un prototipo dentro de su formación, siendo importante adquirir un equipo LEGO MINDSTORMS EV3 versión doméstica #31313 planteándose el siguiente tema de

proyecto de titulación: Implementación De Un Sistema De Telecontrol De Un Robot LEGO MINDSTORMS EV3 Mediante El Software Libre De Lego Ev3 Para Su Aplicación En Computadores Y Dispositivos Android.

1.3 Justificación

El desarrollo del presente proyecto servirá como material didáctico para realizar futuras prácticas en el Laboratorio de Electrónica en lo referente a la manipulación y control de un robot, específicamente del LEGO MINDSTORMS EV3 en base al kit de piezas de robótica educativa.

La elaboración de una guía para los estudiantes del laboratorio de Instrumentación Virtual de la Sección Gestión de Tecnologías-UGT de la Universidad de las Fuerzas Armadas ESPE, facilita el uso del equipo considerando los pasos a seguir para poder telecontrolar y programar el robot LEGO MINDSTORMS EV3; por medio de un entorno de desarrollo de software desde un dispositivo Android o Computador que proporcionan acceso inalámbrico y una interfaz de usuario.

El software EV3 permite realizar una interfaz humano máquina que funciona con los sistemas operativos de WINDOWS y MAC, es decir, cuenta con todos los implementos necesarios para la programación y ajustes del robot LEGO MINDSTORMS EV3.

Además el control del robot se puede realizar en base a la conexión de Wifi donde su estándar habitual utilizado en los hogares es el 802.11n, ofreciendo un rango significativo de cientos de mega bits por segundo y Bluetooth 802.15.1 que servirá para la transmisión de datos de corto alcance donde el protocolo de Bluetooth es importante para desarrollar aplicaciones permitiendo conectarse directamente con el robot para el control del EV3.

1.4 Objetivos

1.4.1 Objetivo General

Implementar un sistema de telecontrol de un robot LEGO MINDSTORMS EV3 mediante el software libre de LEGO EV3 para su aplicación en computadores y dispositivos android.

1.4.2 Objetivos Específicos

- Investigar el funcionamiento y aplicaciones del robot LEGO MINDSTORMS EV3, por medio de la recopilación de información de tesis y proyectos hechos en la web para implementar en el sistema de telecontrol mediante dispositivos android y computadores.
- Realizar la programación y ajustes del robot utilizando la interfaz de programación instintiva de iconos del software libre LEGO MINDSTORMS EDUCATION EV3 para el test de los sensores y motores.
- Ejecutar pruebas de telecontrol utilizando el entorno de desarrollo de software de App Inventor para la creación de una APP que permite manipular los motores y sensores del LEGO MINDSTORM EV3.
- Efectuar pruebas de telecontrol utilizando una Interfaz de programación de aplicación (API) con .NET, mediante el acceso al control de motores y lectura de datos en tiempo real de los sensores para conectarse y controlar vía Wifi el LEGO MINDSTORM EV3.

1.5 Alcance

El proyecto técnico tiene la finalidad de aportar con información detallada de la implementación de un sistema de telecontrol para un robot Lego Mindstorms EV3 en base a software libre. Se presentarán diferentes aplicaciones lo que incluye el ensamblando distintos modelos de robots. Los sensores que se utilizaran son de tacto, distancia o color. Además de varios tipos de motores, programados en el EV3, que es el bloque de control.

El telecontrol contendrá además distintos mandos de programación e indicación hacia los montajes de modelos de robots que se puede realizar en base al kit de piezas de robótica educativa, las cuales vienen en versión doméstica #31313 y versión educativa #45544, para dejar constancia de la mayoría de pruebas de control de sus sensores y motores.

CAPÍTULO II

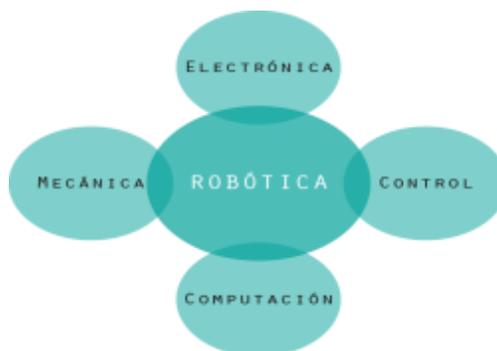
MARCO TEÓRICO

2.1 ¿Qué es la robótica?

La robótica es una rama interdisciplinaria de la ingeniería, que comprende: mecánica, electrónica, eléctrica, teoría del control y de las ciencias de la computación, como se describe en la figura 1. Estudia el análisis, diseño, manufactura y aplicación de máquinas automáticas con cierto grado de inteligencia, capaces de realizar tareas que pueden reemplazar las actividades de un ser humano (Heras, 2017).

Figura 1.

Descripción gráfica de la robótica.



Nota: (Heras, 2017).

2.1.1. Fundamento teórico robot Lego Ev3

Este proyecto está basado en el uso del robot Lego Mindstorms para implementar el telecontrol y diseñarlo de diferentes maneras haciendo uso de sus sensores como el infrarrojo, de color y de tacto.

“LEGO MINDSTORMS EV3 es un set de robótica que tiene varias herramientas para crear y dar órdenes a miles de robots LEGO. Además MINDSTORMS tiene para

construir cinco personajes robóticos básicos que se encuentran en el software de programación de EV3” (Pérez, 2015).

Además, el Bloque EV3 que contiene el robot es inteligente y programable que controla motores, sensores; además proporciona comunicación inalámbrica (bluetooth 802.15.1 y wifi 802.3). Elegir los motores y sensores a utilizar y construir robots tal como quiera que sea, a continuación se muestra la figura de un robot construido (LEGO®MINDSTORMS®, 2013).

Figura 2.

Uno de los modelos de Robots.



Nota: (Boogaarts, 2013)

2.1.2. Descripción general Lego Mindstorms

En 1998 LEGO lanzó al mercado la primera generación de bloques inteligentes, con el nombre LEGO Mindstorms RCX (Robotics Command eXplorer). Se programaba en un lenguaje gráfico muy intuitivo, y posteriormente se añadieron muchos más lenguajes. El bloque RCX posee 3 puertos para sensores y 3 puertos para actuadores, cuyas comunicaciones, en un principio, se realizaban mediante un puerto serie y, posteriormente, por infrarrojos con un periférico USB. En cuanto a las especificaciones

hardware, poseía un microcontrolador Renesas de 8-bit H8/300, una ROM de 16 Kb y una RAM de 32 Kb.

La segunda generación de Mindstorms y la más popular, lanzada en 2006, fue NXT; esta generación se compone de dos versiones: la educativa y la 2.0. La comunicación con el bloque es mediante cable USB y Bluetooth que, además, permite la comunicación con otros bloques y dispositivos móviles. Con respecto al hardware del controlador principal, posee un microcontrolador ATMEL ARM7 de 32 bits a 48 MHz, una memoria flash de 256 KB y una RAM de 64 KB; esto supone una sustancial mejora del hardware con respecto a su predecesor. En cuanto al controlador para los motores, existe un microcontrolador AVR de 8-bit, una memoria flash de 4 Kb y una RAM de 512 bytes. Con respecto a puertos, este modelo presenta tres puertos para actuadores y cuatro puertos para sensores. NXT goza de una gran comunidad. Además, dispone de un gran surtido de sensores oficiales y no oficiales, como cámaras, acelerómetros y brújulas, además de los sensores de luz, ultrasonido, táctil, etc., que contienen los packs convencionales.

La generación actual EV3, lanzada en 2013; su principal cambio se basa en incorporar los últimos avances tecnológicos en el bloque y proporcionar más precisión en sensores y actuadores. El bloque posee cuatro entradas para actuadores y cuatro entradas para sensores, un puerto USB para agregar un conector wifi o realizar conexiones en cadena, un puerto de tarjetas Micro SD para ampliar la memoria del EV3, un altavoz integrado, un receptor de señales infrarrojas y un receptor Bluetooth. Respecto al hardware, se destaca su microcontrolador ARM926EJ-S de 300 MHz, una RAM de 64 MB y 16 MB de memoria Flash. Hoy en día, la comunidad dedicada a la programación de la robótica de LEGO todavía está desarrollando soluciones para la programación del bloque, aun así ya hay algunos lenguajes, en los cuales ya se puede

programar: LabVIEW, leJOS, Monobrick, una toolbox de Matlab aplicada a EV3, APP Inventor aplicada al control del EV3 etc. Este último modelo es al que está enfocado el presente trabajo fin de grado (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018).

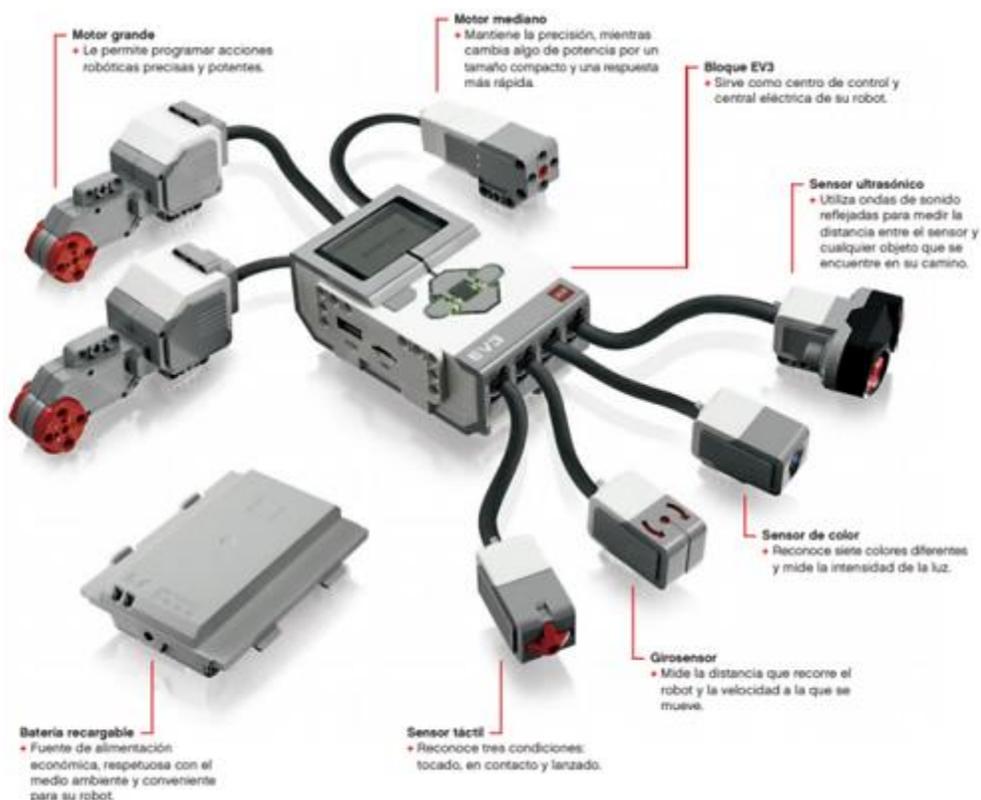
2.2 Tecnología EV3

La tecnología EV3 es la solución de robótica educativa para nivel desde secundaria en adelante, que motiva a los estudiantes a diseñar, construir y programar robots, otorgando un mejor entendimiento de cómo funciona la tecnología en problemas de la vida real (LEGO®MINDSTORMS®, 2013).

2.2.1 Descripción General

Figura 3

Panorama gráfico de la tecnología EV3



Nota: (LEGO®MINDSTORMS®, 2013)

2.2.2 Compatibilidad entre NXT y EV3

Los cables conectores seguirán siendo los mismos en ambas versiones. Eso posibilita el poder hacer combinaciones entre los elementos del NXT y del EV3. Pero con ciertas limitaciones. Si bien todos los sensores y motores oficiales de LEGO Mindstorms NXT son compatibles con el ladrillo EV3, y el software los reconoce como si de los nuevos se tratara, no ocurre lo mismo al contrario. Los sensores nuevos no son compatibles con el ladrillo NXT como se muestra en la figura 4, aunque sí lo son los motores (Nino, 2013).

Figura 4

Ladrillo NXT



Nota: (Nino, 2013)

El **LEGO EV3** es más potente comparado con el LEGO NXT, al ser un producto lanzado 7 años después, en 2013 y con los avances en la tecnología. El EV3 cuenta con mayores recursos de hardware, entre los cuales destaca la mayor capacidad de procesamiento, mayor capacidad de memoria flash, dos puertos USB y una notable velocidad en los puertos de sensores (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018).

A continuación, presenta una tabla con la diferencia entre esos dos ladrillos.

Tabla 1.

Diferencias entre los ladrillos de LEGO.

Característica	NXT	EV3
Procesador	ARM de 32 bits AT91SAM7S256 48MHz 256 KB FLASH RAM	ARM9 300MHz 16MB de Flash
Co-Procesador	Atmel 8-Bit AVR, ATmega48 8MHz 4KB FLASH-RAM RAM de 512 bytes	n/a
Sistema Operativo	Propiedad	Basado en linux
Puertos del sensor	4 puertos análogos Digital: 9600 bit / s (HC)	4 puertos análogos Digital, hasta 460,8Kbit/s (UART)
Puertos de motor	3 con encoders	4 con encoders
Comunicación USB	Velocidad máxima (12 Mbit/s)	Alta velocidad (480 Mbit/s)
Puerto USB	n/a	Daisy-chain (3 niveles) Wifi Dongle Almacenamiento USB
Tarjeta SD	n/a	Micro SD-Card Reader, puede manejar hasta 32GB
Comunicación con dispositivos inteligentes	Android	iOS Android
Interfaz de usuario	4 botones	6 botones con luz de fondo, útil para depurar y mostrar estado
Monitor	LCD Matriz, monocromo 100 x 64 píxeles	LCD Matriz, monocromo 178 x 128 píxeles
Comunicación	Bluetooth USB 2.0	Bluetooth v2.1 USB 2.0 (Para hablar con el PC) USB 1.1 (para encadenamiento).

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

2.2.3 Software de EV3

El software Lego EV3 se instala en un ordenador Windows o Mac y se utiliza para escribir programas, enviarlos al robot EV3 y ejecutarlos en él. Es una herramienta que facilita la introducción visual e intuitiva a la programación. Propone un catálogo de bloques que se van situando en una secuencia lineal a la vez que se configuran sus parámetros de comportamiento. Está orientado a la elaboración de proyectos EV3 dentro de los cuales es posible organizar distintos programas, incorporar recursos multimedia (imágenes, audios, vídeos, y otros) a través del editor de contenidos, añadir comentarios, textos anexos, entre otros (Prieto, 2015).

2.2.3.1 Requisitos Mínimos de Instalación

Los requisitos mínimos del ordenador para instalar el software de Lego EV3 se muestran en la siguiente tabla.

Tabla 2.

Requisitos mínimos de instalación

- | |
|---|
| <ul style="list-style-type: none"> • Sistema operativo: Windows XP, Vista, Windows 7 y Windows 8 o bien Mac 10.6, 10.7 y 10.8 |
| <ul style="list-style-type: none"> • Software instalado: Silverlight 5.0 y Microsoft Dot Net 4.0 |
| <ul style="list-style-type: none"> • Hardware: 2 GB de RAM, procesador de 1,5 GHz, 2 GB de espacio libre en disco duro, resolución de pantalla de 1024x600 y puerto USB 2.0 |

Nota: El software de Lego EV3 NO funciona en tabletas y puede comportarse de forma ralentizada en ordenadores notebook que dispongan sólo de 1 GB de RAM (Prieto, 2015).

2.2.3.2 Instalación del Software

Una vez confirmado que su equipo cumple los requisitos mínimos del sistema, está listo para instalar el software. En la dirección <http://www.lego.com/es-es/mindstorms/downloads/download-software> se puede obtener la versión Home o

Retail del programa para Windows o para Mac. Cierre cualquier otro programa y, luego, haga doble clic en el archivo de instalación en la carpeta de la aplicación del Software de EV3. La instalación comenzará ahora (Prieto, 2015).

2.2.3.2.1 Entorno del Programa

Al iniciar el software de **Legó EV3** se mostrará la página de inicio (figura 5), desde la cual es posible acceder a la programación y a todos los recursos digitales necesarios (Prieto, 2015). En la Página de inicio encontrará las opciones y los recursos como muestra en la tabla siguiente:

Tabla 3.

Opciones y recursos de la página de inicio

<p>1. Pestaña Página de inicio: este botón siempre lo hace regresar a la Página de inicio.</p> <p>2. Descripción general de la actividad: aquí puede obtener acceso, organizar y acceder a vistas previas de los contenidos y comenzar con un proyecto.</p> <ul style="list-style-type: none"> + Instrucciones de construcción para los Modelos principales. + Recursos de Introducción como un breve video de introducción y la Guía de uso del Bloque EV3. + Administración de archivos, donde puede comenzar un nuevo proyecto o abrir uno existente. + Robot educador, cuenta con cuarenta y cuatro tutoriales paso a paso que explican cómo utilizar el Software y el Hardware de EV3. <p>3. Ver: este botón le permite regresar a la Descripción general de la actividad.</p> <p>4. Buscar: busque proyectos con contenidos específicos mediante diferentes opciones de filtro.</p> <p>5. www.LEGOeducation.com/MINDSTORMS: un vínculo al sitio web oficial de LEGO® MINDSTORMS® Education.</p>
--

Nota: (Prieto, 2015)

Figura 5

Entorno del Programa



Nota: (Prieto, 2015)

2.2.3.2.2 Editor de Programas

El editor de programas para EV3 figura 6, consta de las siguientes partes tal como se muestra en la siguiente tabla:

Tabla 4.*Partes que constan el editor de programas*

1. Área de diseño del programa. En este panel se sitúan los bloques para crear la secuencia de programación.
2. Paleta de programación. Contiene los bloques que se utilizarán para el diseño del programa organizado en categorías.
3. Panel de Hardware. Permite establecer la conexión con el bloque EV3, comprobar los sensores y motores conectados y en qué puertos, consultar la versión del firmware instalado en el bloque, actualizarlo, enviar el programa diseñado al bloque para su ejecución, etc.
4. Editor de contenidos. Es un cuaderno digital adicional que se utiliza para documentar el proyecto usando textos, audios, vídeos, etc.
5. Barra de herramientas de programación. Incluye las herramientas básicas para editar el programa: guardar, zoom, deshacer/rehacer, etc.

Nota: (Prieto, 2015)

Figura 6*Entorno del editor de programas*

Nota: (Prieto, 2015)

2.3 Bloque EV3

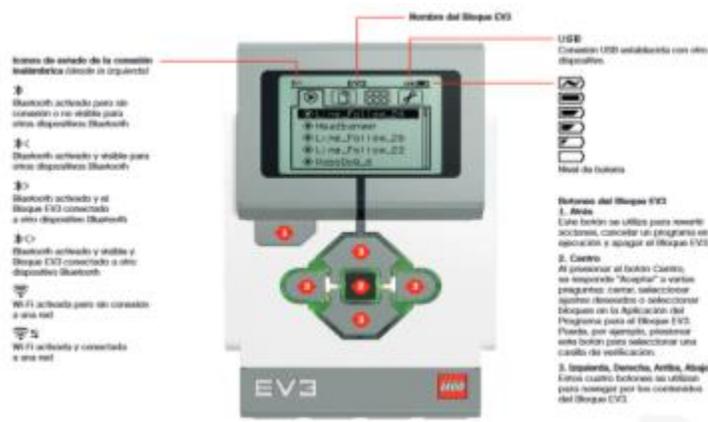
2.3.1 Descripción General

La Pantalla le muestra lo que sucede en el interior del Bloque EV3 y le permite utilizar la interfaz del Bloque EV3 (figura 7). También permite agregar texto y respuestas numéricas o gráficas a la programación o a los experimentos. Por ejemplo, tal vez sea conveniente programar la Pantalla para mostrar una cara feliz (o una cara triste) de una respuesta comparada o para mostrar un número que sea el resultado de un cálculo matemático “para obtener más información sobre el uso del Bloque Pantalla consulte la Ayuda del Software de EV3” (LEGO®MINDSTORMS®, 2013).

Además **los botones del Bloque EV3** le permiten explorar el interior de la Interfaz del Bloque EV3. También pueden utilizarse como activadores de programación. Por ejemplo, programar un robot para elevar los brazos si se presiona el botón Arriba o para bajarlos si se presiona el botón Abajo “para obtener más información, consultar el Uso de los botones del Bloque EV3 en la Ayuda del Software de EV3” (LEGO®MINDSTORMS®, 2013).

Figura 7

Bloque EV3 con especificaciones generales



Nota: (LEGO®MINDSTORMS®, 2013)

La Luz de estado del Bloque EV3 (figura 8) que rodea los botones del Bloque EV3 le indica el estado actual del Bloque EV3. Puede ser verde, naranja o roja y puede parpadear. Los códigos de la Luz de estado del Bloque EV3 se muestran en la siguiente tabla:

Tabla 5.

Explicación de las Luces de estado del Bloque EV3.

Rojo = Inicio, Actualizando, Apagado

Rojo con parpadeo = Ocupado

Naranja = Alerta, Listo

Naranja con parpadeo = Alerta, Ejecutando

Verde = Listo

Verde con parpadeo = Ejecutando programa

Nota: (LEGO®MINDSTORMS®, 2013)

Figura 8.

Luces de estado del Bloque EV3



Nota: (LEGO®MINDSTORMS®, 2013)

2.3.2 Instalación de pilas y Encendido del bloque EV3

2.3.2.1 Instalación de las baterías

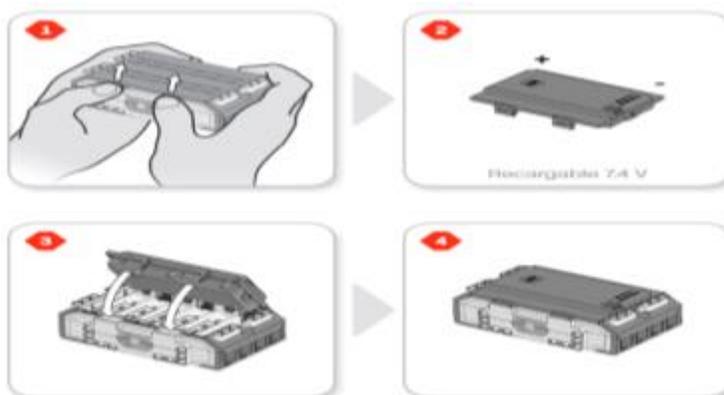
El bloque del LEGO EV3 puede utilizar baterías AA normales o el paquete de Baterías recargables de EV3 que se incluye en el Set principal EV3 LEGO MINDSTORMS Education.

La **Batería recargable de EV3** es una alternativa conveniente y económica para el uso de baterías AA. Puede recargarse mientras está incorporada en el modelo, lo que evita el trabajo de desarmar y volver a armar un robot para cambiar las baterías.

Para instalar la batería recargable en el Bloque EV3, quite la cubierta de la batería que se encuentra en la parte posterior del Bloque EV3 presionando las dos pestañas plásticas en el costado como se observa en la figura 9. Si hay baterías dentro del Bloque EV3, quítelas. Inserte la Batería recargable en las ranuras que sostienen la cubierta de la batería en su lugar y haga presión sobre la batería hasta que encaje en su lugar. La cubierta de la batería no se utiliza (LEGO®MINDSTORMS®, 2013).

Figura 9.

Colocación de batería recargable al Bloque EV3



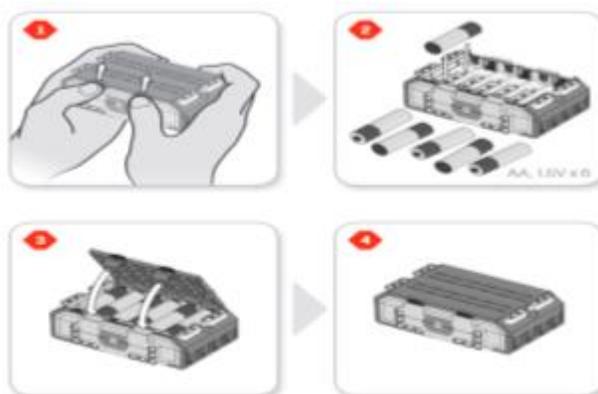
Nota: (LEGO®MINDSTORMS®, 2013)

Además el Bloque EV3 requiere seis baterías AA/LR6 si no se utiliza la Batería recargable de EV3. Se recomiendan las baterías AA de ion de litio alcalinas o recargables. La opción de las baterías AA es buena cuando se desea que el robot sea un poco más pesado.

Para instalar las baterías AA, quite la cubierta de la batería que se encuentra en la parte posterior del Bloque EV3 al presionar las dos pestañas plásticas en el costado. Después de insertar las seis baterías AA, vuelva a colocar nuevamente la cubierta de la batería como se muestra en la figura 10 (LEGO®MINDSTORMS®, 2013).

Figura 10.

Colocación de pilas recargables al Bloque EV3



Nota: (LEGO®MINDSTORMS®, 2013)

2.3.2.2 Encendido del Bloque EV3

Para encender el Bloque EV3, presione el botón Centro. Después de presionar el botón, la Luz de estado del Bloque EV3 se encenderá en color rojo y aparecerá la pantalla de inicio, como se muestra en la figura 11.

Cuando el color de la luz cambia a verde, el Bloque EV3 está listo. Para apagar el Bloque EV3, presione el botón Atrás hasta ver la pantalla de apagado.

La opción Cancelar X estará seleccionada. Utilice el botón Derecha para seleccionar la marca de verificación de Aceptar y, luego, presione el botón Centro para aceptar. El Bloque EV3 estará apagado. Si presiona Aceptar mientras se selecciona la X, volverá a la pantalla Ejecutar reciente (LEGO®MINDSTORMS®, 2013).

Figura 11.

Pantalla de encendido y apagado del bloque



Nota: (LEGO®MINDSTORMS®, 2013)

2.4 Resto de Herramientas del EV3 con sus características

2.4.1 Motor grande EV3

El motor grande es un motor "inteligente" potente. El motor grande (figura 12), se ha optimizado para ser la base motriz de sus robots. Al utilizar el bloque de programación Mover la dirección o Mover tanque, los motores grandes coordinarán la acción simultáneamente (LEGO®MINDSTORMS®, 2013).

Para que funcione a la velocidad exacta; sus especificaciones son las siguientes:

Tabla 6.*Especificaciones del motor grande*

- Tacómetro de retroalimentación de 1 grado de exactitud.
- 160-170 RPM.
- Torque de 20 N/cm.
- Stall torque de 40 N/cm.

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

Figura 12.*Motor Grande*

Nota: (LEGO®MINDSTORMS®, 2013)

2.4.2 Motor mediano EV3

El motor mediano también incluye un sensor de rotación incorporado (con resolución de 1 grado), pero es más pequeño y más liviano que el motor grande. Esto significa que puede responder más rápidamente que el motor grande.

Además, el motor mediano (figura 13) puede programarse para encenderse o apagarse, controlar su nivel de energía o para funcionar durante una cantidad de tiempo o de rotaciones especificadas (LEGO®MINDSTORMS®, 2013).

Figura 13.

Motor Mediano



Nota: (LEGO®MINDSTORMS®, 2013)

2.4.3 Sensor de Color

El sensor de color (figura 14) es un sensor digital que puede detectar el color o la intensidad de la luz que ingresa por la pequeña ventana de la cara del sensor. Este sensor puede utilizarse en tres modos diferentes:

Figura 14.

Sensor de color



Nota: (LEGO®MINDSTORMS®, 2013)

En “**Modo color**”, el sensor de color (figura 15) reconoce siete colores: negro, azul, verde, amarillo, rojo, blanco y marrón, además de sin color. Esta capacidad de diferenciar los colores significa que su robot puede estar programado para clasificar pelotas o bloques de colores, decir los nombres de los colores a medida que los detecta o para detenerse cuando detecta el color rojo (LEGO®MINDSTORMS®, 2013).

Figura 15.

Sensor en Modo color



Nota: (LEGO®MINDSTORMS®, 2013)

En “**Modo intensidad de la luz reflejada**”, el sensor de color mide la intensidad de la luz (figura 16) que se refleja desde una lámpara emisora de luz color rojo. El sensor utiliza una escala de 0 (muy oscuro) a 100 (muy luminoso). Esto significa que su robot puede estar programado para moverse sobre una superficie blanca hasta detectar una línea negra o para interpretar una tarjeta de identificación con código de color (LEGO®MINDSTORMS®, 2013).

Figura 16.

Sensor en Modo Intensidad de Luz Refleja



Nota: (LEGO®MINDSTORMS®, 2013)

En “**Modo intensidad de la luz ambiental**”, el sensor de color mide la intensidad de la luz (Figura 17) que ingresa en la ventana desde su entorno, como la luz del sol o el haz de una linterna. El sensor utiliza una escala de 0 (muy oscuro) a 100 (muy luminoso). Esto significa que su robot puede estar programado para emitir una alarma al salir el sol por la mañana o para detenerse si las luces se apagan (LEGO®MINDSTORMS®, 2013).

Figura 17.

Sensor Modo Intensidad de Luz Ambiental.



Nota: (LEGO®MINDSTORMS®, 2013)

2.4.4 Sensor de Contacto

El sensor táctil (figura 18) es un sensor analógico que puede detectar el momento en el que se presiona y se lanza el botón rojo del sensor. Esto significa que el Sensor táctil puede programarse para actuar según tres condiciones: presionado, lanzado o en contacto (tanto presionado como lanzado) (LEGO®MINDSTORMS®, 2013).

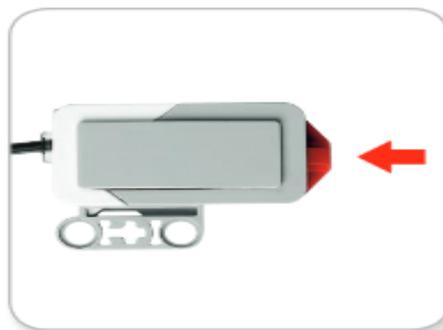
Figura 18.

Sensor Táctil



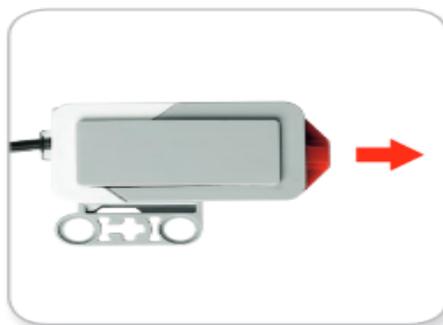
Nota: (LEGO®MINDSTORMS®, 2013)

Con la información del sensor táctil, se puede programar un robot para ver el mundo como lo haría una persona no vidente, es decir, extendiendo un brazo y respondiendo cuando toca algo Presionado, (figura 19), (LEGO®MINDSTORMS®, 2013).

Figura 19.*Sensor Táctil Presionado*

Nota: (LEGO®MINDSTORMS®, 2013)

Se puede construir un robot con un sensor táctil presionado contra la superficie figura 20. Luego, puede programar el robot para que responda (se detenga) cuando esté a punto de pasar el borde de la mesa (cuando el sensor se lanza).

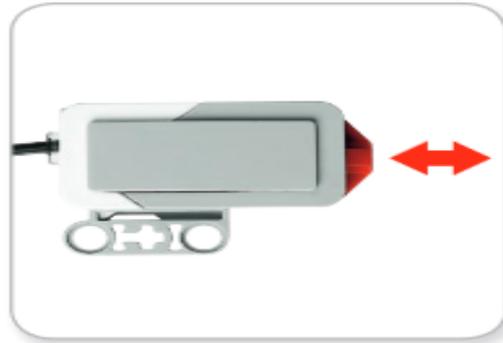
Figura 20.*Sensor Táctil Lanzado*

Nota: (LEGO®MINDSTORMS®, 2013)

Un robot de pelea puede programarse para continuar empujando hacia adelante en dirección a su oponente hasta que este se retire. Ese par de acciones, presionado y lanzado, constituyen el estado en contacto (figura 21).

Figura 21.

Sensor Táctil en Contacto



Nota: (LEGO®MINDSTORMS®, 2013)

2.4.5 Sensor Infrarrojo

El sensor de infrarrojos es digital (figura 22), detecta la proximidad del robot respecto al transmisor infrarrojo remoto (baliza) o bien, a algún obstáculo mediante la luz infrarroja, se puede programar de tres modos diferentes (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018). A continuación se detalla en la tabla número 7.

Figura 22.

Sensor Infrarrojo



Nota: (LEGO®MINDSTORMS®, 2013)

Tabla 7.*Modos de programar el sensor infrarrojo*

<ul style="list-style-type: none"> - Modo Proximidad: el sensor emite ondas de luz que al ser reflejadas obtiene la representación de la distancia entre el objeto que refleja las ondas y el sensor. La escala que usa va de 0 a 100. - Modo Baliza: el sensor detecta al transmisor infrarrojo remoto por medio de las señales enviadas por este siempre que se encuentren en el mismo canal(1-4). - Modo remoto: el sensor recibe información enviada por el transmisor de infrarrojos remoto que puede corresponder a los botones que contiene este último.

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

2.4.6 Transmisor De Infrarrojo Remoto (Baliza)

El transmisor IR remoto (figura 23) puede ser usado para controlar el robot a distancia con el modo remoto, pero su diseño fue creado para trabajar en conjunto con el sensor infrarrojo, sus especificaciones se muestran en la tabla siguiente:

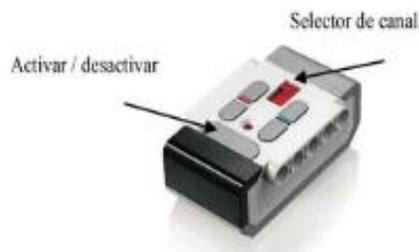
Tabla 8.*Especificaciones del transmisor infrarrojo remoto*

Cuatro canales individuales.
Incluye un botón para activar o desactivar.
LED verde que indica si el transmisor IR remoto está activo.
Autoapagado si la unidad no está en acción durante una hora.
Alcance de hasta dos metros.
Alimentación de 2 baterías AAA.

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

Figura 23.

Transmisor de IR remoto



Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

2.5 Conexiones del Bloque EV3**2.5.1 Conexión de Sensores y Motores**

Es necesario conectar los motores y sensores al Bloque EV3 para que funcionen (LEGO®MINDSTORMS®, 2013).

Con los cables conectores negros planos, conectar los sensores al Bloque EV3 mediante los puertos de entrada 1, 2, 3 y 4 (figura 24).

Figura 24.

Conexión de Sensores.



Nota: (LEGO®MINDSTORMS®, 2013)

Al crear programas mientras el Bloque EV3 no está conectado a su dispositivo, el software asignará sensores a los siguientes puertos predeterminados como se muestra en la siguiente tabla:

Tabla 9.

Especificación de uso predeterminado de los puertos de entrada

Puerto 1: Sensor táctil
Puerto 2: Girosensor/Sensor de temperatura
Puerto 3: Sensor de color
Puerto 4: Sensor ultrasónico/Sensor infrarrojo

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

Si el Bloque EV3 está conectado al dispositivo mientras realiza la programación, el software/Aplicación de programación EV3 identificará automáticamente qué puerto está utilizando para cada sensor y motor.

Con los cables conectores negros planos, conectar los motores al Bloque EV3 mediante los puertos de salida A, B, C y D (figura25) (LEGO®MINDSTORMS®, 2013).

Figura 25.

Conexión de Motores



Nota: (LEGO®MINDSTORMS®, 2013)

Al igual que en el caso de los sensores, si el bloque EV3 no está conectado mientras escribe un programa, cada motor será asignado a los siguientes puertos predeterminados como se muestra en la tabla 10:

Tabla 10.

Especificación de uso predeterminado de los puertos de salida

Puerto A: Motor mediano

Puertos B y C: Dos Motores grandes

Puerto D: Motor grande

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

Si el bloque EV3 está conectado a un dispositivo mientras realiza la programación, el software/Aplicación de programación EV3 asignará automáticamente el puerto correcto a los programas (LEGO®MINDSTORMS®, 2013).

2.5.2 Conexión Bloque EV3 a su equipo

El **Bloque EV3** se puede conectar con el ordenador que contiene el software LEGO EV3 mediante 3 formas:

2.5.2.1 Conexión cable USB

Para conectar el bloque mediante un cable USB hay que seguir los pasos de la siguiente tabla:

Tabla 11.

Pasos de conexión del bloque hacia la PC

-
1. Conectar el extremo mini USB al puerto para PC del bloque EV3 situado junto al puerto D.
 2. El otro extremo del cable USB se conecta al ordenador.
 3. En la pestaña Bloques EV3 disponibles del panel de Hardware se mostrará automáticamente el bloque conectado por USB al ordenador como muestra en la figura 26.
-

Nota: (Prieto, 2015)

Figura 26.

Bloque del panel conectado por USB.



Nota: (Prieto, 2015)

2.5.2.2 Conexión Bluetooth

El ordenador debe disponer de funcionalidad **Bluetooth** o bien necesitará un conector USB para **Bluetooth**. Los pasos para establecer la conexión BT son:

1. Activar la funcionalidad **Bluetooth** (802.15.1) en el bloque EV3 en la pantalla de **Ajustes**.
2. Desde el **panel de Hardware** del **software de Lego EV3** haz clic en la pestaña **Bloques EV3 disponibles** (figura 27).

Figura 27.

Pestaña de bloque EV3 disponibles.



Nota: (Prieto, 2015)

3. Si el bloque EV3 aún no aparece en la lista de dispositivos detectados hacer clic en el botón **Actualizar** para localizar el bloque EV3 y seleccionar la casilla de verificación Bluetooth.

4. En el bloque EV3 aceptar la conexión BT de forma manual, introducir la clave de paso y presionar el botón **Centro** para aceptar. El valor de la contraseña suele ser 1234. Repetir estos pasos en el software de Lego EV3 para completar la conexión entre ambos dispositivos.
5. Cuando se haya establecido conexión se mostrará un símbolo "<>" en la parte superior izquierda de la pantalla del bloque EV3 que confirma la conexión. En el panel de **Hardware** del software Lego EV3 también se mostrará la conexión como se muestra en la figura 28 (Prieto, 2015).

Figura 28.

Muestra de la conexión de Bluetooth



Nota: (Prieto, 2015)

2.5.2.3 Conexión WIFI

El requisito previo para conectar el brick EV3 a la red inalámbrica es disponer de un dongle USB para WIFI. El bloque EV3 no dispone de WIFI integrada y el dispositivo USB para WIFI. Es necesario comprarlo aparte. De momento Lego EV3 sólo reconoce la marca **NetGear** y el modelo **WNA1110 Wireless-N 150**.

1. Insertar el conector USB para WIFI en la ranura USB del brick de Lego EV3 como se muestra a continuación en la figura 29.

Figura 29.

USB introducido en la ranura brick de LEGO

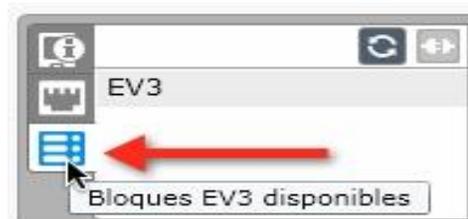


Nota: (Prieto, 2015)

2. En el bloque acceder a la página de **Ajustes** y dentro de ella activar la conexión **WIFI**.
3. Introduce el nombre y contraseña de la red inalámbrica para conectar el bloque.
4. Si la conexión ha tenido éxito se mostrará un icono de WIFI en la esquina superior izquierda de la pantalla del bloque EV3.
5. Desde el **panel de Hardware** del **software de Lego EV3** hacer clic en la pestaña **Bloques EV3 disponibles** como muestra en la figura 30 (Prieto, 2015).

Figura 30.

Pestaña de bloque EV3 disponibles



Nota: (Prieto, 2015)

6. Si el bloque EV3 aún no aparece en la lista de dispositivos detectados haz clic en el botón **Actualizar** para localizar el bloque EV3 y seleccionar la casilla de verificación WIFI que aparecerá.
7. En el panel de **Hardware** del software Lego EV3 también se mostrará la conexión en la pestaña del bloque como se muestra en la figura 31 (Prieto, 2015).

Figura 31.

Muestra de la conexión WIFI.



Nota: (Prieto, 2015)

2.5.3 Conexión del Bloque EV3 a un dispositivo android

Para descargar y ejecutar los programas que se hayan creado en la Aplicación de programación EV3, conectar un dispositivo Android al Bloque EV3 a través de Bluetooth se detallarán los siguientes pasos:

1. Confirme que el Bloque EV3 esté encendido.
2. Active Bluetooth en el Bloque EV3 (figura 32) y asegúrese de que la configuración del iPhone/iPad/iPod no esté seleccionada.

Figura 32.

Pantalla muestra configuración de Bluetooth



Nota: (LEGO®MINDSTORMS®, 2013)

3. Asegurar que esté activado el Bluetooth en el dispositivo android y abrir un programa nuevo o uno existente en la Aplicación de programación EV3.

4. Tocar el icono No hay dispositivos conectados en la esquina superior derecha como muestra en la figura 33.

Figura 33.

Icono de no hay dispositivos conectados.



Nota: (LEGO®MINDSTORMS®, 2013)

5. Pulsar el botón "Conectar".

6. Seleccionar el Bloque EV3 al que se desea conectar de la lista de bloques disponibles.

7. Aceptar la conexión en el Bloque EV3 de manera manual (figura 34), luego, ingrese la clave de paso y presione el botón Centro para aceptar. El valor predeterminado es 1234. Introduzca la misma clave de paso en la Aplicación de programación EV3.

8. Al establecer la conexión, se muestra un símbolo “<>” en la parte superior izquierda de la pantalla del Bloque EV3 (junto al icono de Bluetooth) que confirma la conexión. (LEGO®MINDSTORMS®, 2013).

Figura 34.

Pantalla mostrando la conexión de forma manual



Nota: (LEGO®MINDSTORMS®, 2013)

CAPÍTULO III

DESARROLLO DEL TEMA

3.1 Introducción

El proceso para el ensamblaje y telecontrol del robot móvil con el kit de LEGO Mindstorm en su versión doméstica, se detallará paso a paso en este capítulo además de la programación de diferentes acciones utilizando el software propio de Lego.

También se realizará el testeo de cada uno de los sensores del kit de Lego para su implementación en el telecontrol del robot y su aplicación del entorno de desarrollo de software en APP Inventor y un API (Interfaces de Programación de Aplicaciones) realizado en el software leJOS JAVA.

3.2 Ensamblaje del chasis del robot móvil.

Para el diseño de construcción se necesita diferentes tipos de piezas provenientes del kit de piezas en su versión doméstica (#31313) de LEGO (**ANEXO A**). Luego proceder a la implementación del robot móvil “EV3MEG” para realizar las pruebas de testeo con cada uno de los sensores lo cual se detalla paso a paso a continuación.

3.2.1. Montaje del soporte del motor grande derecho y acople de ruedas.

En primer lugar seleccionar las piezas que se van utilizar, el número marcado junto con la X es la cantidad de piezas requerido que se va a utilizar para el ensamblaje (figura 35).

Figura 35.

Comienzo del ensamblaje.



Nota: (Boogaarts, 2013)

Una vez conectado las primeras piezas se realiza el ensamblaje para acople de la rueda delantera derecha utilizando espigas de conexión con fricción, vigas angulares, bloque angular, un eje de modulo de 6; neumático y pieza central de rueda (figura 36).

Figura 36.

Pasos y piezas para el acople de la rueda



Nota: (Boogaarts, 2013)

Luego se procede a colocar otras piezas del kit como clavijas, espigas de conexión y vigas angulares como se muestra en la figura 37, para el apoyo de acople de la rueda delantera derecha con el motor grande.

Figura 37.

Piezas y montaje de la rueda derecha delantera.

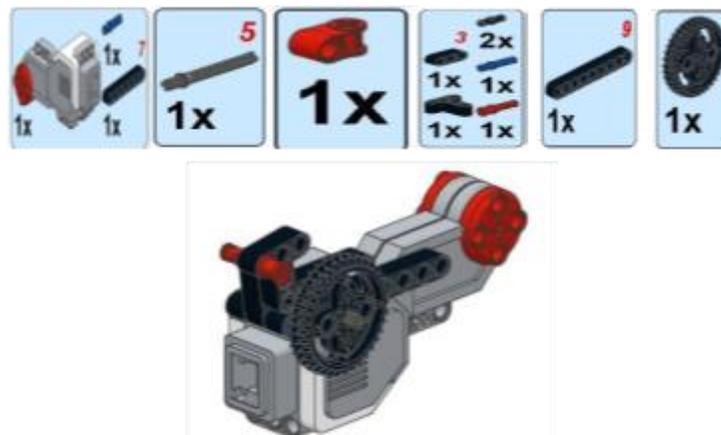


Nota: (Boogaarts, 2013)

Como se muestra en la figura 38 se ensambla el motor grande derecho para posteriormente unirlo al montaje del chasis con la rueda delantera derecha.

Figura 38.

Motor mediano derecho acoplado

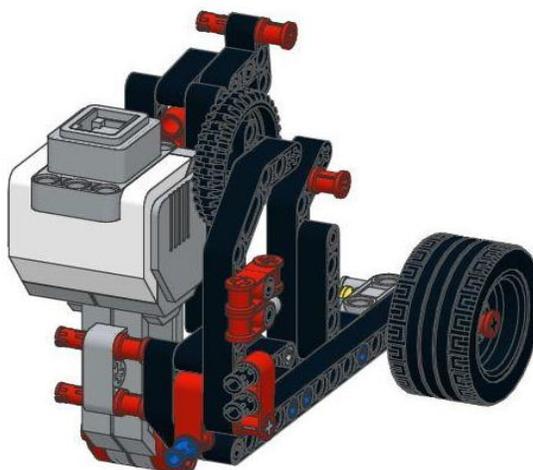


Nota: (Boogaarts, 2013)

Acoplar el motor grande derecho con el montaje del chasis de la rueda delantera tal como se muestra en la figura 39.

Figura 39.

Chasis montaje con rueda delantera.

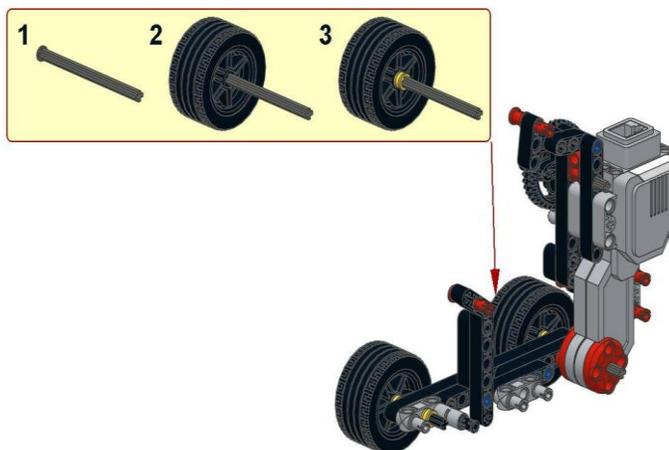


Nota: (Boogaarts, 2013)

Proceder acoplar la rueda trasera derecha al chasis con el motor grande ensamblado como se muestra en la figura 40.

Figura 40.

Rueda trasera montada al chasis

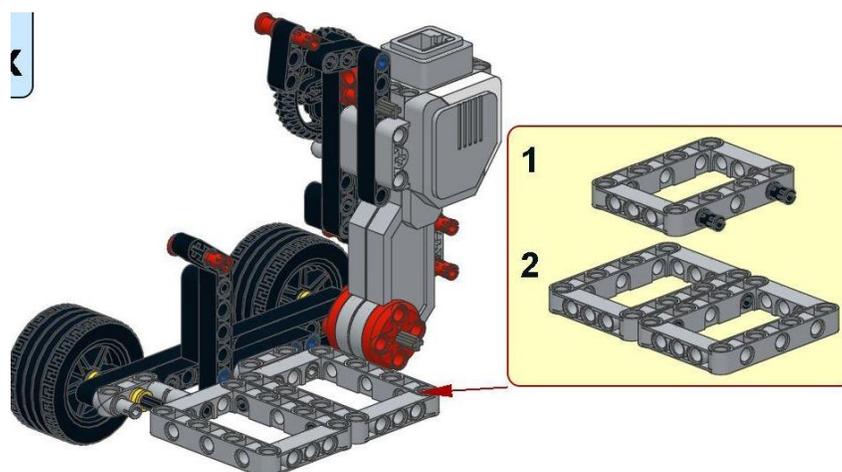


Nota: (Boogaarts, 2013)

Armar de igual forma la parte del montaje para el motor grande izquierdo donde constan de diferentes piezas estructurales (**ANEXO B**). Posteriormente como se muestra en la figura 41, se acopla dos planchas agujereadas al chasis central de abajo del robot este servira para el acople de la parte izquierda del robot.

Figura 41.

Acoplamiento del chasis de abajo del robot.



Nota: (Boogaarts, 2013)

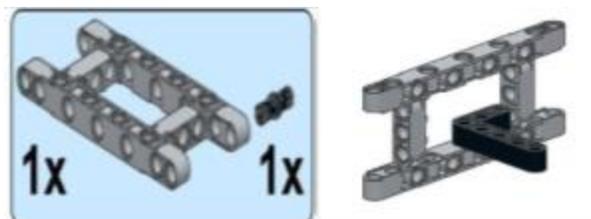
3.2.2 Montaje del soporte para el motor mediano.

El montaje del soporte para el motor mediano, será incrustado en el centro del chasis esto ayudará al acople tanto de los brazos y la parte izquierda del robot EV3MEG.

Para esto en primer lugar se acopla una plancha agujereada, una espiga de conexión además de una viga angular como se muestra en la figura 42.

Figura 42.

Montaje de las primeras piezas del soporte.

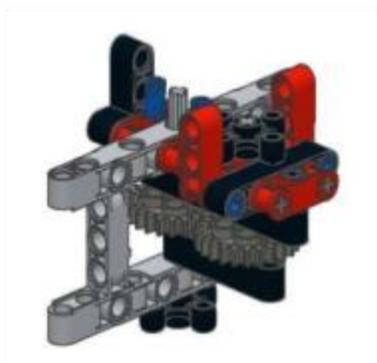


Nota: (Boogaarts, 2013)

Después añadir las demás piezas faltantes de la estructura central del chasis como: engranajes, ejes con módulo de diferente tamaño, espigas de conexión y vigas tal como se muestra en la figura 43.

Figura 43.

Piezas faltantes para el montaje del soporte del chasis central.

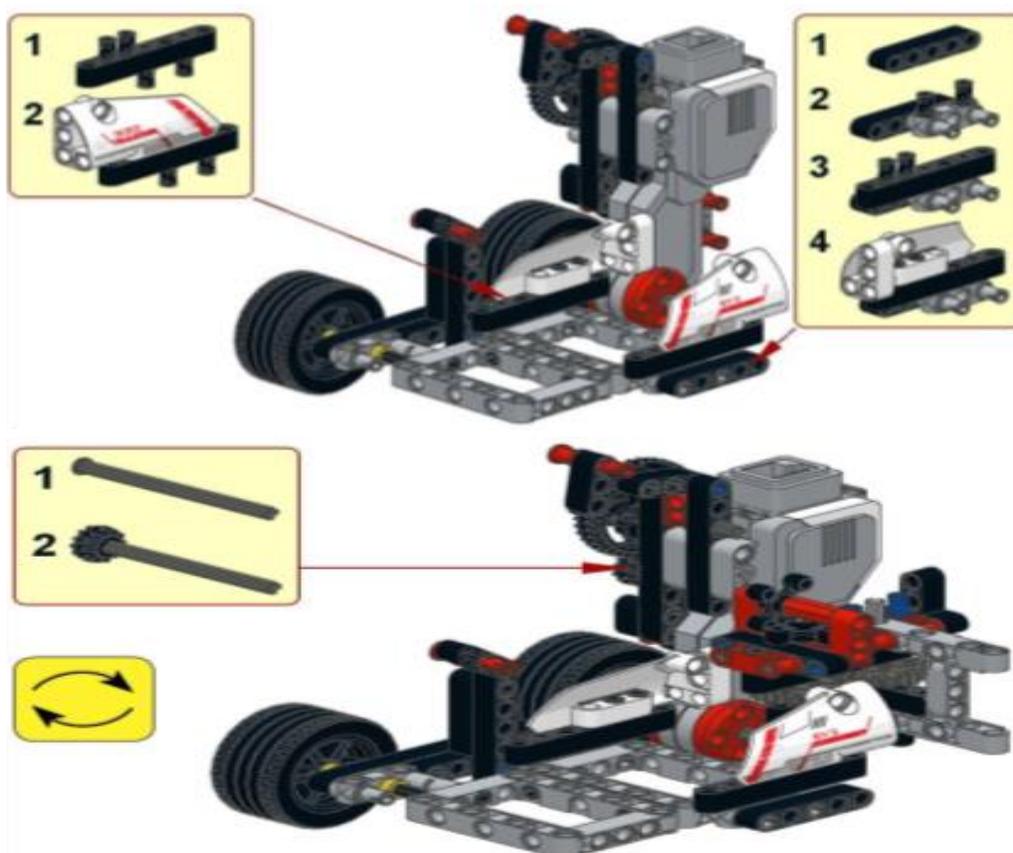


Nota: (Boogaarts, 2013)

Luego de haber culminado el montaje central para soporte del motor mediano se lo acopla al chasis del motor grande derecho, (**ANEXO C**), además se debe utilizar engranajes; un eje de módulo gris, un bloque angular, vigas, paneles curvados(derecha-izquierda) y un conector doble para que se pueda colocar en el chasis central del robot.(Figura 44)

Figura 44.

Montaje del soporte del motor grande al chasis general.

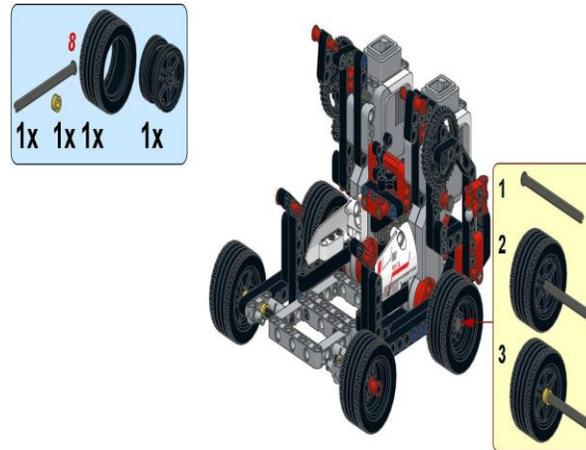


Nota: (Boogaarts, 2013)

Como se muestra en la figura 45 se añade la parte del montaje del chasis izquierdo con su rueda delantera, luego se acopla el montaje del motor grande izquierdo con sus rueda trasera.

Figura 45.

Colocación izquierda del motor grande con sus respectivas ruedas.



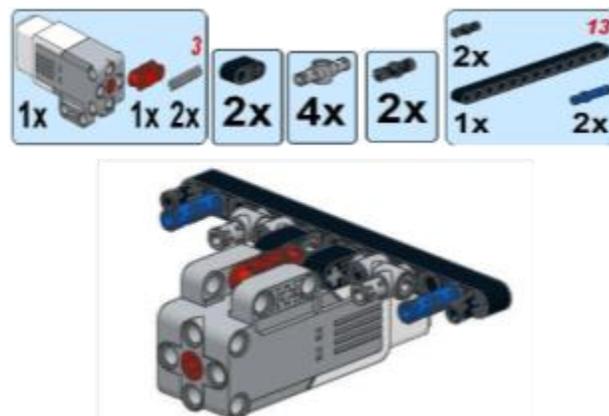
Nota: (Boogaarts, 2013)

3.2.3. Acople del motor mediano al chasis de los motores izquierda y derecha.

Para el acople del motor mediano se necesita ésta vez de un eje de módulo color gris, de bloques transversales (doble y módulo de 2); espigas de conexión simples y una viga de módulo de 13 agujeros como se muestra en la figura 46.

Figura 46.

Acople del motor mediano

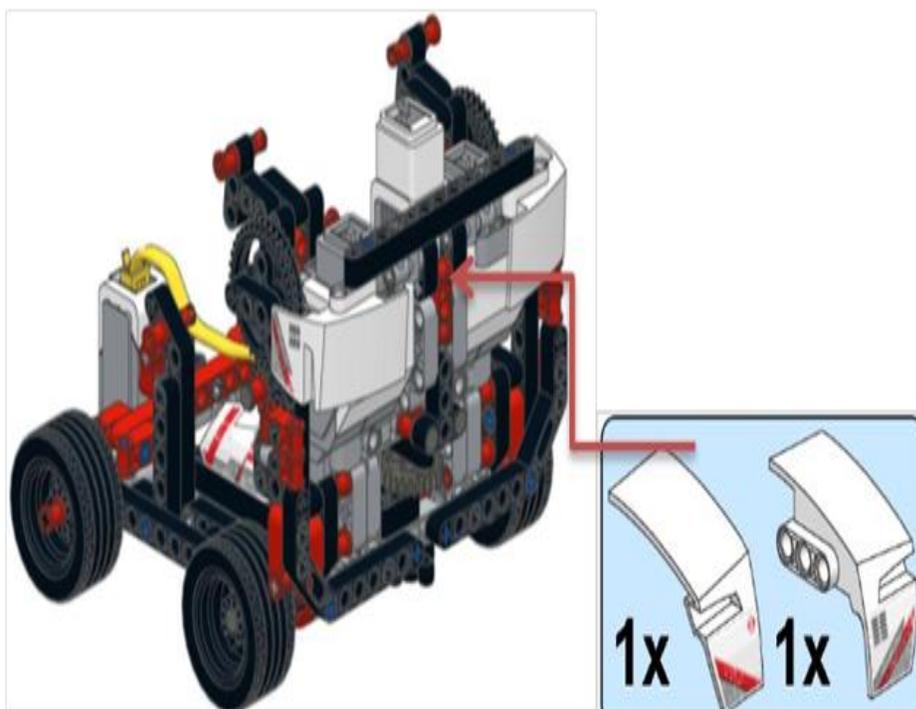


Nota: (Boogaarts, 2013)

Colocar dos espigas de conexión en forma de aspas angulares a los lados del motor mediano montado en el chasis central figura 47, para luego acoplar el mismo en el chasis de los motores grandes como se muestra en el (ANEXO D).

Figura 47.

Colocación del motor mediano con sus aspas angulares y acople al chasis.

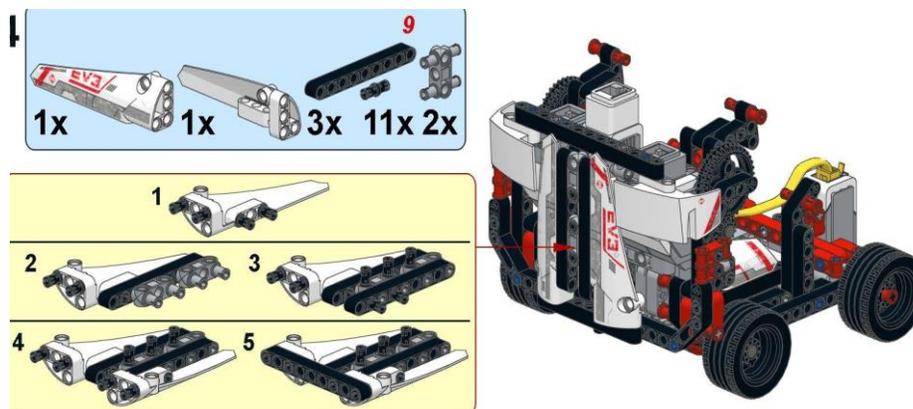


Nota: (Boogaarts, 2013)

Colocar tres vigas de 9 agujeros, espigas de conexión en las dos aspas rectas que irán acoplar atrás del motor mediano, luego se ensambla en el chasis de los motores grandes como se muestra en la figura 48.

Figura 48.

Colocación de dos aspas rectas atrás del motor mediano.



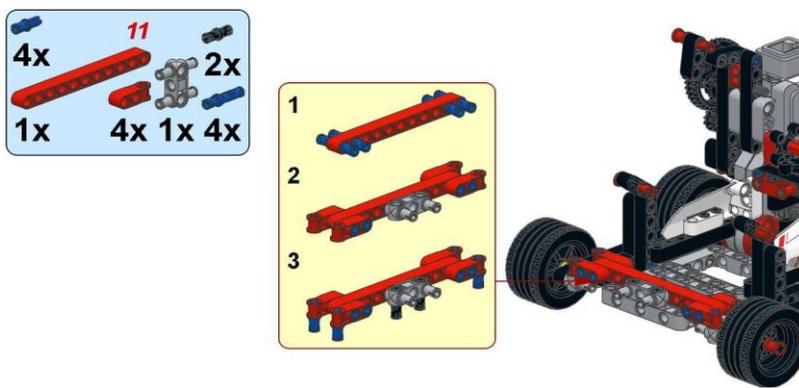
Nota: (Boogaarts, 2013)

3.2.4. Montaje de la parte frontal del robot móvil con el sensor de color.

Para este montaje es necesario que el chasis con los motores (grande y 2 medianos) esté completamente armado. Luego como muestra en la figura 49 se procede a utilizar una viga de 11 agujeros, espigas de conexión simples; una espiga de conexión doble y un bloque transversal color rojo para el montaje de la parte frontal del robot móvil.

Figura 49.

Montaje de las piezas de la parte frontal del robot.

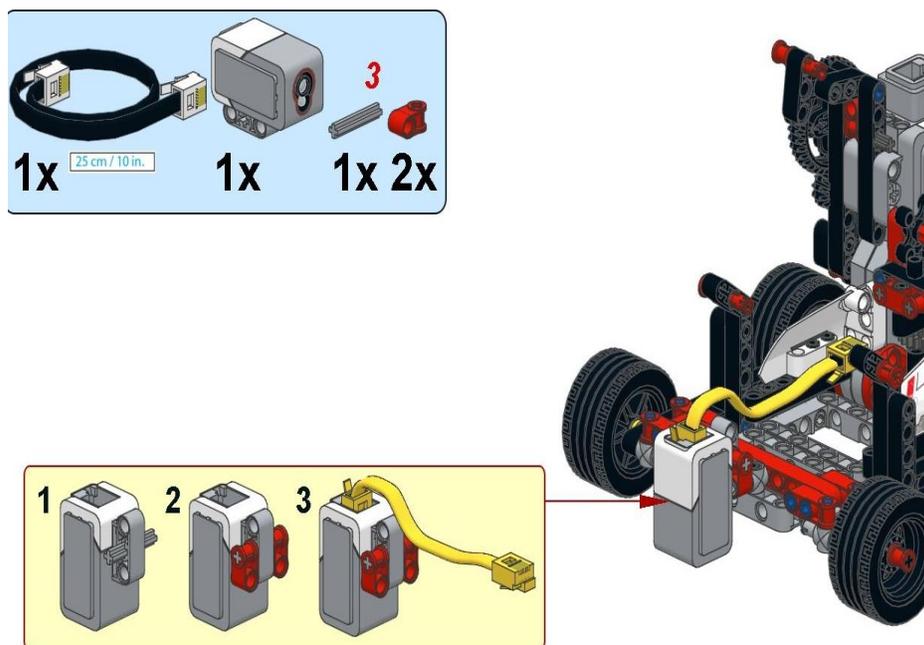


Nota: (Boogaarts, 2013)

Acoplado las piezas en la parte frontal del robot móvil se procede a montar el sensor de color y luminosidad con la ayuda de una espiga simple y dos bloques transversales color rojo como se muestra en la figura 50.

Figura 50.

Ensamblaje del sensor de color y luminosidad en el robot



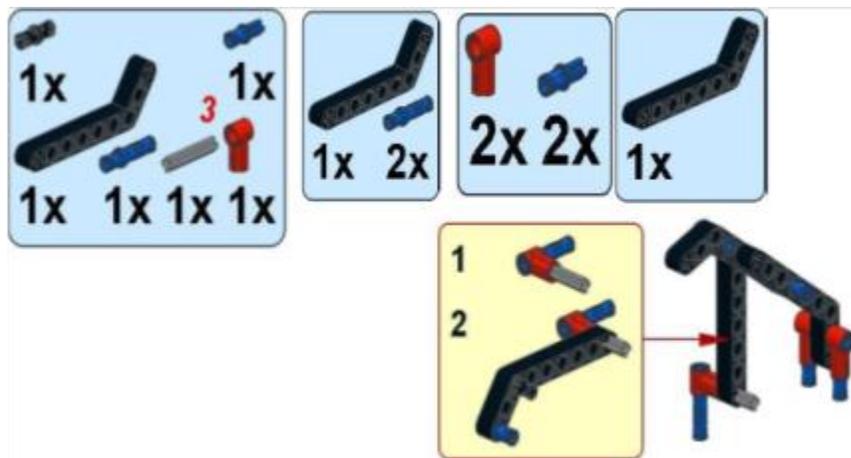
Nota: (Boogaarts, 2013)

3.2.5. Montaje del sensor infrarrojo al cuerpo del robot móvil.

Hay que declarar que esta parte será primordial para la cabeza y el tronco que se visualizará en el robot móvil. Para el acople del sensor se necesita primero armar el tronco o cuello como se muestra en la figura 51 que sujetará al mismo con la ayuda de tres vigas angulares; espigas de conexión (simples, con fricción, con eje) y tres bloques transversales simples.

Figura 51.

Acoplamiento del tronco del sensor infrarrojo para sujetarlo

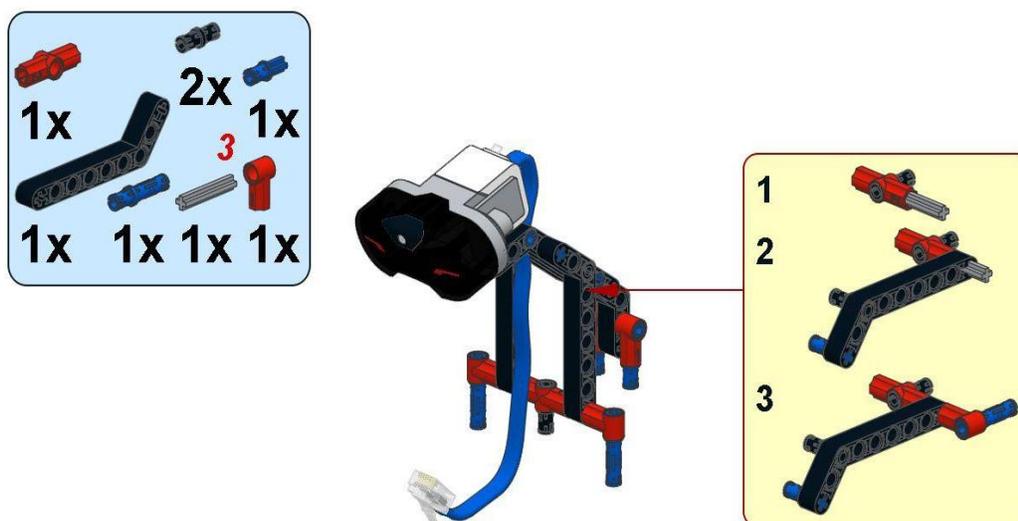


Nota: (Boogaarts, 2013)

Posteriormente se conectan las piezas faltantes para culminar de acoplar de manera que este sujetado el sensor infrarrojo como se muestra en la figura 52.

Figura 52.

Sujeción del sensor infrarrojo con el tronco armado.

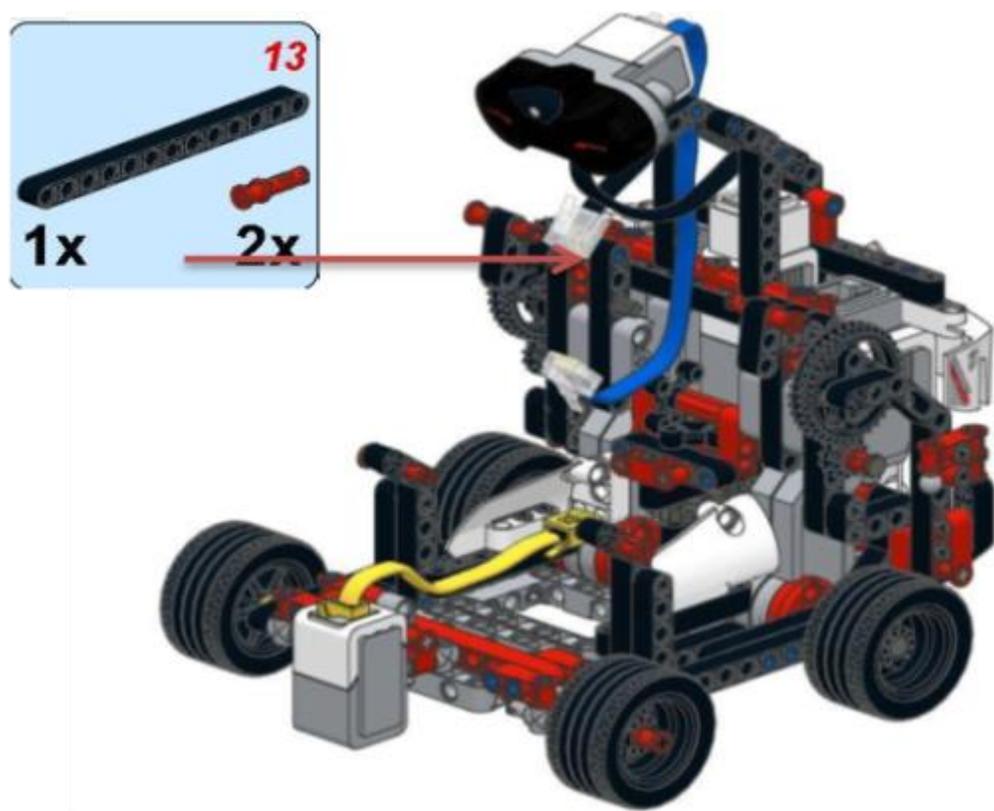


Nota: (Boogaarts, 2013)

En la figura 53 se culmina acoplando las últimas tres piezas de la estructura del sensor infrarrojo conectándolo al robot móvil siendo este la cabeza y tronco.

Figura 53.

Montaje del sensor infrarrojo en el robot móvil.



Nota: (Boogaarts, 2013)

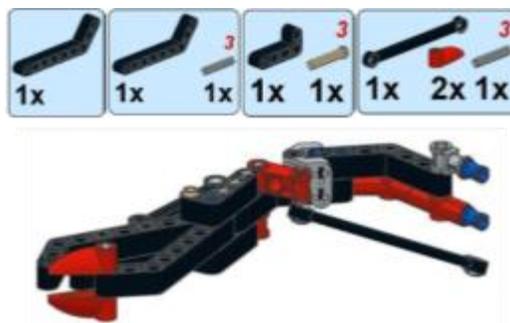
3.2.6. Montaje del brazo móvil para el robot EV3MEG.

Para este montaje primero se arma el acople del brazo como se muestra en la figura 54 con la ayuda de bloques angulares color rojo, espigas de conexión simples y vigas angulares.

pinzas o tenazas que serán montadas al robot móvil EV3MEG como se muestra en la figura 56.

Figura 56.

Ensamblaje de las pinzas del robot móvil.



Nota: (Boogaarts, 2013)

Hay que argumentar que el otro brazo robótico faltante se debe realizar de la misma manera siguiendo los pasos anteriores (**ANEXO E**). Para finalizar en la figura 57 se indica cómo se tiene que montar ambos brazos robóticos al robot.

Figura 57.

Acoplamiento de los brazos robóticos al EV3MEG.



Nota: (Boogaarts, 2013)

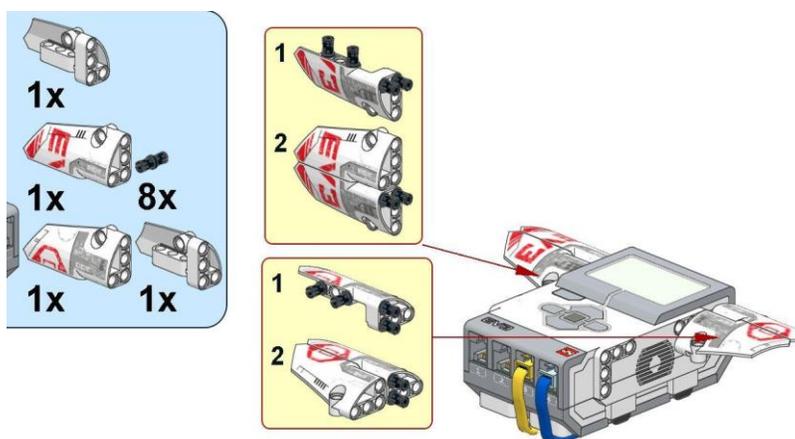
Para finalizar con el ensamblaje total del robot se necesita acoplar el bloque EV3, y además conectar mediante su cableado propio los sensores y motores en cada uno de sus puertos conectores del bloque.

3.3 Ensamblaje del Bloque EV3 y cableado de sensores y motores.

Primeramente se acopla en los dos lados laterales del bloque EV3 dos pares de piezas de aspas rectas de 4x2 con ocho espigas de conexión como se muestra en la figura 58, estas serán la parte torácica del robot móvil.

Figura 58.

Acople de aspas rectas en el bloque EV3.



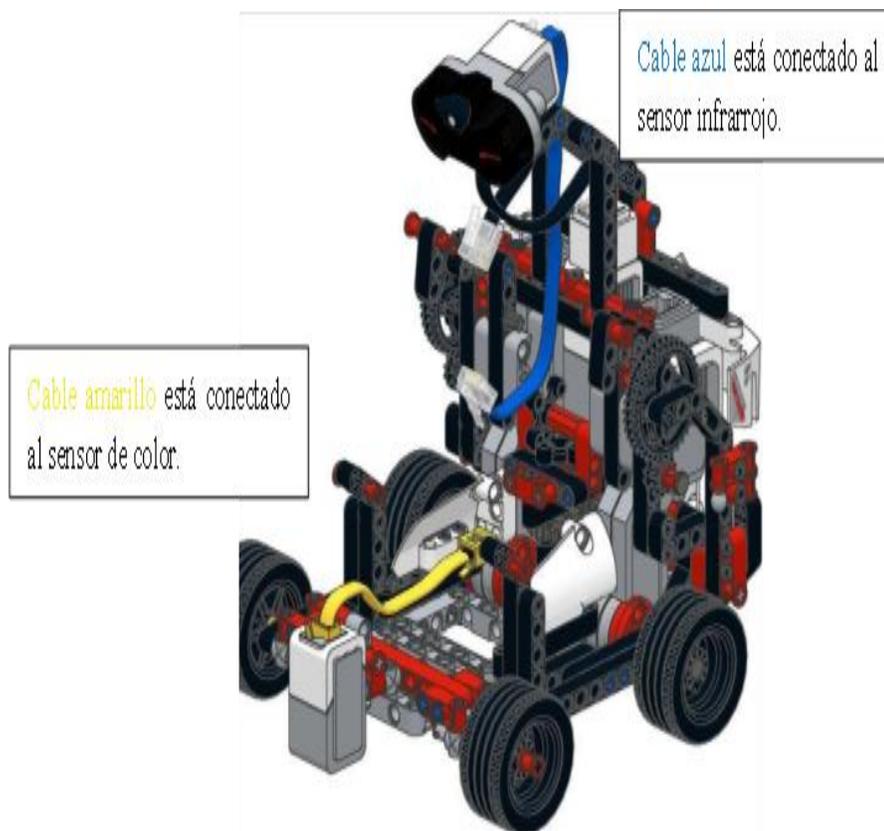
Nota: (Boogaarts, 2013)

3.3.1 Cableado y Conexión de sensores (infrarrojo y luminosidad).

En la figura anterior en el bloque EV3 se puede observar en los puertos de entrada 3 y 4 respectivamente dos cableados que sobresalen en la parte inferior una de color azul y la otra de color amarillo estos son los cables (figura 59) que se conectarán al sensor infrarrojo y de luminosidad o de color en sus respectivos puertos de entrada.

Figura 59.

Demostración de los sensores con sus respectivos cableados



Nota: (Boogaarts, 2013)

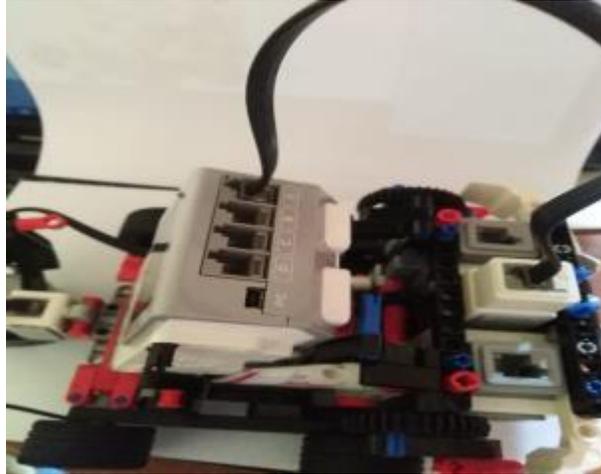
3.3.2 Cableado y Conexión de los puertos de salida para los motores.

En primer lugar se necesita saber cuáles son los puertos de salida correspondientes a cada motor revisar la Tabla 9, para proceder a la unión de los motores al bloque EV3.

Luego de revisar la tabla se conectan los cables de cada motor en los puertos de salida. El puerto A corresponde al motor mediano como se muestra en la figura 60.

Figura 60.

Conexión del Puerto A al motor mediano



Posteriormente para los motores grandes que darán movimiento a las ruedas principales le corresponden los puertos de salida B y C respectivamente como se muestra en la figura 61.

Figura 61.

Conexión de los puertos B y C a los motores grandes



3.4 Programación en el software LEGO EV3.

Para realizar la programación de los sensores y motores se necesita tener instalado el software LEGO MINDSTORMS EV3 Home Edition en la PC o MAC.

Para obtener el software e instalarlo se puede trasladar al siguiente link <https://www.robotix.es/es/descargar-software-lego-mindstorms-education-ev3>, luego se conecta desde el bloque EV3 hacia la PC un cable USB como se muestra en la figura 62; que ayudará a transferir los datos de información de la programación adecuada de sensores y motores.

Cabe recalcar que para realizar las programaciones o calibración se necesita desmontar cada uno de los sensores del robot móvil y además de sencillas instrucciones de programa recomendadas por LEGO.

Figura 62.

Cable USB conectado del Bloque EV3 a la PC.

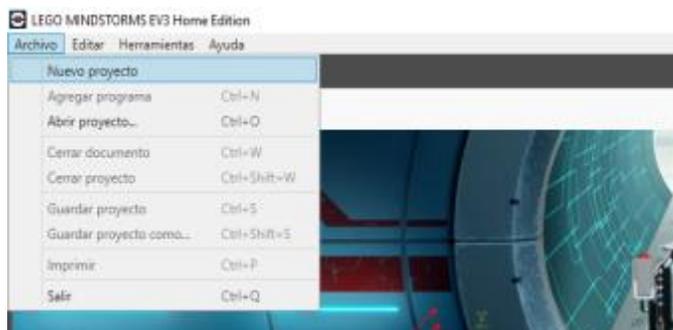


3.4.1 Programación para la calibración del sensor de color o luminosidad.

Para calibrar este sensor se ingresa en el software LEGO MINDSTORMS EV3 Home Edition, se mueve el cursor hacia la esquina superior izquierda en archivo y dar clic en nuevo proyecto como se muestra en la figura 63.

Figura 63.

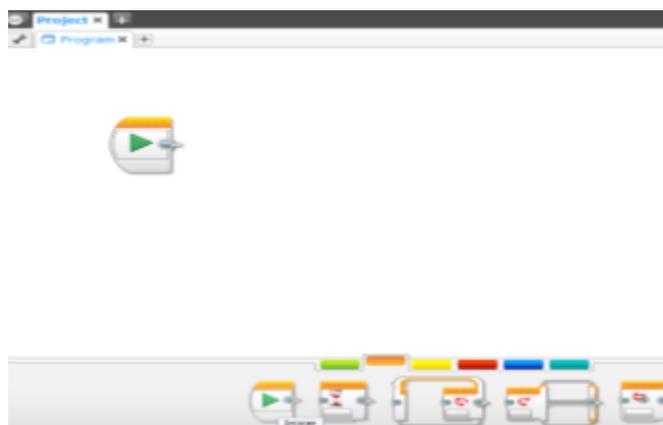
Entorno del software para el nuevo proyecto.



Luego de ingresar al nuevo proyecto se inicia la programación para la calibración del sensor. En la pantalla del nuevo proyecto se visualiza el bloque de flujo naranja (iniciar) por defecto como se muestra en la figura 64.

Figura 64.

Proyecto nuevo abierto con el bloque de flujo predeterminado.

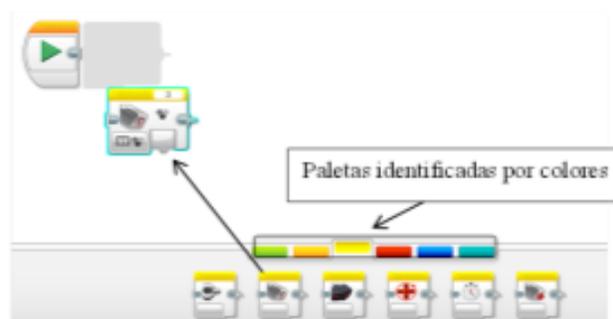


Los bloques que se utilizará en la programación están situados en la parte inferior del software esta consta de los iconos de paletas identificadas por colores característicos dependiendo su uso.

Como se muestra en la figura 65, el sensor de color está en el icono de la paleta de color amarillo, basta con arrastrar y colocar el bloque desde la parte inferior al área donde se está realizando la programación para la calibración.

Figura 65.

Selección de la paleta de colores el sensor de color.



Luego en la parte inferior izquierda del bloque del sensor de color dar clic y se abrirá un cuadro con tres opciones (medida-comparar-calibrar) se guía el cursor en calibrar, luego intensidad de luz reflejada y dar clic en reiniciar tal como se muestra en la figura 66.

Figura 66.

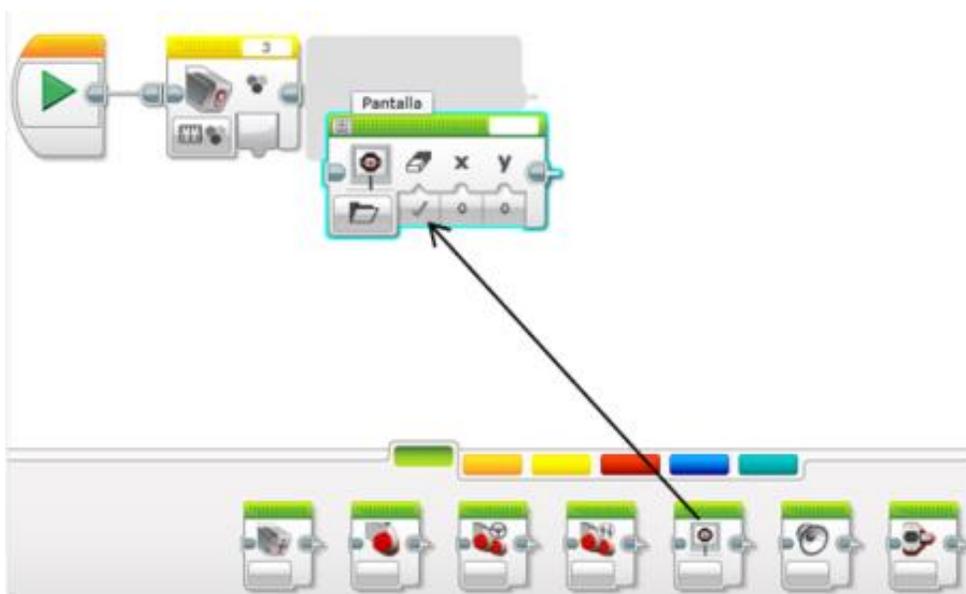
Pasos a seguir para iniciar calibración.



La opción reiniciar hará que los valores reflejados de luz en el sensor no estén definidos por tanto se debe definir los valores. A continuación se arrastra de los componentes de la paleta color verde, el bloque de pantalla como se muestra en la figura 67, que facilitará visualizar las instrucciones que el usuario tendrá que configurar.

Figura 67.

Colocación del bloque de pantalla en la programación



Se configura el bloque de pantalla deslizando el cursor en la parte inferior izquierda del bloque donde se visualizará diferentes opciones ir a texto, después clic en pixeles y en la parte superior derecha donde se encuentra la palabra MINDSTORMS se edita y se sobrescribe la palabra BLANCO tal como se muestra en la figura 68; para que el usuario visualice donde tiene que colocar el sensor en una superficie plana color blanca.

Figura 68.

Configuración del bloque de pantalla en la programación.



Luego se coloca en la programación un bloque de espera para que el usuario pueda tocar los botones del EV3 y se configura dicho bloque guiando el cursor en la parte inferior izquierda se desplazará un menú de opciones; ir a botones del bloque EV3 luego comparar y clic en botones del bloque EV3 tal como se muestra en la figura 69.

Figura 69.

Configuración del bloque de espera.



Posteriormente se configura el conjunto de ID del bloque de espera para que el usuario pueda presionar el botón del centro (2) y comparar cuando esté en contacto tal como se muestra en la figura 70.

Figura 70.

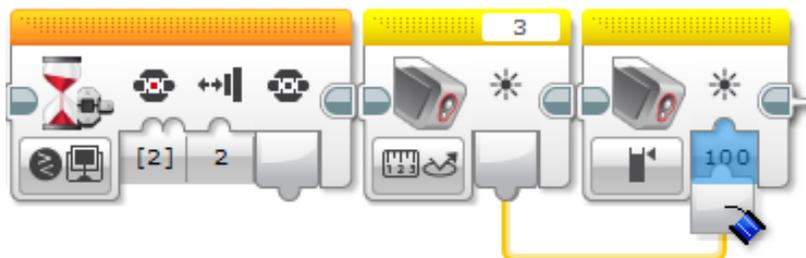
Configuración del conjunto ID y Estado del bloque de espera.



Partir después por medio del sensor que está en el puerto de entrada 3 a medir la cantidad de luz reflejada asignando bloques del sensor para calibrar dicha medida al 100 por ciento, como se muestra en la figura 71, es decir se visualizará la medida máxima de la parte de la superficie blanca.

Figura 71.

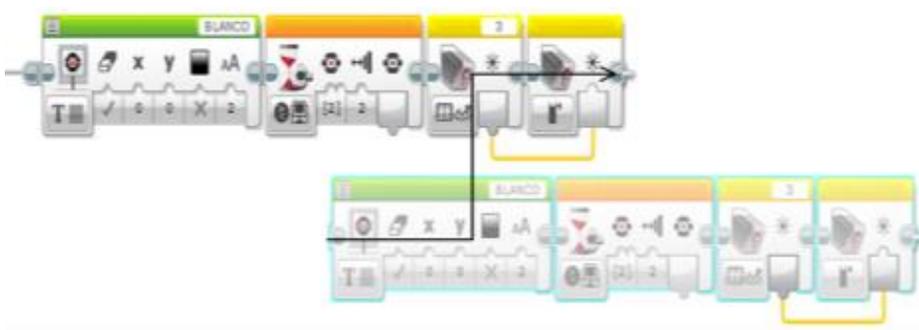
Colocar los bloques de sensor para calibrar y medir la cantidad de luz reflejada.



Después tal como se muestra en la figura 72, llevar el cursor a seleccionar los 4 últimos bloques que se configuraron anteriormente y con las teclas (ctrl+c) y (ctrl+v) copiar y pegar respectivamente estos bloques a continuación de los tratados.

Figura 72.

Copiar y pegar los 4 últimos bloques de programación.



Después de realizar lo anterior, como en la figura 73 se configura cada uno de los bloques (pantalla, espera y sensor), para visualizar y calibrar la medida de luz reflejada de la superficie negra, esta es la cantidad mínima de luz reflejada del sensor.

Figura 73.

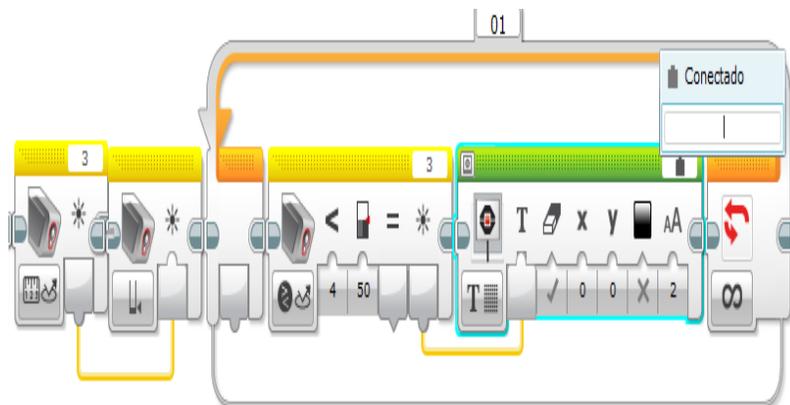
Configuración para visualizar y calibrar la luz reflejada en la superficie negra.



Para terminar la programación de calibración utilizar un bucle el cual contendrá dentro, un bloque del sensor de color, para configurarlo se debe llevar el cursor hacia la esquina inferior izquierda clic en comparar luego dar clic en intensidad de la luz reflejada para mostrar las medidas del sensor y también usar un bloque de pantalla; este último ayudará a visualizar dichos valores como se muestra en la figura 74.

Figura 74.

Últimos pasos para programar la calibración del sensor de color.



Terminado la programación se descarga hacia el Bloque físico EV3 llevando el cursor hacia el Panel de Hardware y luego se ubica en descargar tal como se muestra en la figura 75 y visualizar en la pantalla del panel del hardware los valores de la calibración de estado tanto para el color de la superficie blanca y negra.

Figura 75.

Panel de Hardware listo para enviar a descargar.



Por último se visualiza en el BLOQUE EV3 físico los valores de medida tanto para el color de superficie blanca y negra como se muestra en la siguiente figura. Tener en cuenta que con los demás colores que dispone la captación de luz del sensor de luminosidad o color se puede realizar de la misma manera.

Figura 76.

Valores visualizados en el bloque físico.



3.4.2 Programación del Sensor Táctil a través del software EV3

Al igual que el sensor de color, al realizar la programación respectiva para el uso adecuado del sensor táctil hay que recalcar que el bloque del sensor táctil no posee la opción calibrar ya que son más utilizados dos estados de medida 1 y 0 para su buen funcionamiento.

Además cabe señalar la opción Comparar Estado tiene tres tipos de selección para el sensor, que están registrados en un menú de opciones cada uno de estos se debe definir según los valores de estado convenientes. En la siguiente tabla se detalla a que estado pertenece cada uno.

Tabla 12.

Tipos de estado del sensor Táctil

0 = No está presionado
1 = Presionado
2 = En contacto

Nota: (LEGO®MINDSTORMS®, 2013)

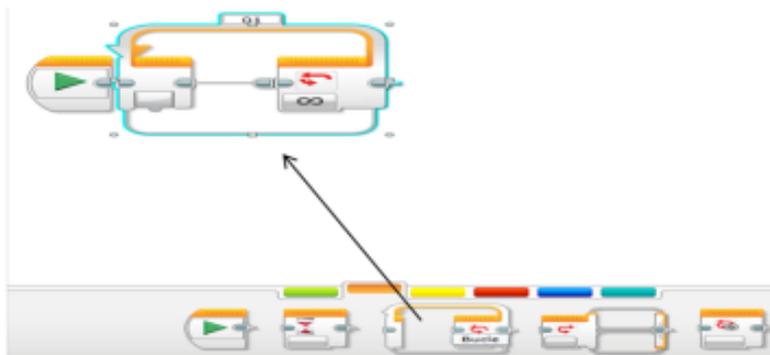
A continuación se detallará paso a paso como realizar la programación para el uso óptimo del sensor táctil.

Primero se debe ingresar a un nuevo proyecto, se inicia la programación, en la pantalla del nuevo proyecto se observa el bloque de flujo naranja (iniciar) por defecto, además los bloques que se utilizará en la programación están situados en la parte inferior del software esta consta de paletas identificadas por colores característicos dependiendo su uso.

Como se muestra a continuación en la figura 77, primero utilizar un bucle, está en la paleta de color naranja basta con arrastrar y colocar el bloque desde la parte inferior hacia el área donde se está realizando la programación para la buena utilización del sensor.

Figura 77.

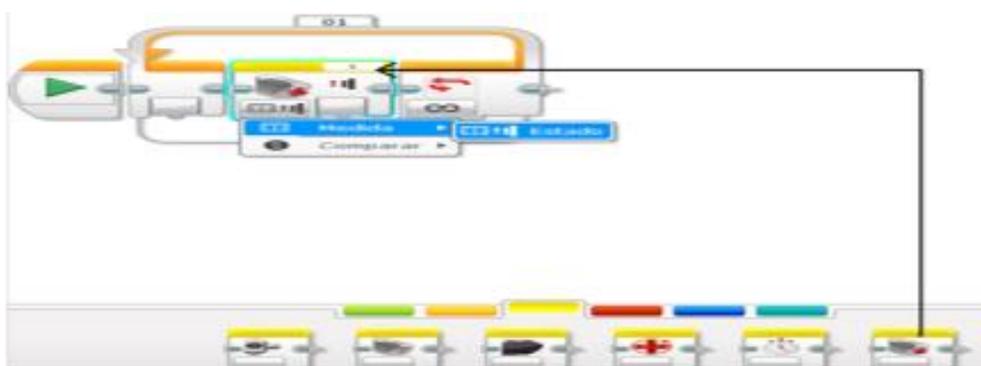
Colocación del bucle en la programación.



Luego el sensor táctil está en la paleta de color amarillo, se arrastra y se coloca dentro del bucle. Cuando se da clic en la parte inferior izquierda del sensor táctil, se abrirá un cuadro con dos opciones (medida-comparar) se guía el cursor en medida y luego dar clic en estado tal como se muestra en la figura 78.

Figura 78.

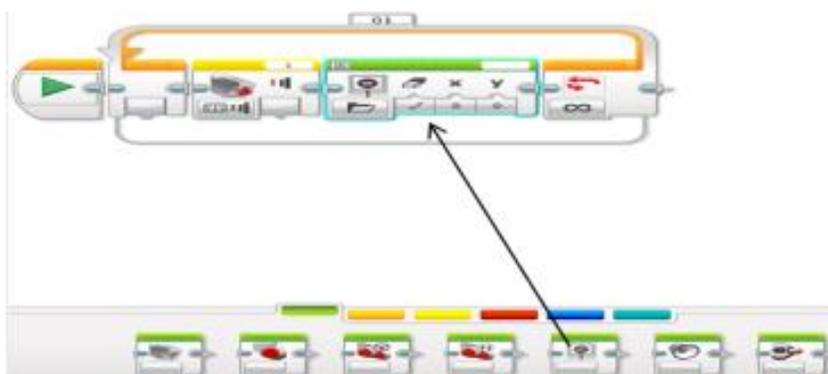
Colocación y configuración del sensor táctil dentro del bucle.



A continuación arrastrar de los componentes de la paleta color verde, el bloque de pantalla como se muestra en la figura 79, que facilitará visualizar las instrucciones que el usuario tendrá que configurar.

Figura 79.

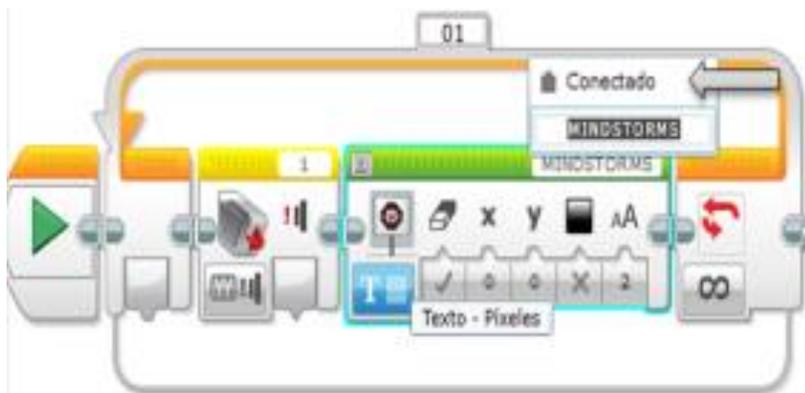
Colocación del bloque de pantalla dentro del bucle.



Luego se configura el bloque de pantalla deslizando el cursor en la parte inferior izquierda del bloque donde se visualizará diferentes opciones: en texto, después clic en píxeles y en la parte superior derecha, arriba donde dice la palabra conectado dar clic, tal como se muestra en la figura 80; para que el usuario pueda visualizar el estado del sensor.

Figura 80.

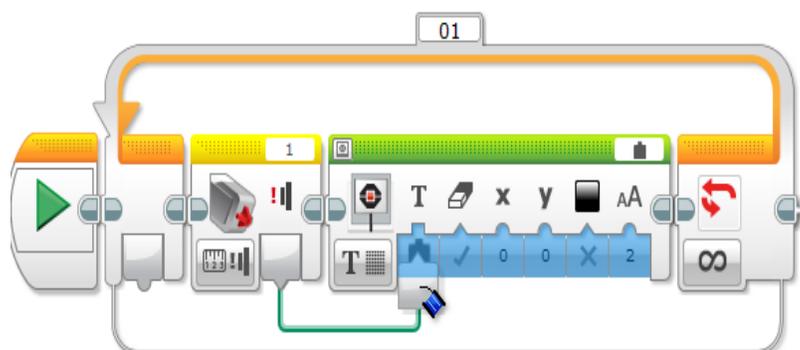
Configuración del bloque de pantalla.



Colocar la parte de salida de estado de medida lógica del sensor hacia la entrada de texto del bloque de pantalla tal como se muestra en la figura 81. Esta acción permitirá visualizar los valores de medida de estado más utilizado en el sensor táctil.

Figura 81.

Configuración para la visualizar los valores lógicos de medida



Terminado la programación se descarga hacia el Bloque físico EV3 llevando el cursor hacia el Panel de Hardware y luego ubicarse en descargar tal como se muestra en la figura 82 y visualizar en la pantalla del panel del hardware los valores lógicos de medida.

Figura 82.

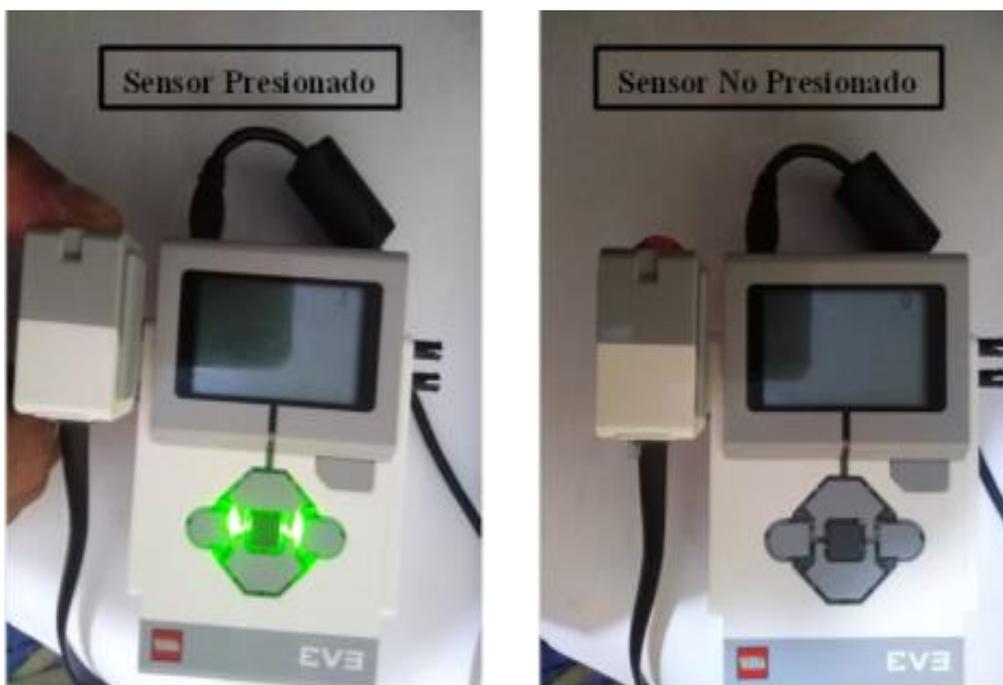
Vista del panel del Hardware para enviar la descarga.



Descargado el programa en el bloque EV3 físico se puede visualizar en la figura 83 los dos estados comúnmente usados por el sensor táctil dando el numero 0 cuando no está presionado y el 1 cuando está presionado, en forma general serán lógicas de falso y verdadero respectivamente.

Figura 83.

Comprobación de estados más utilizados por el sensor táctil.



3.4.3 Programación del sensor Infrarrojo.

Al igual que el Sensor Touch (táctil), al realizar la programación respectiva para el uso adecuado del sensor infrarrojo hay que recalcar que el bloque del sensor infrarrojo no posee la opción calibrar ya que tiene un menú de opciones diferente para su buen funcionamiento.

Las opciones de medida (modo de estado) del sensor tiene tres tipos de selección, estos están registrados en un menú de opciones cada uno de estos se debe definir según la aplicación y valores de estado convenientes. En la siguiente tabla se detalla los modos y usos al que pertenece cada uno.

Tabla 13.

Tipos de estado del sensor Infrarrojo

Modo Proximidad:	En Modo Proximidad, el Sensor Infrarrojo utiliza las ondas de luz reflejadas por un objeto para calcular la distancia entre el sensor y el objeto mencionado. Informa la distancia mediante valores entre 0 (muy cerca) y 100 (muy lejos), no como un número específico en centímetros o pulgadas. El sensor puede detectar objetos que se encuentran a una distancia de hasta 70 cm, según el tamaño y la forma del objeto.
Modo Baliza:	Elija uno de los cuatro canales de la Baliza Infrarroja Remota con el Selector de canal color rojo. El Sensor Infrarrojo detectará una señal de la baliza que coincida con el canal que ha especificado en el programa hasta una distancia aproximada de 200 cm en la dirección a la que está orientado. Una vez realizada la detección, el sensor puede calcular la dirección general (orientación) de la baliza y su distancia (proximidad).
Modo Remoto:	En Modo Remoto, el Sensor Infrarrojo puede detectar qué botón (o combinación de botones) se ha presionado en la Baliza Infrarroja Remota.

Nota: (España, 2015)

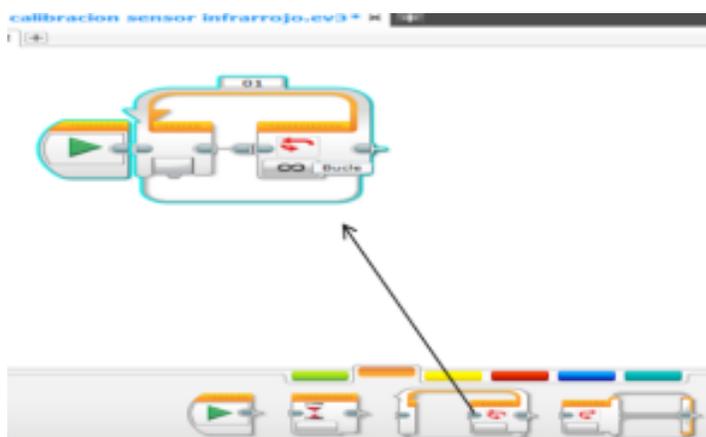
A continuación se detalla paso a paso cómo realizar la programación para el uso óptimo del sensor infrarrojo en su modo de medida de proximidad.

Primero se debe ingresar a un nuevo proyecto, se inicia la programación y en la pantalla del nuevo proyecto se observa el bloque de flujo naranja (iniciar) por defecto, además los bloques que se utilizará en la programación están situados en la parte inferior del software esta consta de paletas identificadas por colores característicos dependiendo su uso.

Como se muestra a continuación en la figura 84, primero utilizar un bucle que está en la paleta de color naranja, basta con arrastrar y colocar el bloque desde la parte inferior hacia el área donde se está realizando la programación para la buena utilización del sensor en su modo proximidad.

Figura 84.

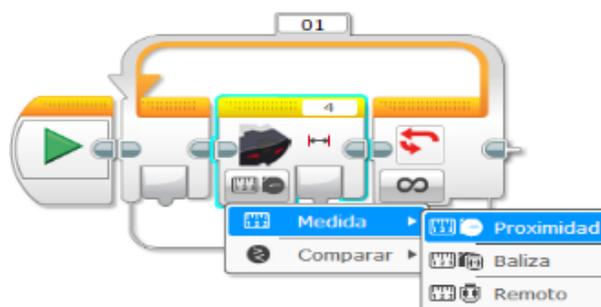
Colocación del bucle en la programación.



Luego el sensor Infrarrojo se encuentra en la paleta de color amarillo, arrastrar y colocar dentro del bucle. Cuando se da clic en la parte inferior izquierda del sensor infrarrojo, se abrirá un cuadro con dos opciones (medida-comparar) guiar el cursor en medida y luego dar clic en proximidad tal como se muestra en la figura 85.

Figura 85.

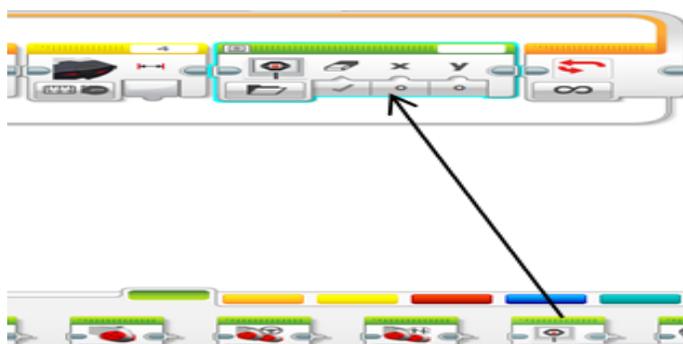
Colocación y configuración del sensor infrarrojo dentro del bucle.



A continuación arrastrar de los componentes de la paleta color verde, el bloque de pantalla como se muestra en la figura 86, que facilitará visualizar las instrucciones que el usuario tendrá que configurar posteriormente.

Figura 86.

Colocación del bloque de pantalla dentro del bucle



Luego se configura el bloque de pantalla deslizando el cursor en la parte inferior izquierda del bloque donde se visualizará diferentes opciones: ir a texto, después clic en pixeles y en la parte superior derecha dar clic en la palabra conectado, tal como se muestra en la figura 87; para que el usuario pueda visualizar el estado de proximidad del sensor.

Figura 87.

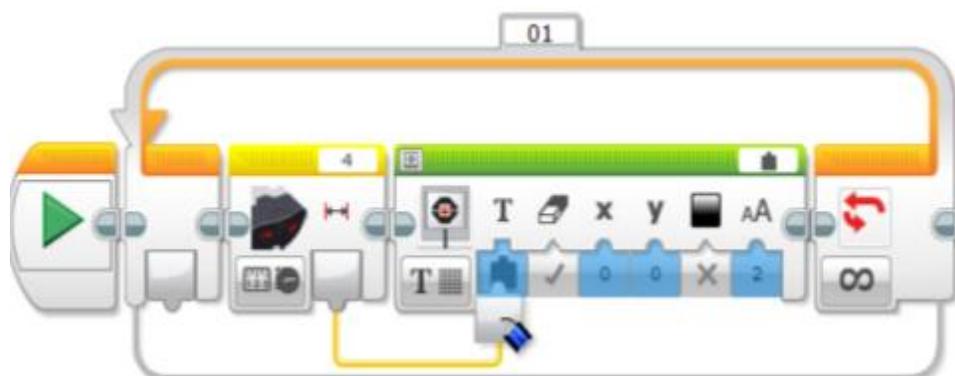
Configuración del bloque de pantalla.



Colocar la salida de estado de medida lógica del sensor hacia la entrada de texto del bloque de pantalla tal como se muestra en la figura 88. Esta acción permitirá visualizar los valores de medida de estado (Proximidad) del sensor infrarrojo en el bloque EV3 físico.

Figura 88.

Configuración para visualizar los valores lógicos de modo medida (Proximidad).



Terminado la programación se descarga hacia el Bloque físico EV3 llevando el cursor hacia el Panel de Hardware y luego ubicar en descargar tal como se muestra en la figura 89 y visualizar en la pantalla del panel del hardware los valores lógicos de medida.

Figura 89.

Vista del panel del Hardware para enviar la descarga.



Descargado el programa en el bloque EV3 físico se puede visualizar en la figura 90, las medias del modo de estado en proximidad comúnmente usado por el sensor infrarrojo dando desde el 0 que es el valor muy cerca y el 100 el valor muy lejos, en forma general serian la distancia de luz reflejado hacia un objeto del sensor.

Figura 90.

Comprobación de estado en modo proximidad por el sensor infrarrojo.

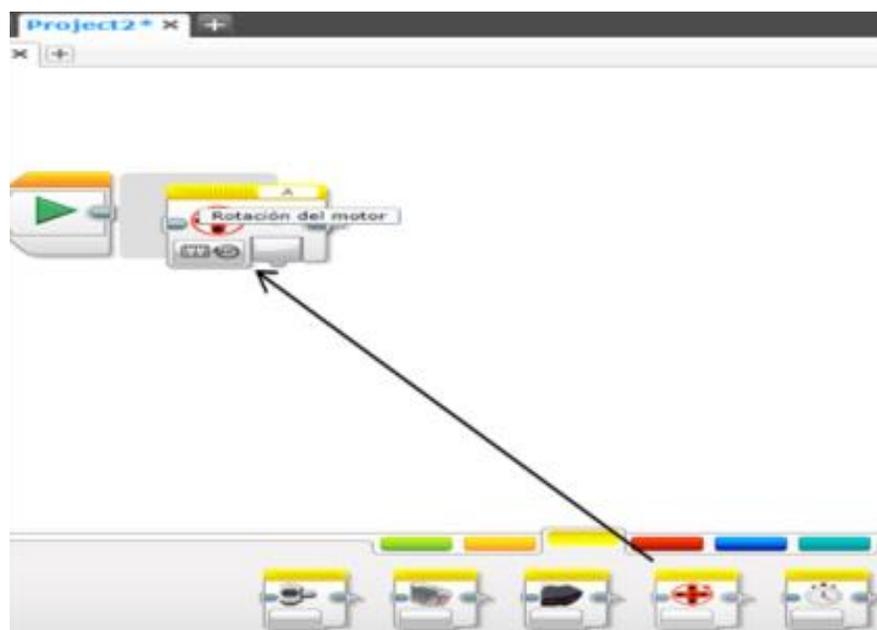


3.4.4 Programación para uso en línea recta de los motores grandes.

Para la utilización de cada motor hacer similar al sensor infrarrojo, realizando la programación respectiva para el uso adecuado de los dos motores grandes adquiridos en el kit de piezas de LEGO en su versión doméstica. Hay que recalcar que el bloque del motor grande está en el icono de paleta color verde y para comenzar la calibración abrir un nuevo proyecto y se utilizará un bloque amarillo de rotación de motor, basta con arrastrar y colocar el bloque desde la parte inferior al área donde se está realizando la programación para la calibración como se muestra en la figura 91

Figura 91.

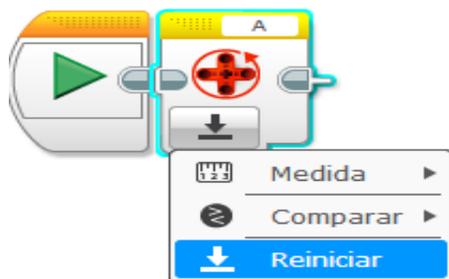
Primeros pasos de programación de los motores grandes.



Como se muestra en la figura 92, en la parte inferior izquierda del bloque de rotación del motor dar clic y se abrirá un cuadro con tres opciones (medida-comparar-reiniciar) guiar el cursor en reiniciar y luego dar clic nuevamente.

Figura 92.

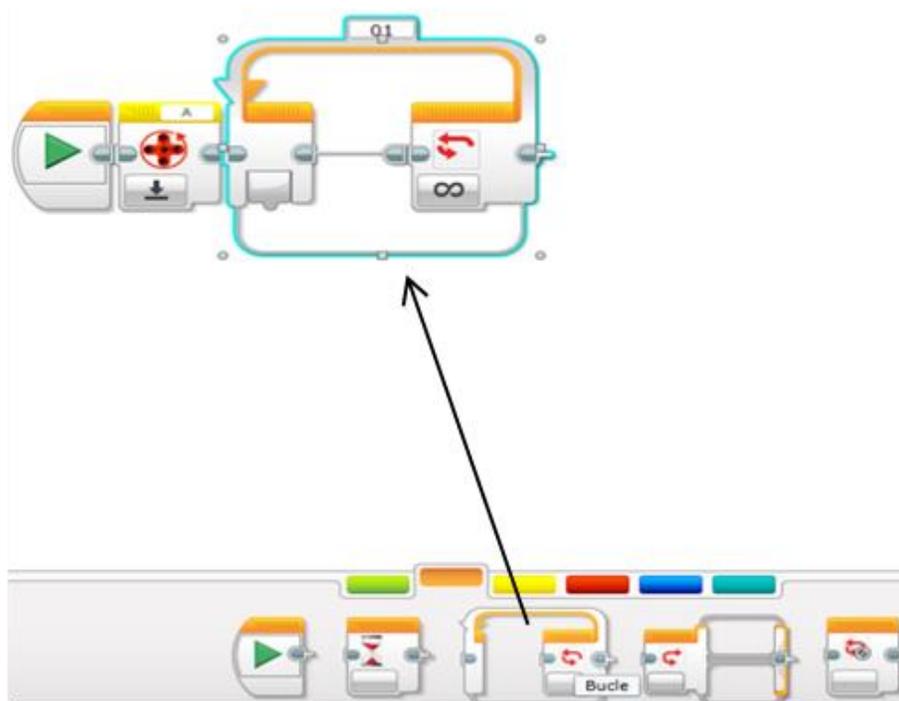
Configuración del bloque de rotación del motor.



Luego utilizar un bucle que está en la paleta de color naranja, basta con arrastrar y colocar el bloque desde la parte inferior, hacia el área donde se está realizando la programación para la calibración del motor tal como se muestra en la figura 93.

Figura 93.

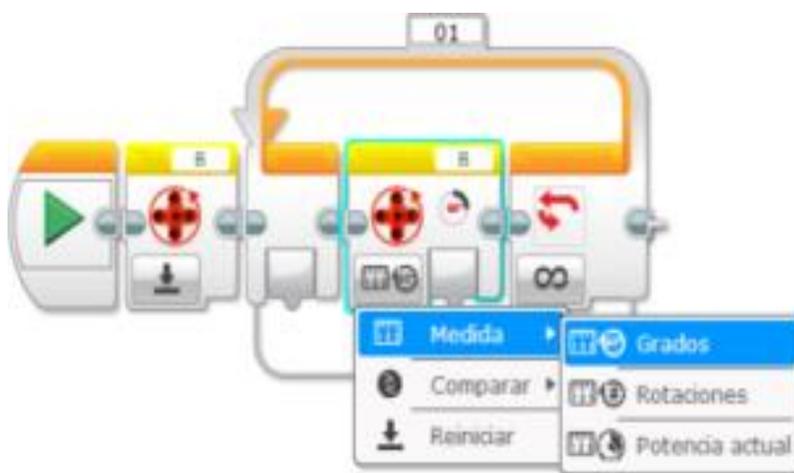
Colocación de un bucle en la programación.



Después arrastrar y colocar dentro del bucle, el bloque de rotación del motor que está en el icono de la paleta color amarillo. Tal como se muestra en la figura 94 al dar clic en la parte inferior izquierda del bloque, se abrirá un cuadro con tres opciones (medida-comparar-reiniciar) guiar el cursor en medida y clic en grados esta medida es la lectura que dará el motor en la calibración.

Figura 94.

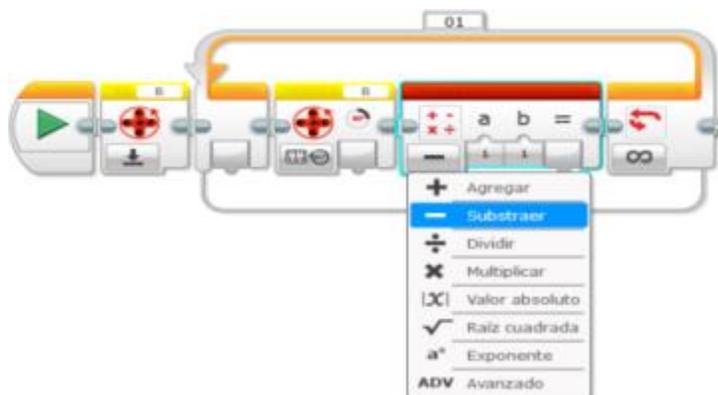
Configuración dentro de bucle del bloque de rotación del motor.



A continuación arrastrar hacia dentro del bucle, el bloque de matemática, como se muestra en la figura 95. La instrucción para este bloque es de resta así que se configura el mismo dando clic en la parte inferior izquierda del bloque; se abrirá un cuadro con opciones matemáticas (agregar – sustraer – dividir – multiplicar - valor absoluto - raíz cuadrada – exponente - avanzado) guiar el cursor en sustraer y dar clic. Hay que recalcar que el usuario tendrá que configurar posteriormente el motor para ver si no se invierte.

Figura 95.

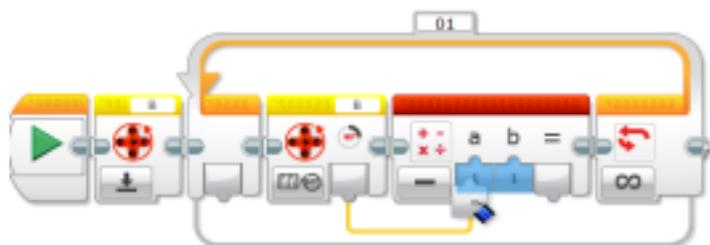
Colocación y configuración del bloque de matemática.



Como se muestra en la figura 71, tomar la salida del bloque de rotación que está dentro de bucle y asignar a la entrada del bloque de matemáticas (a), para calibrar dicha medida substrayendo 1 a los grados del motor, es decir que comprueba si un motor está invertido o es hay más velocidad que otro motor.

Figura 96.

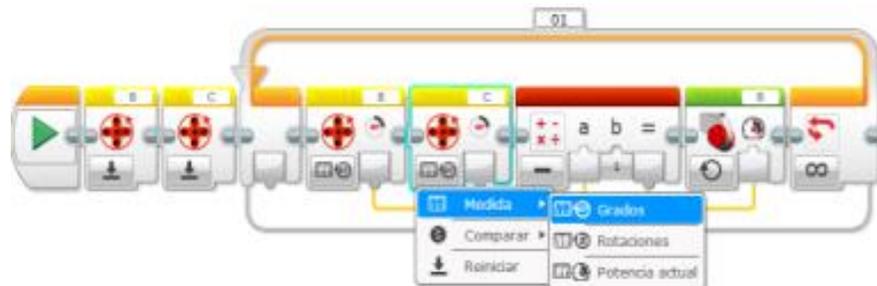
Conexión desde el bloque rotación del motor hacia el bloque matemático



Luego colocar y unir la salida del bloque de matemáticas (=) a un bloque de motor grande dentro del bucle como se muestra en la figura 97. Éste conectado en el puerto de salida B moverá al motor grande físico observando su velocidad óptima y a la vez que no esté invertido.

Figura 99.

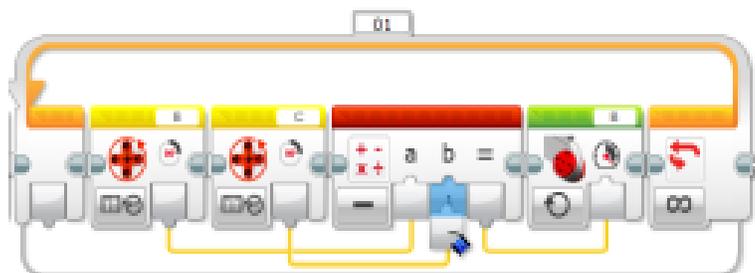
Configuración de los dos nuevos bloques.



Dentro de bucle en el bloque de rotación del motor (puerto C), tomar su salida y asignar a la entrada del bloque de matemáticas (b) tal como se muestra en la figura 100, para comprobar que el primer motor no sea más que el otro, es decir comprobar si un motor está invertido o hay más velocidad que en el otro motor.

Figura 100.

Conexión del bloque de rotación hacia el bloque de matemáticas.



Para autocorregir que el primer motor no sea mayor que el segundo se necesita multiplicarlo por una tasa de corrección (0,5) que aplicará en la parte (b), para que el movimiento de los motores no se haga demasiado brusco. Por lo tanto arrastrar de los componentes de la paleta color roja, el bloque de matemáticas y configurar para multiplicar como se muestra en la figura 101.

Figura 101.

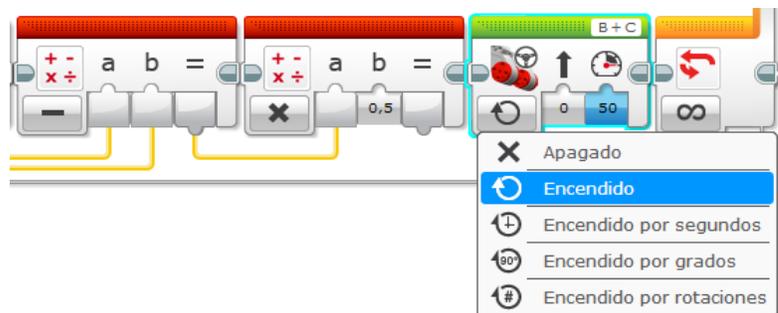
Añadir el bloque matemático y configurar para multiplicar.



Para el mejor rendimiento de los motores grandes colocar el bloque mover la dirección en vez del bloque verde del motor grande y así dejar los motores en un funcionamiento normal de 50 por ciento que sería lo óptimo además ajustar y acondicionar guiando el cursor en la parte inferior izquierda del bloque mover la dirección dar clic y se abrirá un cuadro con cinco opciones (Apagado – Encendido – Encendido por segundos – Encendido por grados - Encendido por rotaciones) guiar el cursor en encendido y luego dar clic nuevamente e inmediatamente se acoplan los parámetros tal como muestra la figura 102.

Figura 102.

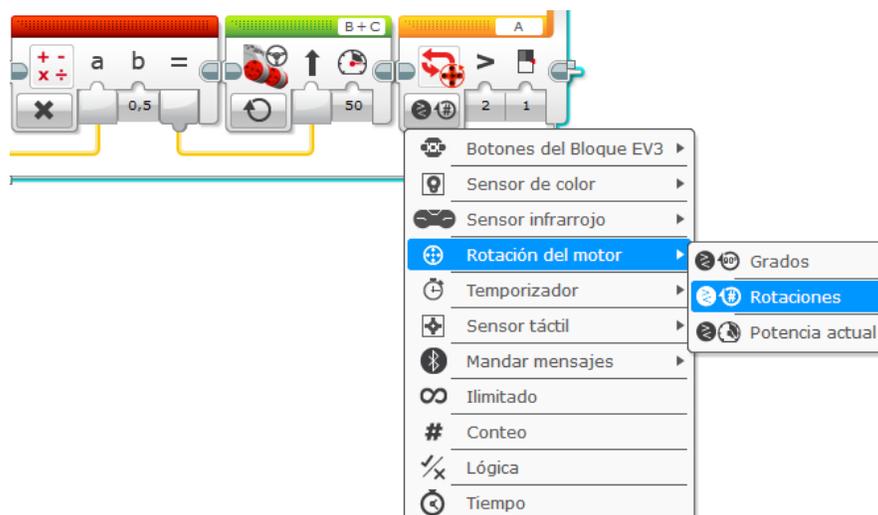
Configuración del bloque verde Mover la Dirección



El resultado de la multiplicación (=) se conecta a la entrada de dirección del bloque verde. Luego ajustar el parámetro del bloque del ciclo como se muestra en la figura 103, acondicionar guiando el cursor en la parte lateral derecha del bucle por defecto está en la opción ilimitado; dar clic y se abrirá un cuadro con diferentes opciones (Botones del bloque EV3 – Sensor de color – Sensor infrarrojo – Rotación del motor – Temporizador – Sensor táctil – Mandar mensajes – Ilimitado – Conteo – Lógica - Tiempo) guiar el cursor en rotación del motor y luego dar clic en rotaciones, esto hará que se detenga tan pronto como la velocidad de cada motor sea mayor o igual al número que se necesita para el uso óptimo de los mismos.

Figura 103.

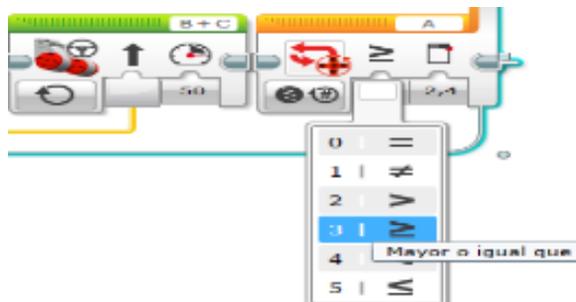
Configuración del bucle para rotación de los motores grandes.



Para finalizar se necesita configurar los parámetros de rotación del bucle, como en la figura 104 dirigir el cursor hacia la parte del comparador del bucle y seleccionar el número 3; este será para darle la comparación de (mayor o igual que) a los motores y colocar un valor límite de 2,4 para la rotación de los motores.

Figura 104.

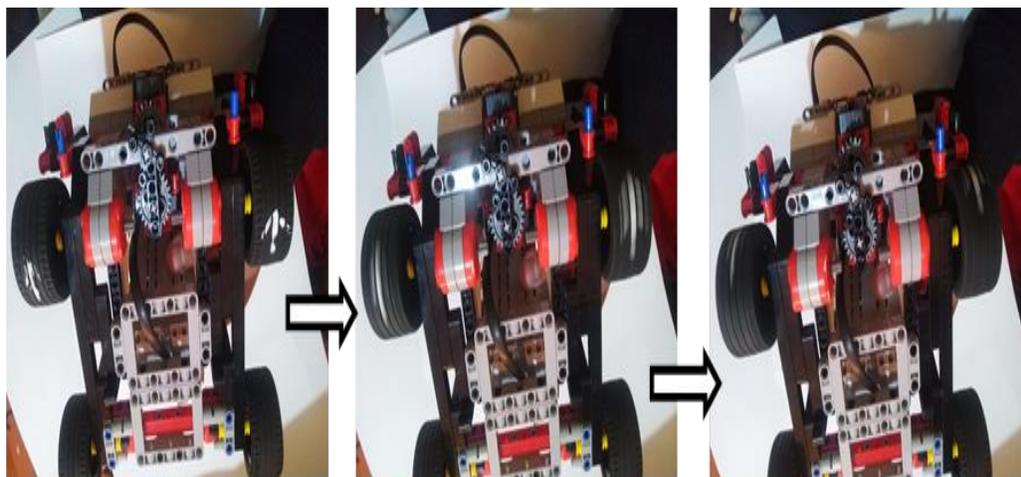
Asignación de parámetros en el bucle.



Descargar el programa hacia el bloque EV3 físico y se puede visualizar en la figura 105 el funcionamiento óptimo de los motores grandes en forma general ambos tendrán la velocidad igual y el lado de rotación similar para los dos motores.

Figura 105.

Funcionamiento de secuencia óptima de rotación de los motores



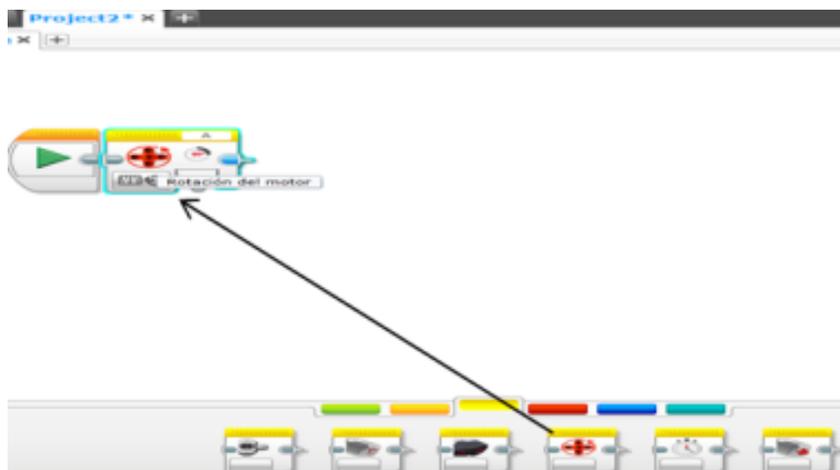
3.4.5 Programación para control de velocidad del motor mediano.

Para realizar la programación respectiva para el uso adecuado del motor mediano, hay que recalcar que la anterior programación es exclusiva sólo para el motor grande y para comenzar la Programación del motor mediano abrir un nuevo proyecto y

se utilizará un bloque amarillo de rotación de motor, basta con arrastrar y colocar el bloque desde la parte inferior al área donde se está realizando la programación para la calibración como se muestra en la figura 106.

Figura 106.

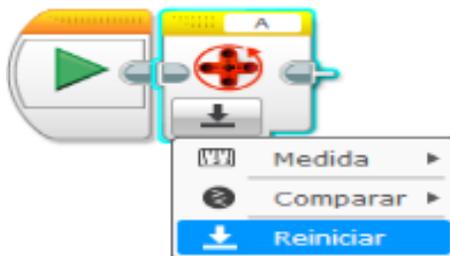
Primeros pasos de Programación del motor mediano.



Como se muestra en la figura 107, en la parte inferior izquierda del bloque de rotación del motor dar clic y se abrirá un cuadro con tres opciones (medida-comparar-reiniciar) guiar el cursor en reiniciar y luego dar clic nuevamente.

Figura 107.

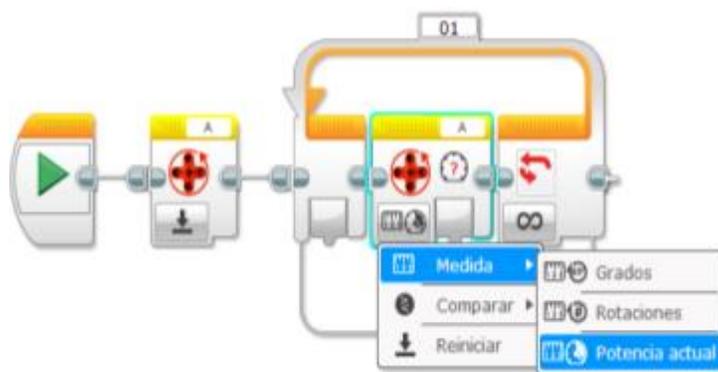
Configuración del bloque de rotación del motor



Luego utilizar un bucle que está en la paleta de color naranja, basta con arrastrar y colocar el bloque desde la parte inferior, hacia el área donde se está realizando la programación del motor mediano y después arrastrar y colocar dentro del bucle, el bloque de rotación del motor que está en el icono de la paleta color amarillo. Tal como se muestra en la figura 108, dar clic en la parte inferior izquierda del bloque, se abrirá un cuadro con tres opciones (medida-comparar-reiniciar) guiar el cursor en medida y clic en potencia actual esta medida dará paso a la medida de lectura que dará el motor en la calibración.

Figura 108.

Colocación del bucle y bloque de rotación con su configuración

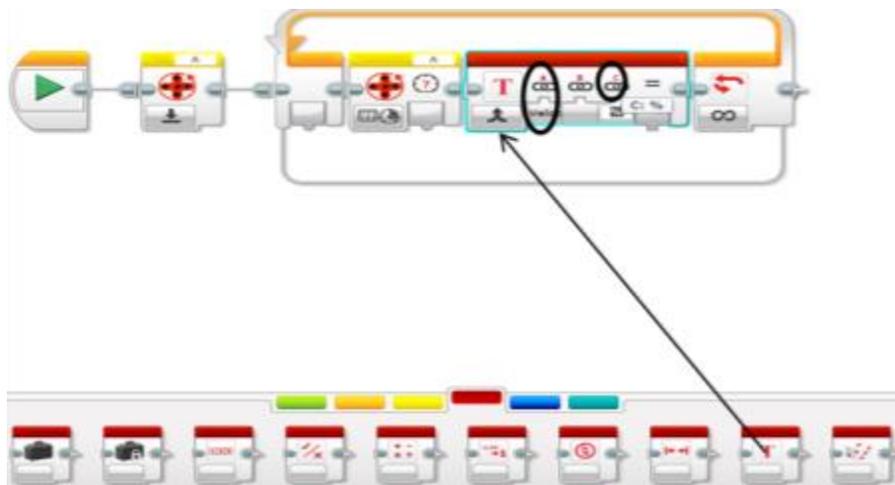


A continuación arrastrar hacia dentro del bucle, el bloque de texto, como se muestra en la figura 109. La instrucción para este bloque es colocar el texto de visualización en la pantalla física del EV3 y configurar el mismo dando el texto (velocidad) en el bloque A y (%) en el bloque C.

Hay que recalcar que el usuario tendrá que configurar posteriormente el motor para ver si no se invierte el movimiento ya que el motor mediano tiene el objetivo de mover objetos con poca carga (Torque).

Figura 109.

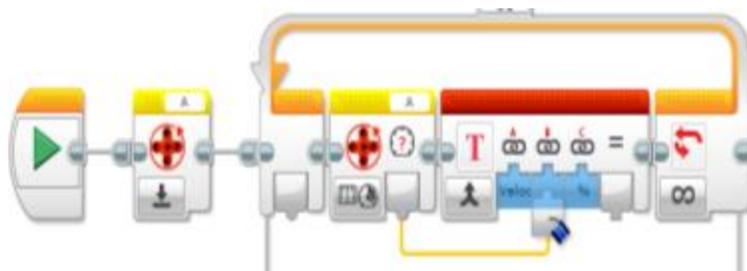
Añadir y configurar el bloque matemáticas



Luego tomar la salida del bloque de rotación figura 110, que está dentro de bucle y asignar a la entrada del bloque de texto (b), para él envié de la medida de la potencia actual del motor comprobando si el mismo está o no está en la velocidad adecuada.

Figura 110.

Conexión desde el bloque de rotación del motor hacia el bloque de texto.



Luego colocar el bloque de pantalla y se lo configura deslizando el cursor en la parte inferior izquierda del bloque donde se visualizará diferentes opciones: ir hacia texto, después clic en pixeles y en la parte superior derecha dar clic en la palabra conectado.

Figura 111.

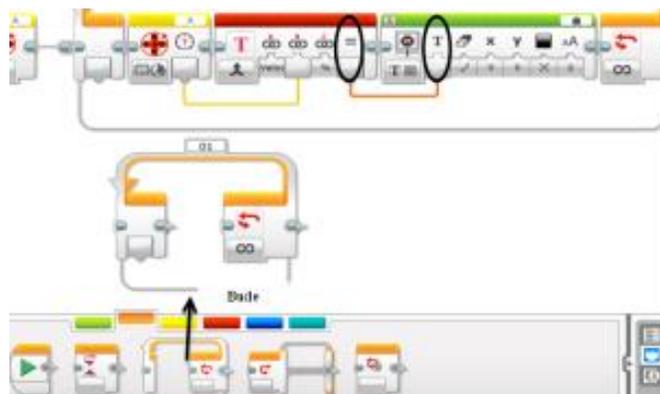
Añadir y configurar el bloque de pantalla



A continuación unir la salida del bloque de texto (=), al bloque de entrada de pantalla (T) dentro del bucle cómo se muestra en la siguiente figura. Además arrastrar otro bucle debajo de los bloques programados este servirá para dar las órdenes de dependencia de estado del motor.

Figura 112.

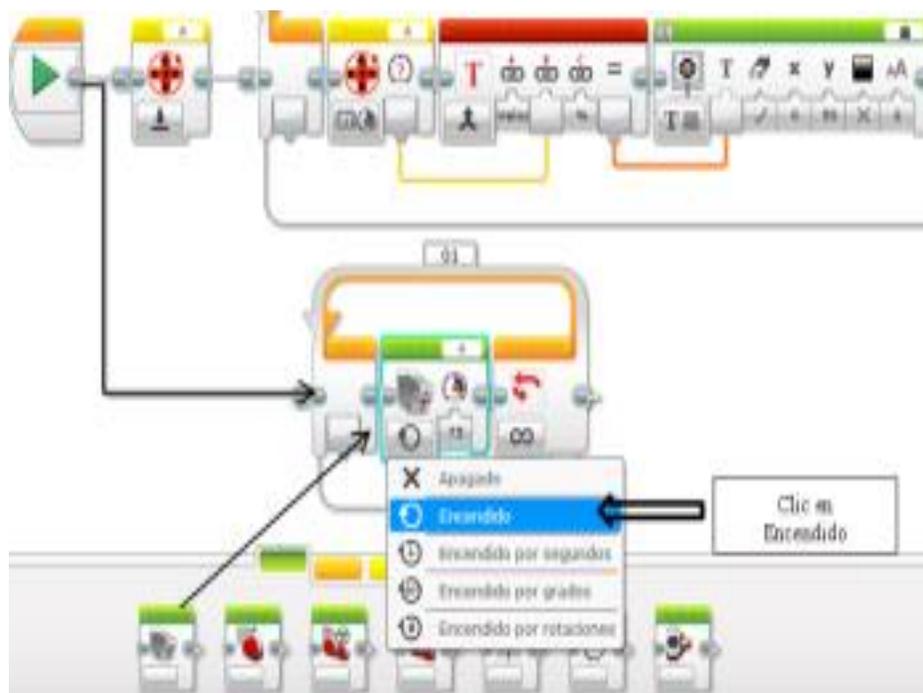
Unión de salida con la entrada de bloques programados y añadir un nuevo bucle.



Añadir desde la salida del bloque a iniciar una unión hacia el nuevo bucle y se arrastra desde el icono de programación color verde un bloque de motor mediano y se configura guiando el cursor en la parte inferior izquierda del bloque del motor mediano dar clic y se abrirá un cuadro con cinco opciones (Apagado – Encendido – Encendido por segundos – Encendido por grados – Encendido por rotaciones) guiar el cursor en encendido y luego dar clic nuevamente; este dará al motor la opción de encendido, es decir, que dé el movimiento dependiendo la calibración en la velocidad.

Figura 113.

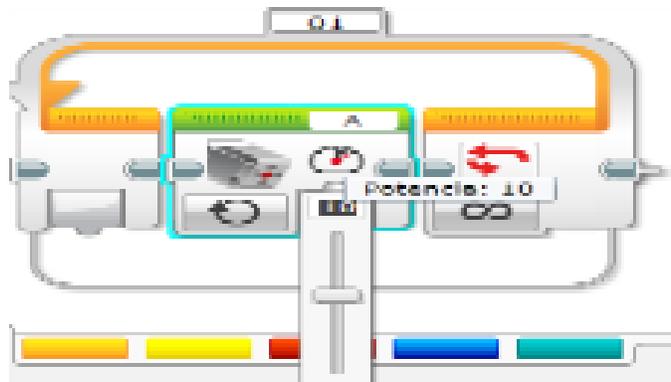
Unión del nuevo bucle al programa y añadir un bloque de motor mediano.



Para finalizar configurar la potencia (velocidad) del bloque motor mediano tal como se muestra en la figura 114, acorde la necesidad del usuario, se configura primero en una potencia de 10% y luego una de 50% descargando ambos al bloque físico.

Figura 114.

Configuración de la potencia del Bloque motor mediano



Luego de haber descargado el programa con las dos cantidades de velocidad hacia el bloque físico EV3, como se muestra en la figura 115, la más ideal y óptima es dejarle en una potencia de velocidad intermedia es decir en un 50%.

Figura 115.

Funcionamiento óptimo de la velocidad del motor mediano.



3.5 Telecontrol del Robot Lego Mindstorms EV3

3.5.1 Telecontrol mediante la Baliza Infrarroja Remota del robot móvil EV3MEG

El control IR remoto funciona de la misma manera que los controles de televisión, básicamente consiste en pulsos emitidos por luz infrarroja que no es visible para el ojo humano. Por ello se necesita primordialmente configurar el ladrillo o brick EV3 manualmente para que los pulsos emitidos por el control IR sean recibidos mediante un efecto fotoeléctrico hacia el mismo. A continuación se detallará paso a paso la configuración del ladrillo EV3 para su posterior control.

Primeramente encender el ladrillo EV3, activado el mismo usar los botones izquierda/derecha en la parte delantera del bloque y trasladarse hacia la tercera opción de la pantalla del brick tal como se muestra en la figura 116.

Figura 116.

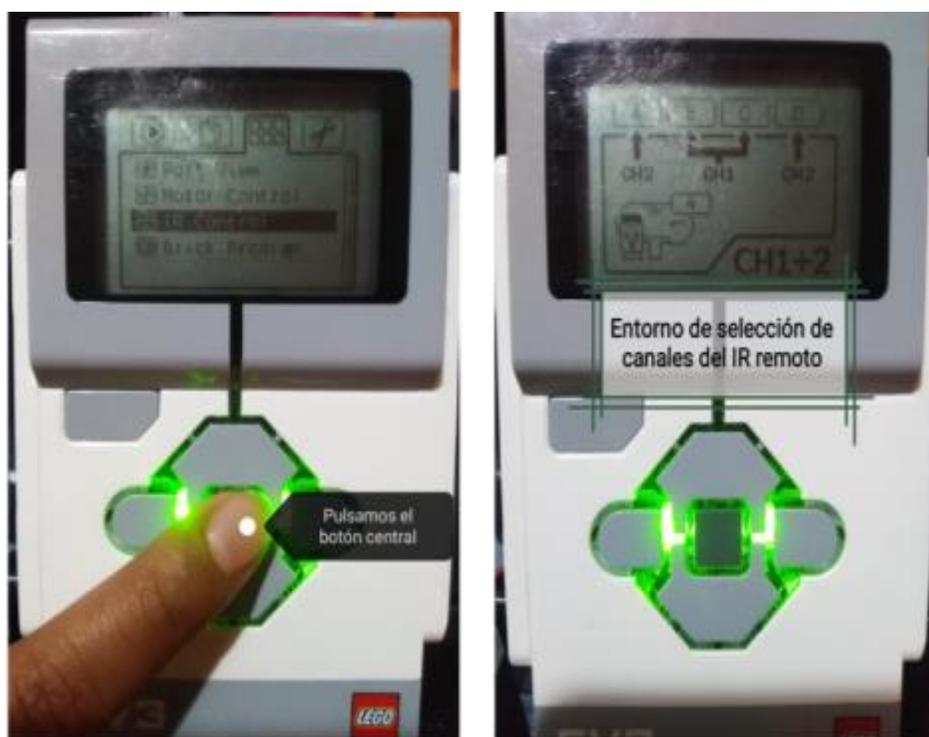
Inicio de configuración del brick



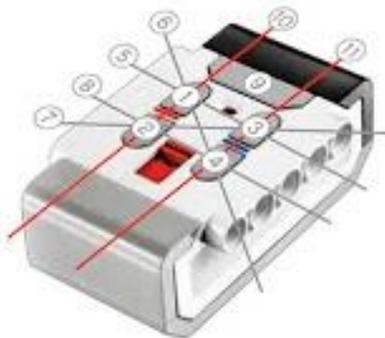
Luego tal como se muestra en la figura 117, usando los botones arriba / abajo de la parte delantera del bloque, se despliega tocando el botón abajo y se traslada hacia IR control; pulsar en el botón central del brick y se debería seleccionar si bien el canal uno, dos, tres o cuatro en el brick se selecciona pulsando el botón central de la parte delantera del ladrillo EV3.

Figura 117.

Selección y Entorno del IR remoto



Para el uso de la baliza remota para el control IR se debe asegurar que el interruptor de mando a distancia esta cambiado al canal apropiado. Para una mejor explicación tal como se muestra en la figura 118, la baliza infrarroja remota se lo puede utilizar como control remoto para el robot EV3MEG.

Figura 118.*Baliza Infrarroja Remota*

Nota: (LEGO®MINDSTORMS®, 2013)

A continuación se detalla en la siguiente tabla las once posiciones o combinaciones de botones posibles de la baliza infrarroja remota.

Tabla 14.*Combinaciones de botones de la baliza infrarroja remota*

0 =	Ningún botón (y el Modo de baliza está desactivado).
1 =	Botón 1.
2 =	Botón 2.
3 =	Botón 3.
4 =	Botón 4.
5 =	Botón 1 y Botón 3.
6 =	Botón 1 y Botón 4.
7 =	Botón 2 y Botón 3.
8 =	Botón 2 y Botón 4.
9 =	Modo de baliza activado.
10 =	Botón 1 y Botón 2.
11 =	Botón 3 y Botón 4.

Nota: (LEGO®MINDSTORMS®, 2013)

Para emplear el telecontrol físico por medio de la baliza infrarroja remota asegurarse que el interruptor de mando a distancia está cambiado al canal apropiado. En la figura 119 se utiliza la combinación de botón número 9, la cual activará el modo de baliza luego colocar ya sea en la combinación de botones número 10 o 11 para controlar los motores del EV3 usando el mando a distancia debido a la disposición de los motores en el robot móvil EV3MEG.

Figura 119.

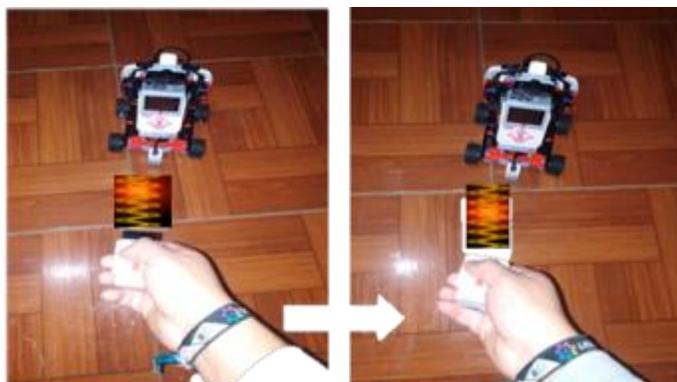
Activar modo baliza y Colocar canal adecuado



Por último verificar que no se tenga ningún obstáculo para la recepción de la luz infrarroja visualizar el funcionamiento del control del robot móvil mediante el modo infrarrojo baliza tal como se muestra en la figura 120.

Figura 120.

Control del robot móvil mediante la baliza infrarroja remota



3.5.2 Telecontrol del robot móvil EV3MEG mediante la creación de una aplicación en APP Inventor 2

Para este ejemplo de telecontrol se mostrará cómo realizar la Aplicación y establecer la comunicación inalámbrica (Bluetooth) del robot Lego Mindstorms EV3 para controlar sus motores desde App Inventor.

Cabe recalcar en la creación de la aplicación hay que tener en claro cuál es el objetivo y el diseño ah alcanzar con esta implementación del telecontrol del robot móvil mediante Bluetooth, para conseguir esto, cómo se muestra en la siguiente tabla el App inventor divide el desarrollo de la aplicación en dos partes:

Tabla 15.

División de desarrollo de la APP

Diseñador.	La ventana Diseñador permite hacer realidad el diseño en mente.
Bloques.	La opción Bloques está designada a la parte de programación de los eventos que ocurren al interactuar con la APP.

Nota: (Acosta, 2016)

Tal como se muestra en la tabla 16 la ventana diseñador está dividida en cuatro grandes apartados.

Tabla 16.

División de apartados.

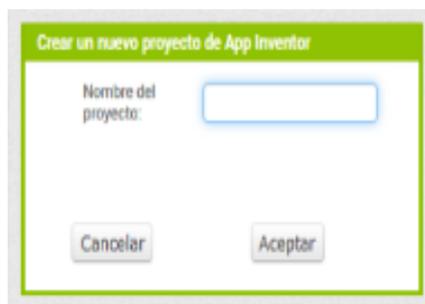
Paleta:	Se encuentra a la izquierda de la pantalla y recoge todos los componentes con los que se puede trabajar a la hora de crear nuestras aplicaciones (botones, sprite, imágenes, sonidos,...)
Visor:	Simula una pantalla de un dispositivo móvil, aquí ir añadiendo los diferentes componentes y dando forma al aspecto que tendrá la App que se había imaginado.
Componentes:	Este apartado recoge en una lista aquellos componentes que finalmente formarán parte de la aplicación, será a partir de esta lista y el último apartado, propiedades, como se terminará nuestro diseño.
Propiedades:	Último gran apartado de esta ventana «diseñador». Seleccionando cada uno de los componentes podrá modificar sus propiedades (modificar textos, cambiar colores, añadir imágenes, entre otros).

Nota: (Acosta, 2016)

Con las aclaraciones anteriores en primer lugar se necesita ingresar en la plataforma de App Inventor 2 (<http://ai2.appinventor.mit.edu/>) tanto si se usa la versión online como la versión descargable, ambas son iguales. Al ingresar a la plataforma online se debe colocar un correo de usuario para ingresar a un nuevo proyecto, luego tal como se muestra en la figura 121, colocar el nombre del nuevo proyecto dentro de la pantalla que se despliega en la misma.

Figura 121.

Entorno de pantalla para la creación del nuevo proyecto.

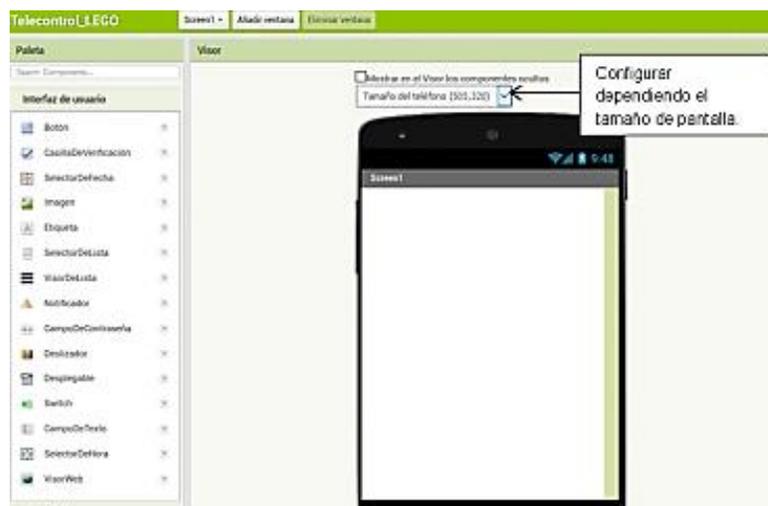


Luego al ingresar al nuevo proyecto se visualizará un entorno de pantalla que es la disposición de nuestra plantilla de visor este estará siempre dentro de una ventana (Screen1) y configurar sus propiedades de texto con el nombre de “Telecontrol LEGO”.

Además este entorno consta de las diferentes opciones de configuración provenientes de App Inventor 2 para la aplicación del telecontrol del robot móvil EV3MEG. Configurar en la opción de visor, el tamaño de pantalla bien sea para teléfono, tablet o monitor, tal como se muestra en la figura 122.

Figura 122.

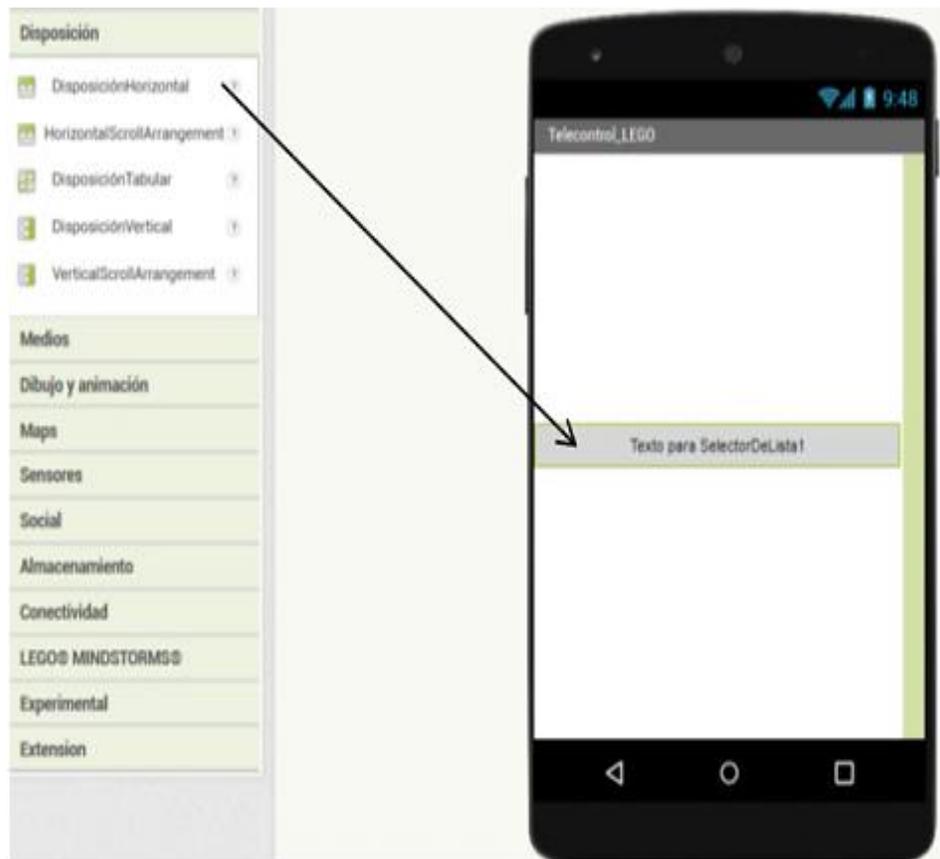
Entorno y Configuración de pantalla.



Dirigir el cursor hacia la parte inferior de la paleta, ir hasta disposición y arrastrar la disposición horizontal hacia la imagen de pantalla del visor tal como se muestra en la figura 123 y además configurar la disposición horizontal en el apartado de propiedades.

Figura 123.

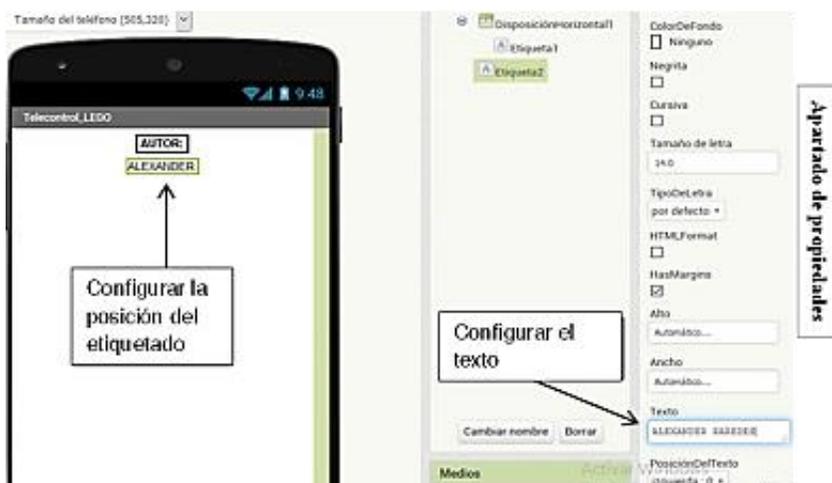
Inicio de creación del diseño de la aplicación.



Luego colocar dos etiquetas que se encuentra en el apartado de paleta, dar clic en interfaz de usuario y la primera se arrastra dentro de la disposición horizontal, está se define en su propiedad de texto para mostrar (Autor). En la segunda definir el nombre del autor tal como se muestra en la siguiente figura.

Figura 124.

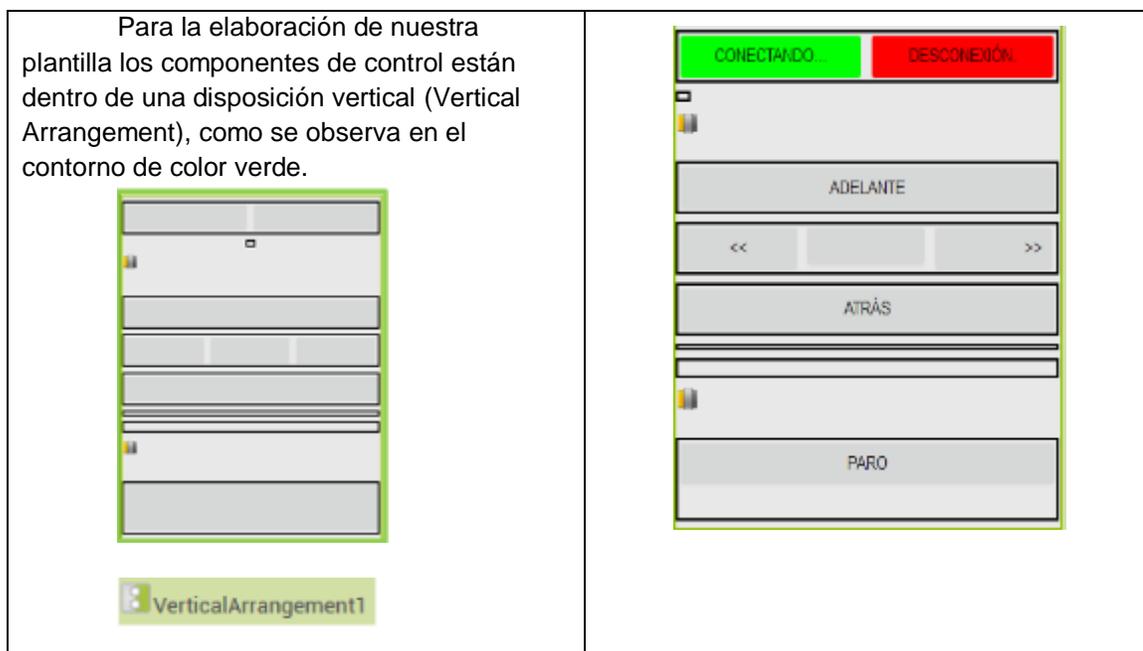
Colocación de dos Etiquetados.

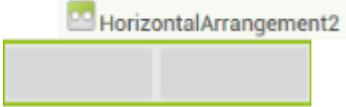
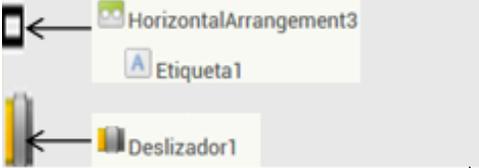
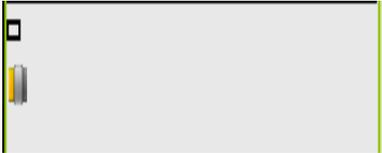
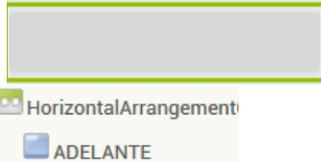


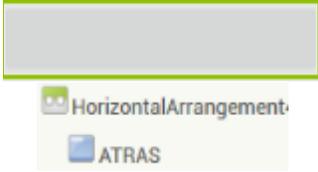
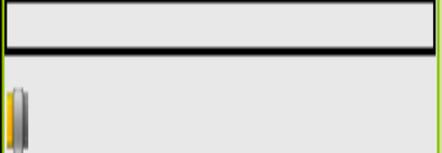
Posteriormente usar una plantilla donde se colocará una disposición vertical para añadir dentro de la misma diferentes componentes que ayudarán al control general del robot y colocarlos según como se muestra en la figura 125.

Figura 125.

Plantilla de componentes para el control.



<p>Luego, usar para los botones de conexión y desconexión una disposición horizontal (Horizontal Arrangement); que es un elemento que permite mostrar de izquierda a derecha los componentes que se van a colocar.</p> 	
<p>Para el ajuste de la potencia de los motores emplear un deslizador (slider) y dentro de una disposición horizontal usar una etiqueta que nos ayudará a la visualización de la cantidad de potencia del deslizador.</p> 	
<p>Para el botón de adelante usar una disposición horizontal, este espacio estará a lo largo del eje horizontal, con un alineamiento vertical centrado.</p> 	
<p>Para mostrar los botones (izquierda y derecha), que tendrá el grupo de componentes para el control del robot, utilizar una disposición horizontal además de un selector de lista que estará entre la mitad de los botones, este último ayudara a seleccionar bien sea el botón hacia la izquierda o la derecha según el uso del robot.</p> 	

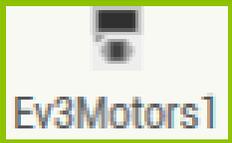
<p>Para el botón de atrás usar una disposición horizontal, este espacio estará a lo largo del eje horizontal, con un alineamiento vertical centrado.</p> 	
<p>Para el ajuste de potencia del motor mediano correspondiente a los brazos del robot usar un deslizador (slider), y dentro de una disposición horizontal usar una etiqueta que nos ayudará a la visualización de la cantidad de potencia del deslizador.</p> 	
<p>Por último para finalizar con el grupo de componentes que se visualizan en el control del robot se usará una disposición horizontal la cual contendrá el botón (Paro) este espacio estará a lo largo del eje horizontal, con un alineamiento vertical centrado.</p> 	

Además cabe recalcar que existen componentes no visibles para realizar la conectividad vía Bluetooth con EV3MEG como el notificador de interfaz de usuario, COMPONENTES LEGO MINDSTORMS, Texto de voz, entre otros. Éstos se ubican debajo de nuestra pantalla de diseño con su seudónimo respectivo, al seleccionar cualquier componente, en el cuadro de propiedades se indicará la información de cada uno de los equipos.

Figura 126.*Componentes no visibles*

A continuación en la siguiente figura se hará una explicación breve de los componentes no visibles utilizados:

Figura 127.*Explicación breve de los elementos no visibles*

<p>Ev3Motors</p> 	<p>Cada vez que se incluya este componente se debe arrastrar hacia el diseño de screen (Pantalla), para que sea incluido como elemento de la aplicación; además es un bloque que proporciona dos interfaces de alto y de bajo nivel para robots LEGO MINDSTORMS.</p> <p>Para el uso de los motores del Kit hay que especificar cada uno, hay excepción cuando se tiene que incluir a todos en uno solo.</p>
<p>Cliente Bluetooth</p> 	<p>Este componente sirve para establecer comunicación y la conexión con los robots tanto EV3 como NXT.</p>
<p>TextoAVoz</p> 	<p>Es un elemento que ayudará a proporcionar voz al texto que se ponga en este caso para indicar si el robot va para arriba, abajo derecha e izquierda.</p>

<p>Ev3UltrasonicSensor</p> 	<p>Este componente sirve para el manejo del sensor ultrasónico pero se acoplo al uso del sensor infrarrojo.</p>
<p>Ev3TouchSensor</p> 	<p>Este componente sirve para el manejo del sensor de contacto.</p>
<p>Ev3ColorSensor</p> 	<p>Este componente sirve para el manejo del sensor de color.</p>

La programación en bloques se realiza de una forma didáctica, esta permite una mejor comprensión de los elementos propios utilizados en el visor de pantalla que es una programación en forma de Scratch y a continuación se detalla los pasos realizados para la programación en bloques:

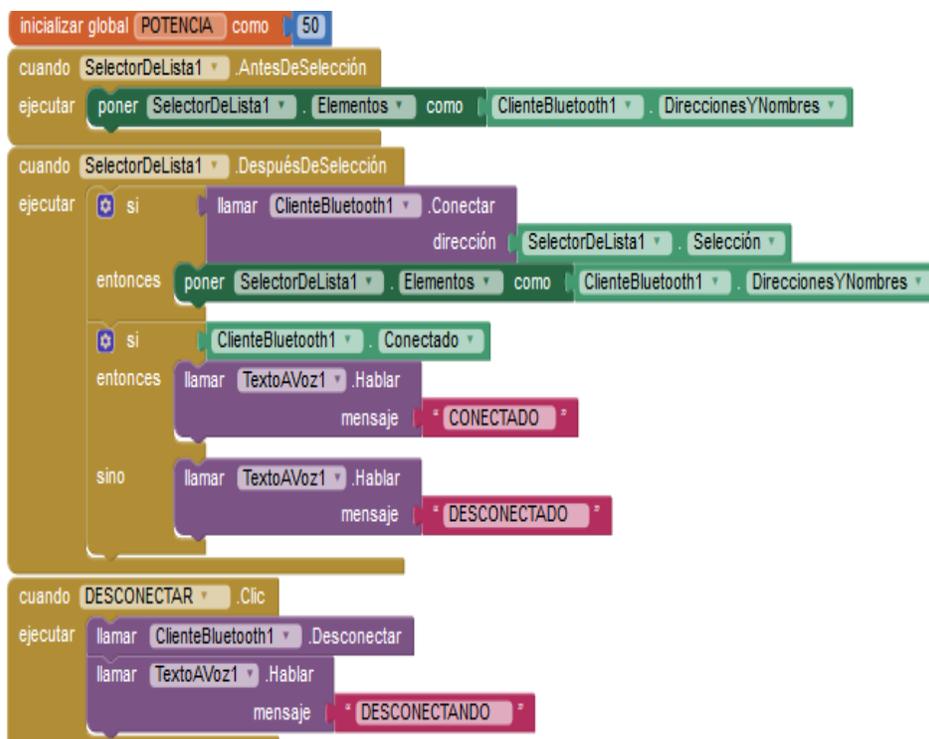
Bloque.Paso1.

A. Inicializar

Inicializar usando una variable (POTENCIA) que representa la potencia de los motores, con valor inicial de 50 además usar un evento de Selector de Lista1, Antes de Selección, se tiene que configurar su elemento Cliente Bluetooth1, Direcciones y Nombres. Después de esto se puede conectar nuestro robot Ev3 emparejando al evento de Selector de Lista1, Después de Selección, Sí se conecta con éxito el Selector de Lista1 se configurará en Incapaz o Desconectado, pero otro botón se configurará en Conectado para que el usuario lo opere.

Figura 128.

Inicialización del Bloque.



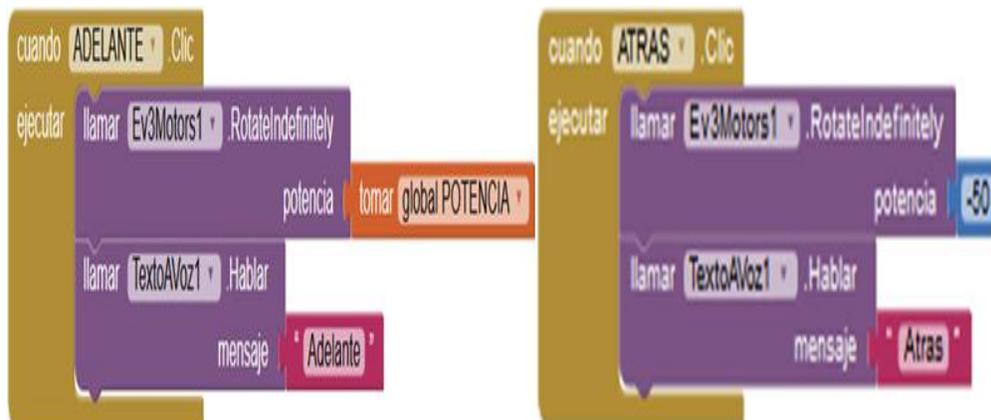
Bloque.Paso2.

B. Botones de Control (Avance y Retroceso)

Con la excepción del botón stop o paro, los dos botones principales que se utiliza llamarán al comando `Ev3Motors1.RotateIndefinitely` cuando se haga clic, con global `POTENCIA`. Tomar el botón adelante como ejemplo y se tiene los siguientes parámetros: variable de potencia tomados de global `POTENCIA` significa que se tomará la misma velocidad para los dos motores y posteriormente llamar `TextoAVoz` que nos dará un mensaje de ir hacia adelante en el caso del botón de atrás este se aplicará la potencia variable por `-50` para hacer que los motores giren en sentido contrario y posteriormente llamar `TextoAVoz` que nos dará un mensaje de ir hacia atrás (Figura 129).

Figura 129.

Bloque de Adelante y Retroceso



Bloque.Paso3.

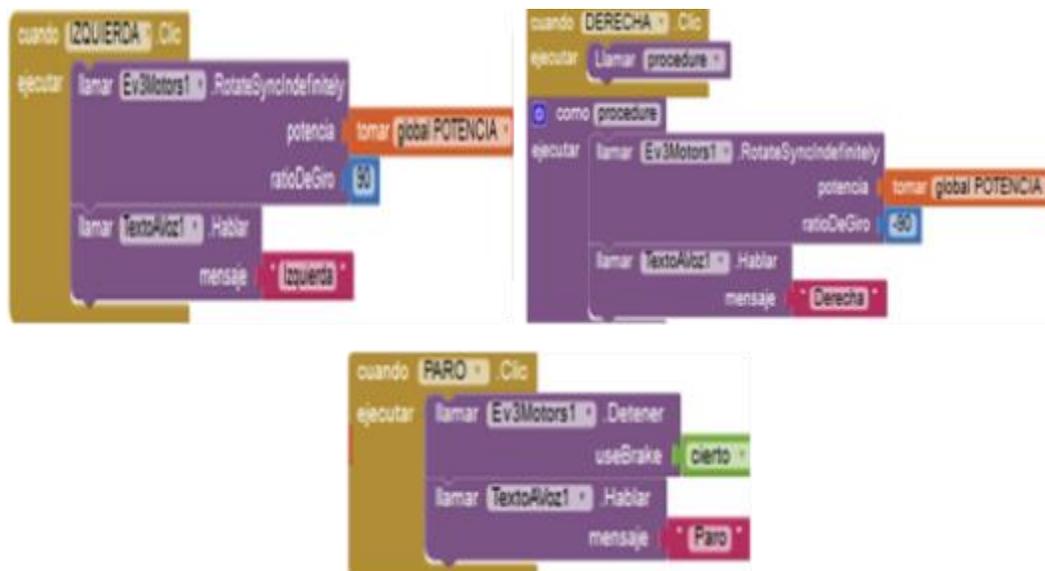
C. Gire a la Izquierda / Derecha y Paro

En cuanto a los botones de izquierda, derecha y paro, llamar al comando `Ev3Motors1.RotateIndefinitely`, cuando se haga clic a su vez en el botón izquierda tendrán un `ratioDeGiro` de 90 y clic en el botón derecha tendrán un `ratioDeGiro` -90 este valores son los más nítidos para hacer un giro y consecutivamente a todos estos se llamarán a `TextoAVoz`.

Finalmente en para el botón Paro para detener el robot, se puede usar el comando `RotateIndefinitely` con 0 de potencia para lograr el mismo efecto. Pero a su vez se llama `Ev3Motors1.Detener` con un `useBreak` en cierto para detener por completo al robot. (Figura 130).

Figura 130.

Bloque de Izquierda, Derecha y Paro.



Bloque.Paso4.

D. Control deslizante para controlar la velocidad del motor.

Para el evento deslizador1.PosiciónCambiada, establecer la variable de potencia en la posición de pulgar del control deslizante, por lo tanto, la potencia del motor también cambia dependiendo la posición del pulgar (0-100) y también poner en Etiqueta1 en texto para tomar la posición del pulgar esto ayudará a visualizar la potencia del motor. (Figura131).

Figura 131.

Bloque de uso del deslizador1.



Bloque.Paso5.

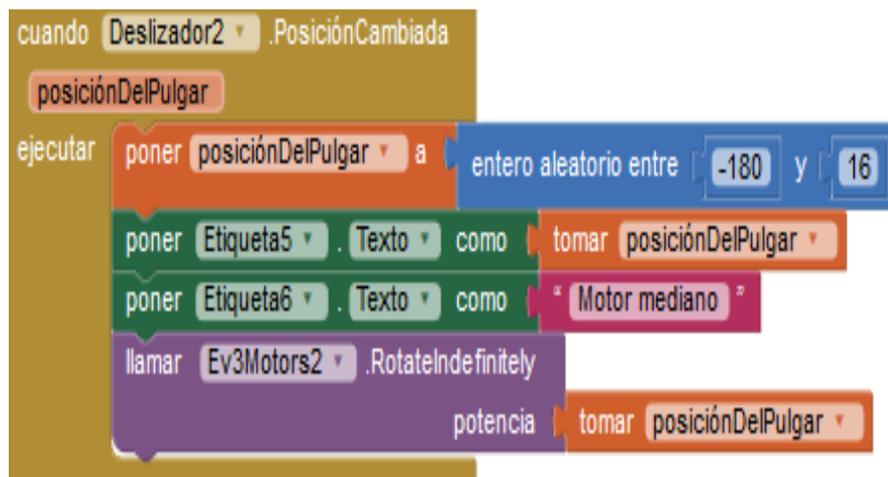
E. Control deslizante para controlar la velocidad del motor mediano.

Para el evento deslizador2.PosiciónCambiada, establecer la variable (entero aleatorio entre) en la posición de pulgar del control deslizante, por lo tanto, la potencia del motor también cambia dependiendo la posición del pulgar (-180, 16) estos son los valores escogidos para el robot por el uso del motor mediano y también poner en Etiqueta5 y 6 textos para tomar la posición del pulgar y el nombre del motor usado correspondientemente, esto ayudará a visualizar la potencia del Ev3Motor2.

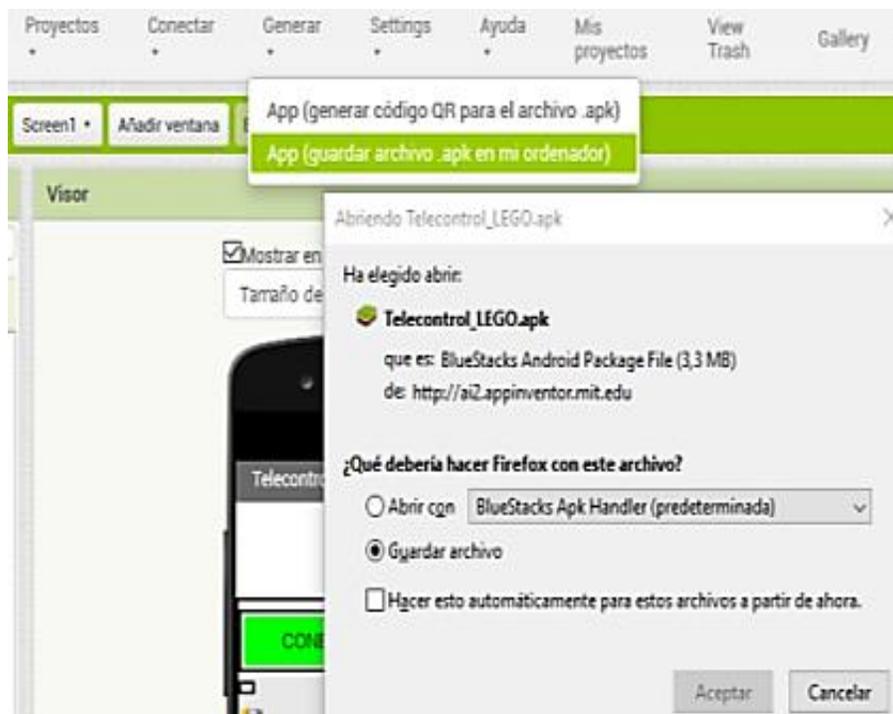
(Figura132).

Figura 132.

Bloque de uso del deslizador2.



Luego de esto se genera un APK en el ordenador desde el MIT APP Inventor (http://ai2.appinventor.mit.edu/?locale=es_ES#5121542631063552) para después descargarlo al teléfono android de su preferencia además cabe recalcar que se puede generar un código QR procedente del entorno de desarrollo de software, en su mayoría es más fácil utilizar el APK generado que el código QR. (Figura133)

Figura 133.*Generar Código APK*

3.5.3 Telecontrol WIFI del robot móvil EV3MEG mediante leJOS JAVA for LEGO MINDSTORMS instalado en la PC.

Los pasos que se debe realizar para el telecontrol Wifi del robot mediante el uso de la PC son los siguientes:

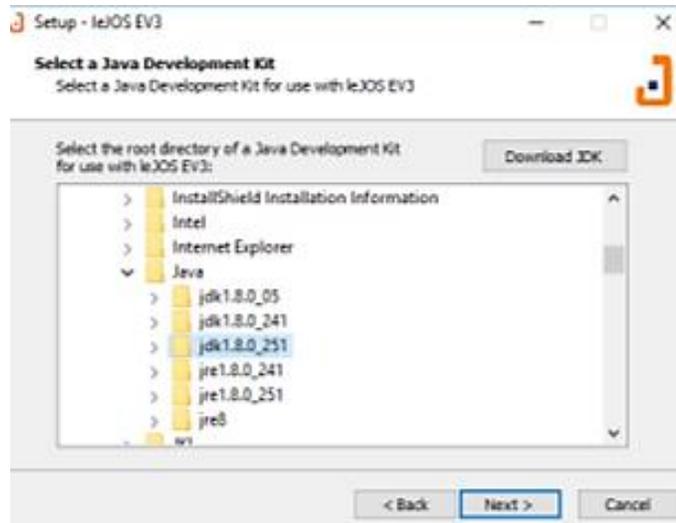
Instalación de leJOS en Windows

Tener en cuenta la última actualización de leJOS que es 0.9.1, donde la pantalla a continuación (Figura134) se ha actualizado para reflejar su versión. Descargar el instalador desde <https://sourceforge.net/projects/ev3.lejos.p/files/> y ejecutarlo. Al momento de estar ejecutándolo debería ver la siguiente pantalla:

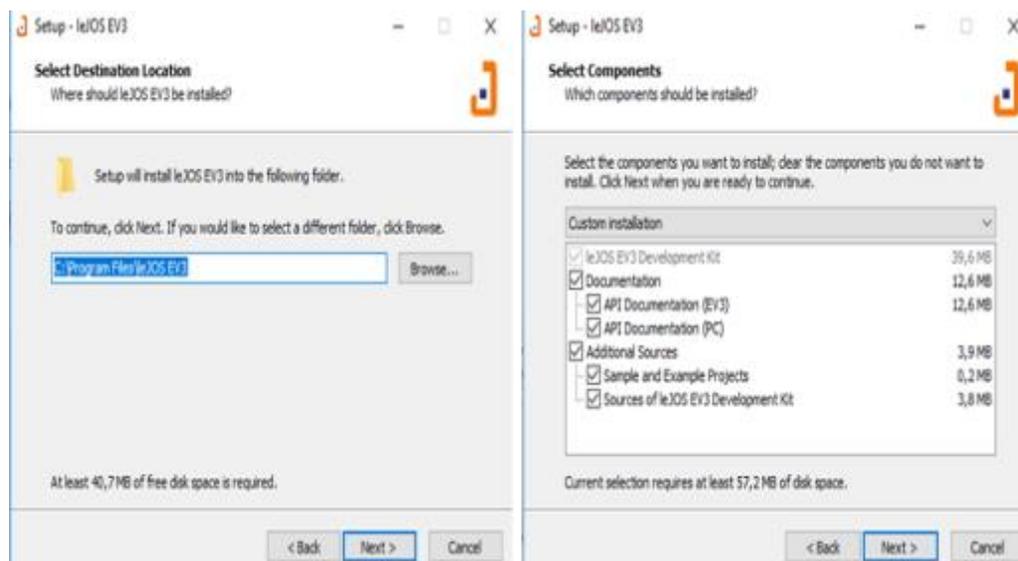
Figura 134.*Instalación Setup-leJOS*

Nota: (Navarro, Cuevas, Pérez, Esquivel, & Oliva, 2018)

Luego de dar clic en siguiente, se debe elegir el JDK que se desee usar en la PC para compilar los programas leJOS se recomienda Java 7 u 8 para esta versión. Sin embargo, para usar esta adaptación lo más fiable es la JDK seleccionada en la Figura 135, esto deberá crear un compacto perfil adecuado, por lo que actualmente es más fácil usar Java 7 en el EV3 y Java 8 en la PC (Köhler, Shaw, & Griffiths, 2016). Si aún no tiene el JDK adecuado se debe dar clic en Download JDK y seguir las instrucciones si está bien hacer clic en siguiente y debería ver:

Figura 135.*Instalación del JDK*

Seleccionar el destino de ubicación para instalar el JDK y luego dar siguiente a menos que se tenga requisitos especiales, elija el destino predestinado y haga clic en siguiente y se debería ver:

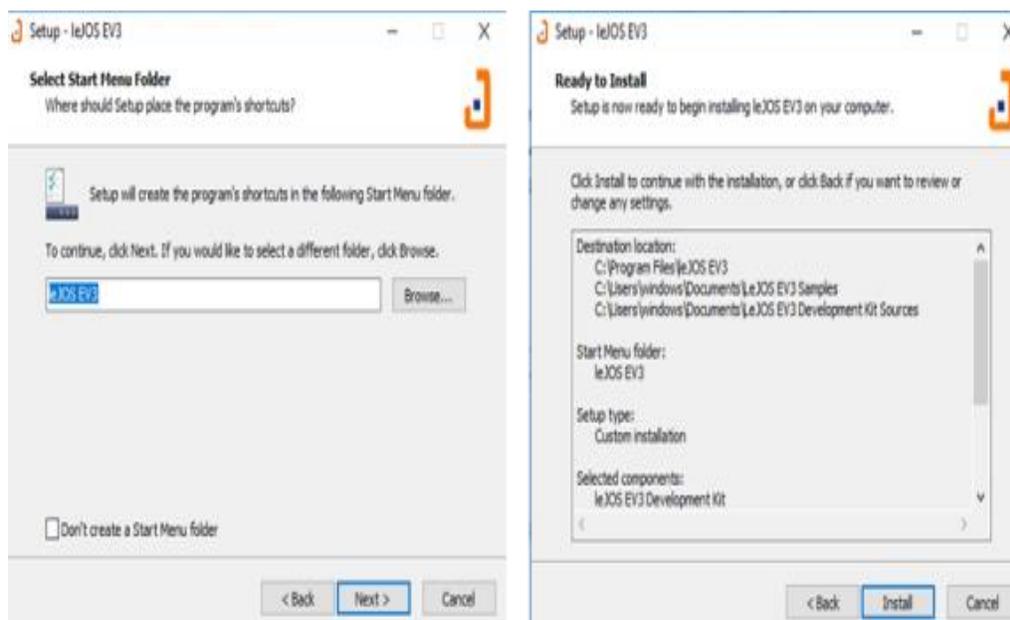
Figura 136.*Selección de Destinatario y Componentes*

Por lo general, es mejor instalar todos los componentes y fuentes adicionales. Cuando haya elegido los elementos opcionales que desea instalar, haga clic en siguiente.

Seleccionar la carpeta de menú de inicio, la ubicación predeterminada y hacer clic en siguiente, además se puede elegir si no se quiere mostrar el menú el inicio dando clic en la parte inferior izquierda en Don't create a start menu folder y posterior a esto se nos mostrará la carpeta que contiene el Install de leJOS e instalar (Köhler, Shaw, & Griffiths, 2016).

Figura 137.

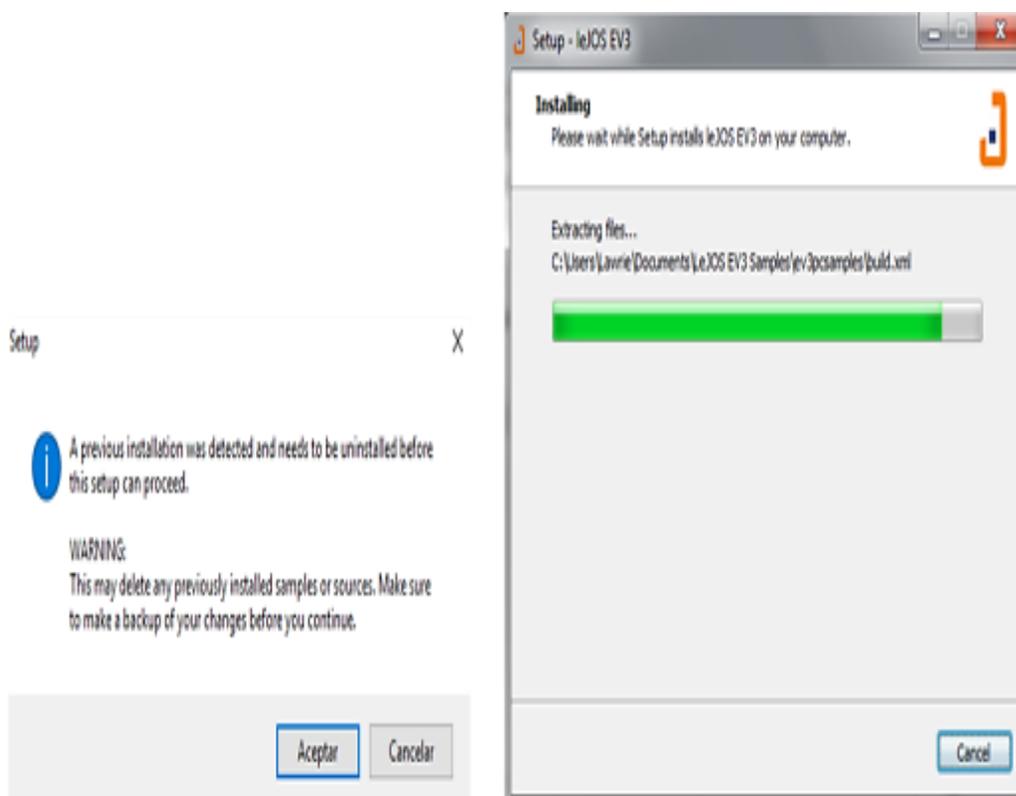
Selección Carpeta de Inicio e Instalación.



Si se tiene una instalación previa de leJOS EV3, se presentará un aviso de Setup de desinstalación dar clic en aceptar (ok) y se desinstalara la instalación anterior. Luego comenzará la nueva instalación mientras la instalación está en proceso se verá lo siguiente:

Figura 138.

Aviso de Setup y proceso de Instalación.

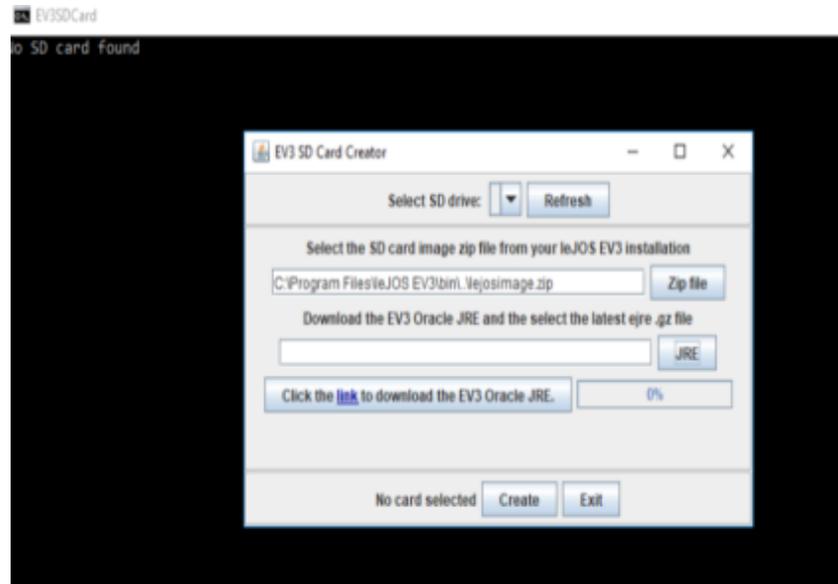


Se nos abrirá una ventana (Figura 139) para crear una tarjeta SD que contenga una versión 0.9.1-alpha de leJOS EV3, a menos que ya tenga una se necesitará una tarjeta SD en blanco de al menos 2 GB y no más de 32 GB. Las tarjetas SDXC no son compatibles con el hardware EV3. Si está reutilizando uno de una versión anterior de leJOS EV3, debe eliminar todos los archivos del mismo y la tarjeta SD debe estar formateada como FAT32, debe estar vacía y debe tener al menos 400 MB de espacio libre.

Inserte su tarjeta SD en su unidad SD, asegúrese de que la casilla de verificación permanezca marcada, y haga clic en Finalizar, y se iniciará la utilidad ev3sdcards (Köhler, Shaw, & Griffiths, 2016).

Figura 139.

Creación del EV3 SD CARD.



El archivo elegido para que la imagen leJOS se escriba en el disco es la instalación que se acaba de crear y debe ser correcta. Si aún no ha descargado Oracle JRE para el EV3, haga clic en el enlace y se lo dirigirá al sitio web de Oracle para descargarlo.

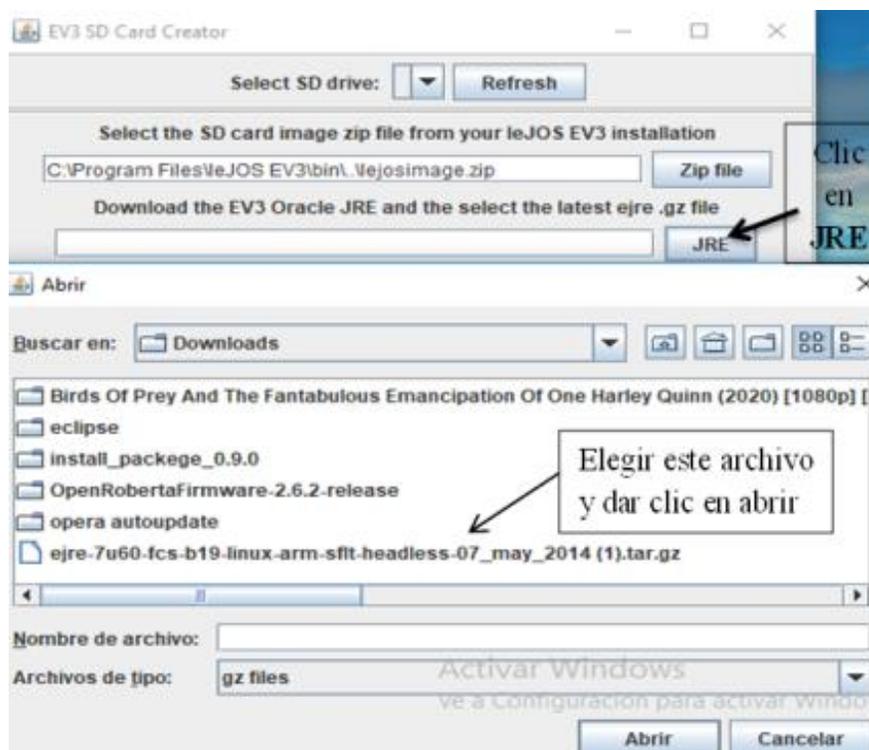
Crear una cuenta, si aún no tiene una. Si por alguna razón el enlace no lo lleva a la página de descarga de Oracle para EV3 / ARM JRE, intente descargarlo desde esta ubicación

<http://www.oracle.com/technetwork/java/embedded/downloads/javase/javaseemeddedev3-1982511.html>. Asegúrese de elegir Java 7 JRE, no Java 8 SDK, a menos que planea crear un perfil compacto Java 8.

Cuando haya descargado el Oracle JRE, haga clic en el botón JRE y elija el archivo este debe estar en su directorio de Descargas, y su nombre comienza ejre y termina .gz (Köhler, Shaw, & Griffiths, 2016).

Figura 140.

Selección de JRE en el directorio de descargas.



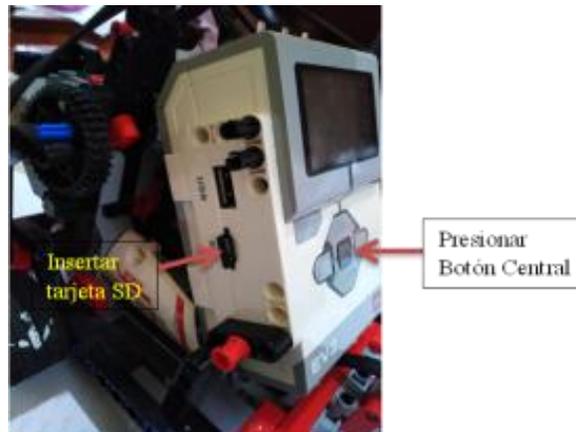
Asegurarse que la tarjeta en la que se encuentra su unidad SD esté seleccionada y su nombre aparezca junto al botón Crear. Si su unidad SD no está en la lista de unidades, intente presionar Actualizar.

Una vez que la unidad y los archivos estén seleccionados, presione Crear y los archivos se escribirán en su tarjeta SD. Cuando se hayan escrito los archivos, verá un mensaje al respecto.

Luego debe expulsar la tarjeta SD de manera segura, insertarla en su ladrillo EV3 y presionar el botón central para encender el EV3 (Figura 141) (Köhler, Shaw, & Griffiths, 2016).

Figura 141.

Inserción de tarjeta SD y encendido del bloque.



Se debería ver el logotipo de leJOS EV3 en su pantalla LCD EV3 (Figura 142), y verá el progreso del formateo y la creación de la imagen de la tarjeta SD. Tardará aproximadamente unos 8 minutos, y al finalizar el proceso mantendrá una breve secuencia de notas y se mostrará el menú leJOS EV3 (Köhler, Shaw, & Griffiths, 2016).

Figura 142.

Secuencia de inicio de leJOS



Luego proceder a configurar el dongle Wifi (**Anexo F**), es recomendado usar el NETGEAR WNA1100, y luego comenzar a desarrollar programas leJOS.

Instalar el complemento Eclipse

Para la realización de programas leJOS es importante utilizar el IDE de Eclipse con el complemento necesario para Eclipse leJOS EV3, para descargar el complemento dirigirse a la página siguiente <http://www.eclipse.org/downloads/>, donde la versión recomendada es la Eclipse IDE para desarrolladores en JAVA (Köhler, Shaw, & Griffiths, 2016).

Figura 143.

Página Principal de Download de Eclipse.



Nota: (Eclipse, s. f.)

Una vez dado clic en descargar, se presenta esta otra pantalla (Figura 144) se guarda el eclipse en las descargas del computador e instalar con su complemento en su PC.

Figura 144.

Pantalla Generada para descargar.

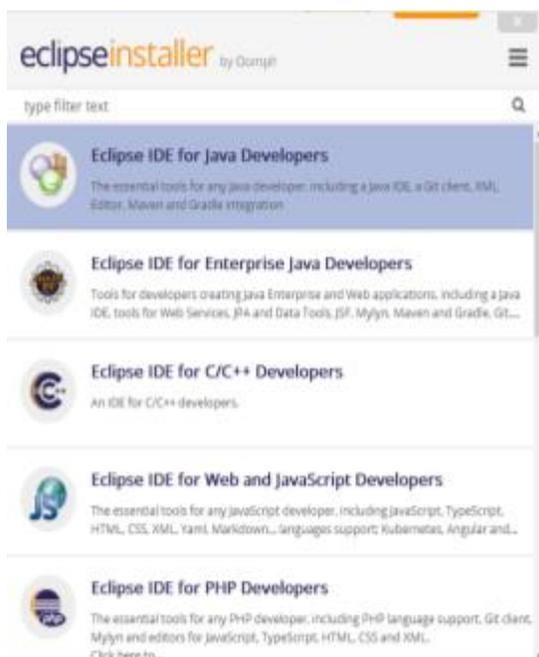


Nota: (Eclipse, s. f.)

Una vez que haya iniciado la descargar como administrador, se deberá instalar dando clic en el complemento de Eclipse IDE for Java Developers (Figura 145).

Figura 145.

Selección del Complemento del Eclipse

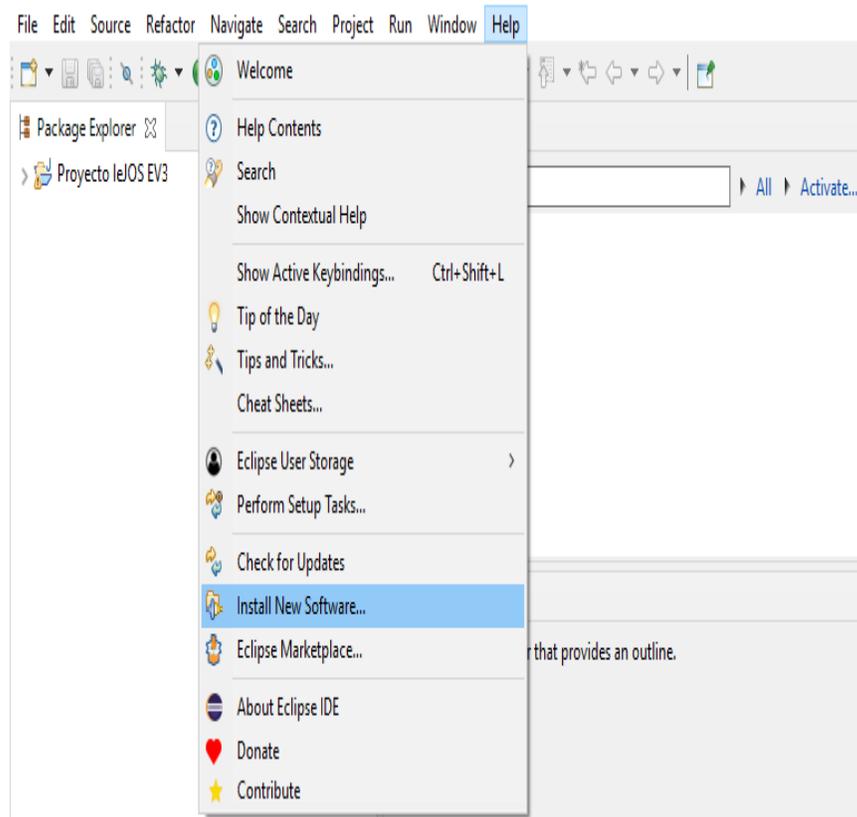


Nota:(Eclipse, s. f.)

Luego de instalar el complemento y al abrirse Eclipse hacer clic en el botón de ayuda y seleccionar “Instalar nuevo software” para añadir el repositorio leJOS EV3.

Figura 146.

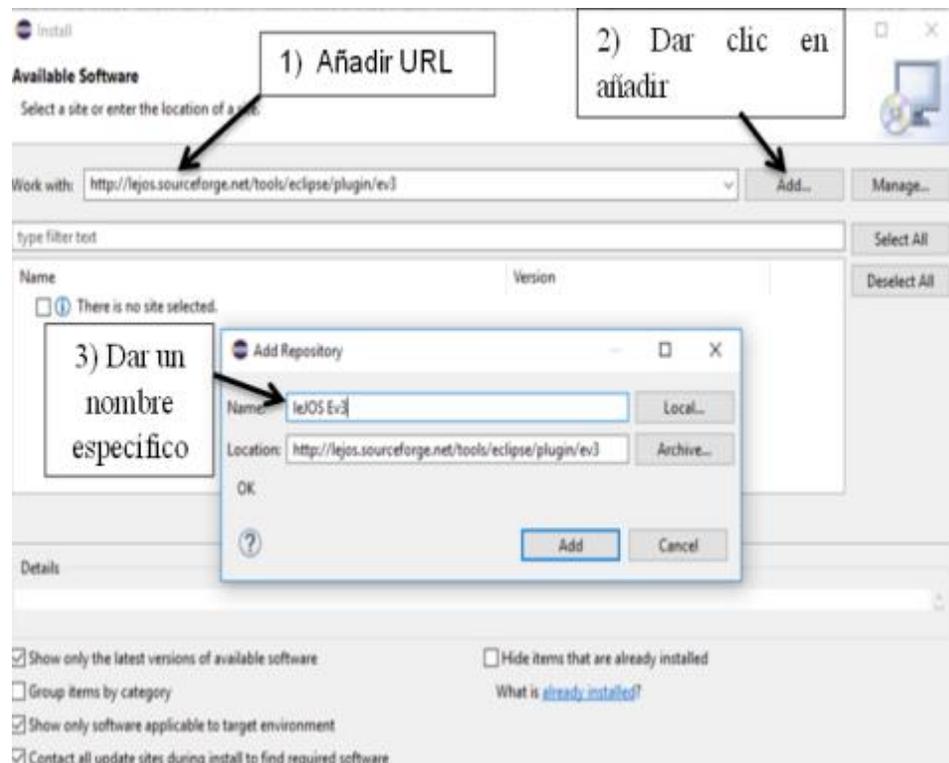
Instalación del nuevo software de leJOS.



Cuando se abra el cuadro de diálogo añadir en “Trabajar con”, la URL del repositorio <http://lejos.sourceforge.net/tools/eclipse/plugin/ev3> y dar clic en añadir luego se aparecerá una nueva carpeta en la que debe dar un nombre adecuado, por ejemplo leJOS Ev3 (Köhler, Shaw, & Griffiths, 2016).

Figura 147.

Configuración para añadir el nuevo software.

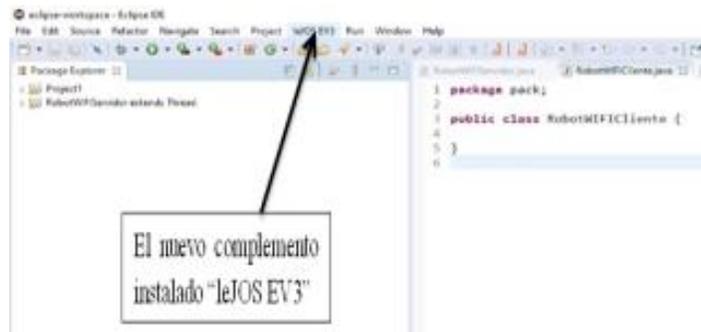


Luego debe hacer clic en la casilla de añadir comenzará la instalación del nuevo software para leJOS. Alternativamente, puede buscar leJOS EV3 en Eclipse Marketplace e instalarlo desde allí.

Posterior al culminar la instalación le pedirá que acepte la licencia, le dirá que el complemento ha sido firmado y luego le pedirá que reinicie Eclipse. Una vez que Eclipse se haya reiniciado, el complemento debería estar disponible y se verá junto a los ítems e íconos de menú.

Figura 148.

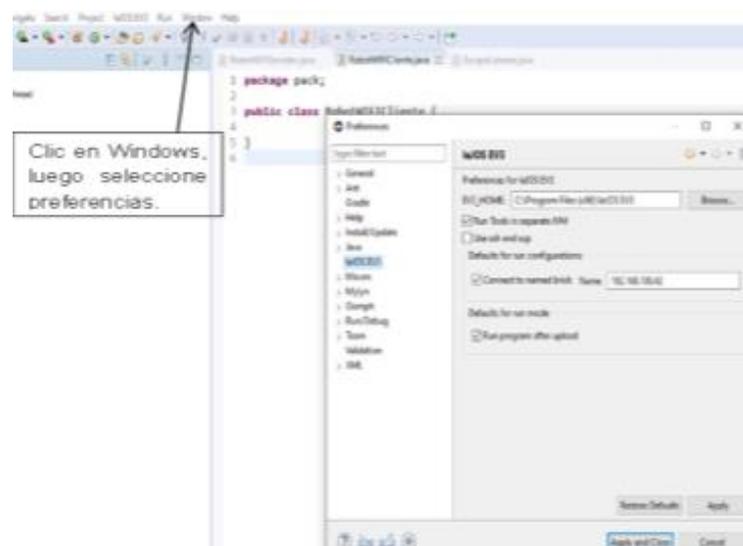
Visualización del nuevo software.



Para el uso de eclipse con leJOS EV3 primero configurar EV3_HOME con las preferencias. Para hacer esto, hacer clic en el menú de Windows y seleccione preferencias y luego leJOS EV3. Se debe configurar EV3_HOME con la red wifi del robot LEGO además dejar "Ejecutar herramientas en una JVM separada" y Usar el código IP generado al conectar el dongle hacia el acceso Wifi y clic en aplicar y el programa se cerrará después de cargar".

Figura 149.

Configuración del EV3_HOME.

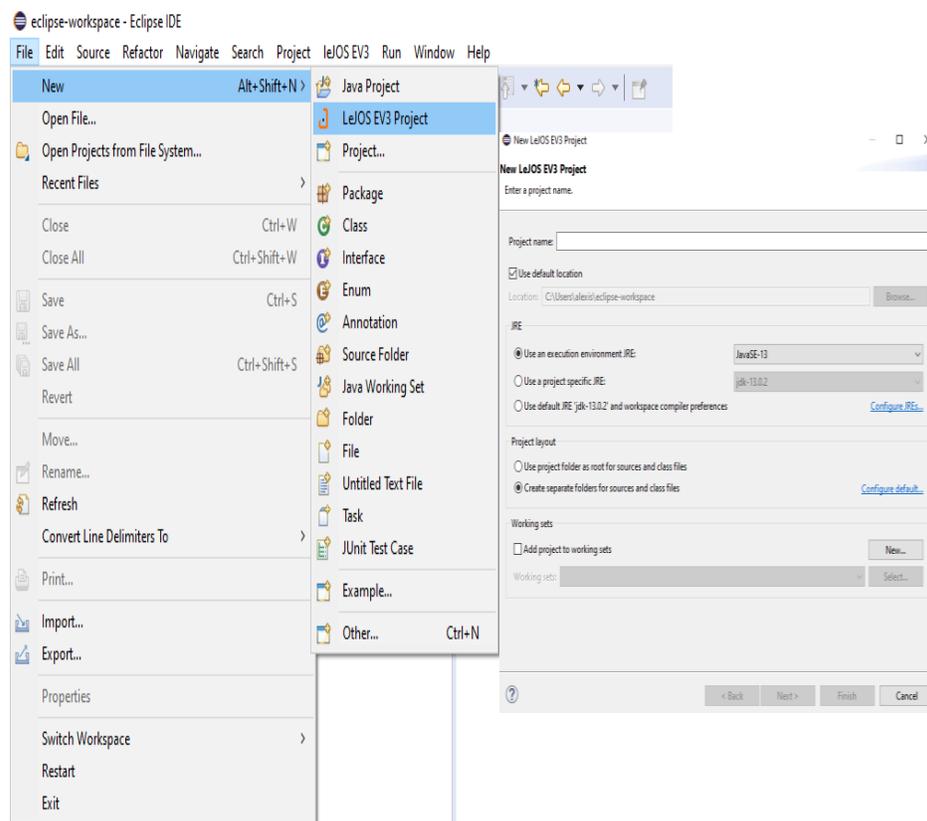


Desarrollo de un Proyecto nuevo con leJOS JAVA ECLIPSE

Para crear un nuevo proyecto leJOS EV3. Para hacer esto, seleccione Archivo--- Nuevo--- Proyecto...en el menú, y luego se visualizará una nueva ventana “New leJOS EV3 Project”.

Figura 150.

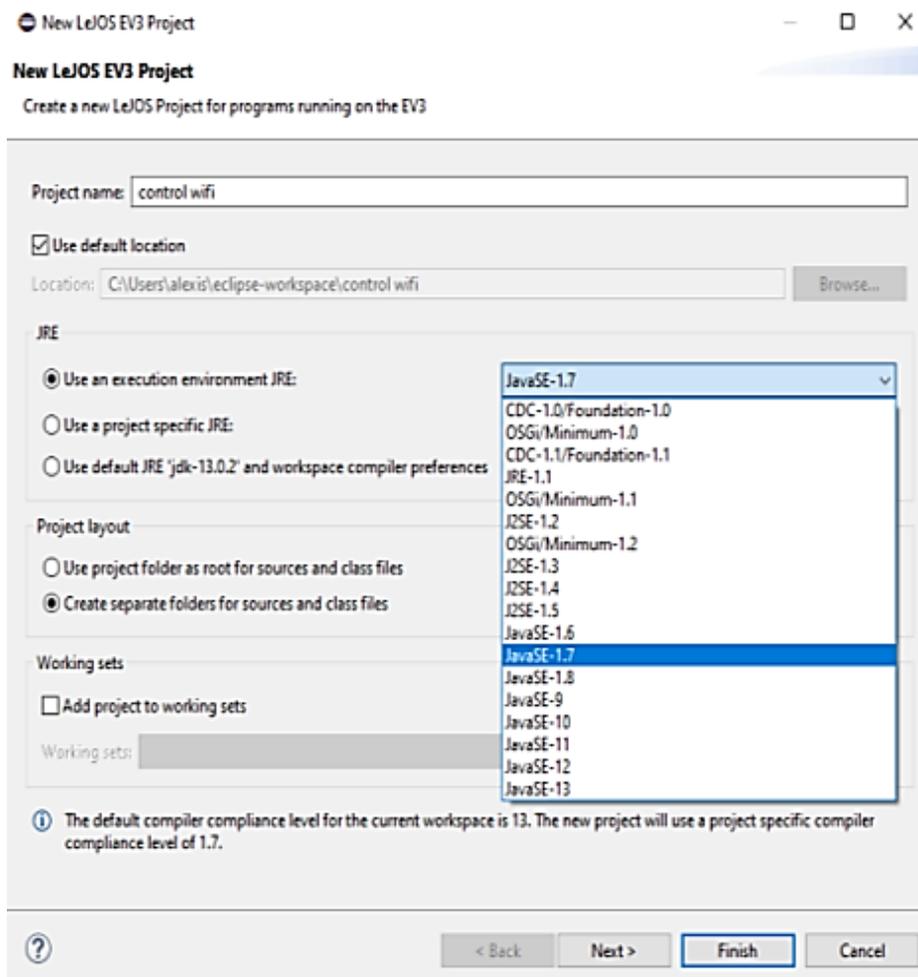
Inicio de Creación de Nuevo Proyecto.



Luego, asignar un nombre a su proyecto y seleccionar el uso de ejecución JRE “JAVASE-1.7” este es el recomendado para el uso de ECLIPSE para posteriormente dar clic en finalizar.

Figura 151.

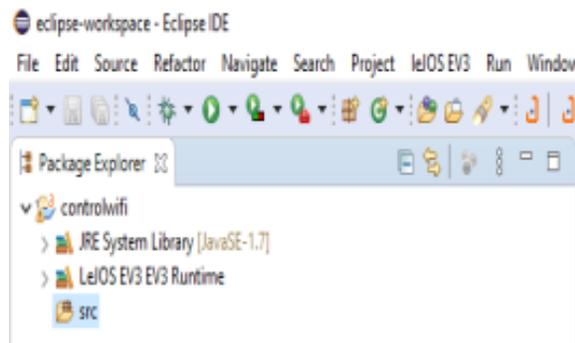
Asignar Nombre del Proyecto y uso de JRE



Se mostrará en el “Package Explorer” el nombre de nuestro nuevo proyecto y también se puede observar el **src** que es donde ira el código de fuente, debajo de éste se tiene el Java Runtime Enviroment “JRE” que es la biblioteca del sistema con (JavaSE-1.7) y posteriormente se tiene el leJOS EV3 Runtime.

Figura 152.

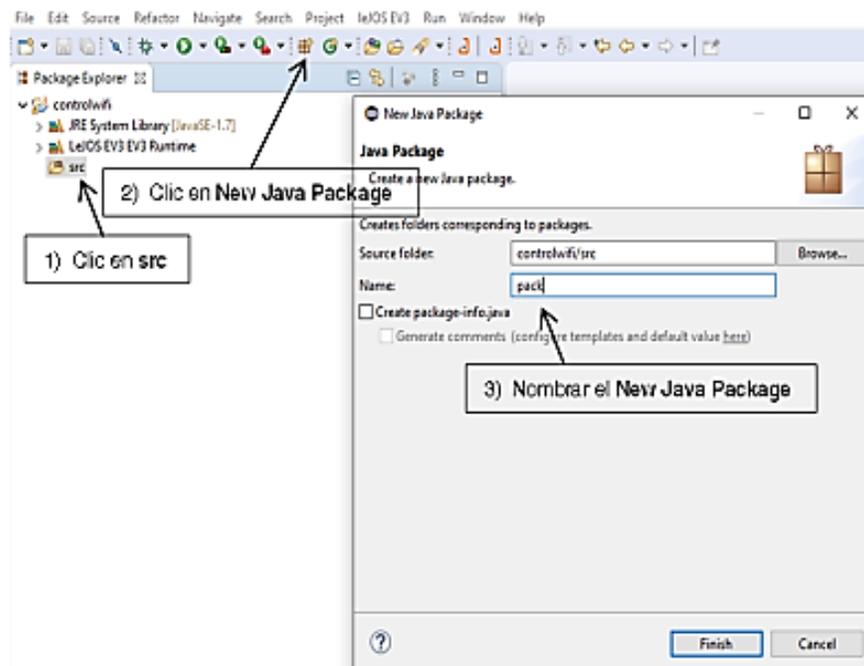
Vista de Contenido del Paquete Explorador.



Hacer clic en **src** y luego llevar el cursor hacia “Nuevo paquete Java” que en Ingles seria New Java Package se nos abrirá una ventana para crear una nueva carpeta correspondiente para el Paquete Java y asignar el nombre de pack y luego clic en finalizar.

Figura 153.

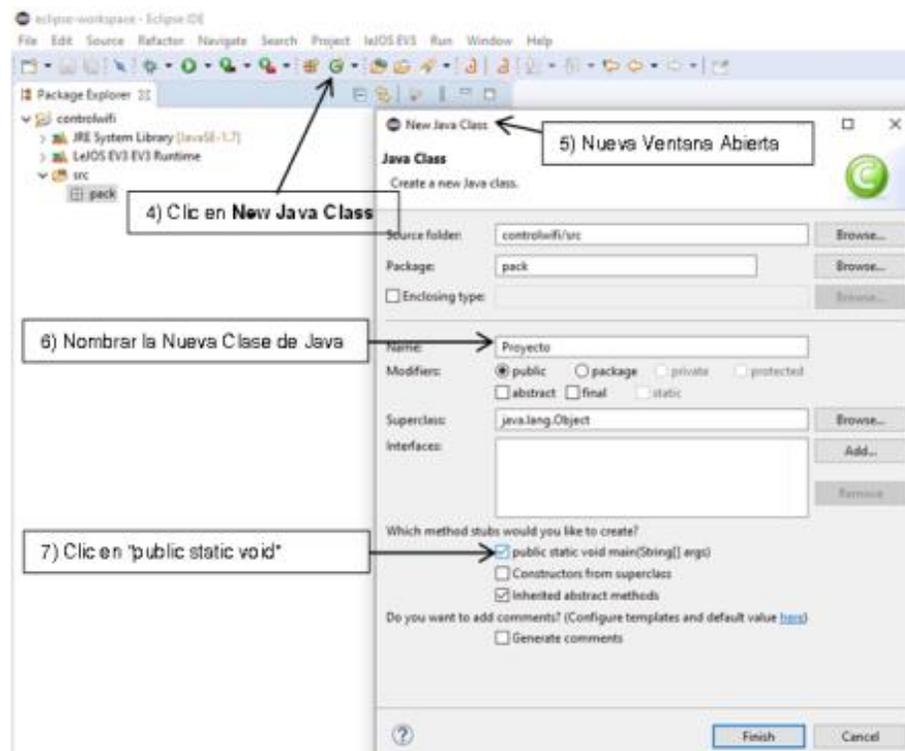
Creación de una nueva carpeta en src.



Creada la nueva carpeta “pack” dirigir el cursor hacia crear una Clase Nueva de Java en ingles “New Java Class”, esta abrirá una nueva ventana para crear una Clase nueva de Java donde se nombrará el nombre (proyecto) y a su vez dar clic en vacío público estático principal “public static void main” donde se señala que será un método público a crear desde esta carpeta y luego clic en finalizar.

Figura 154.

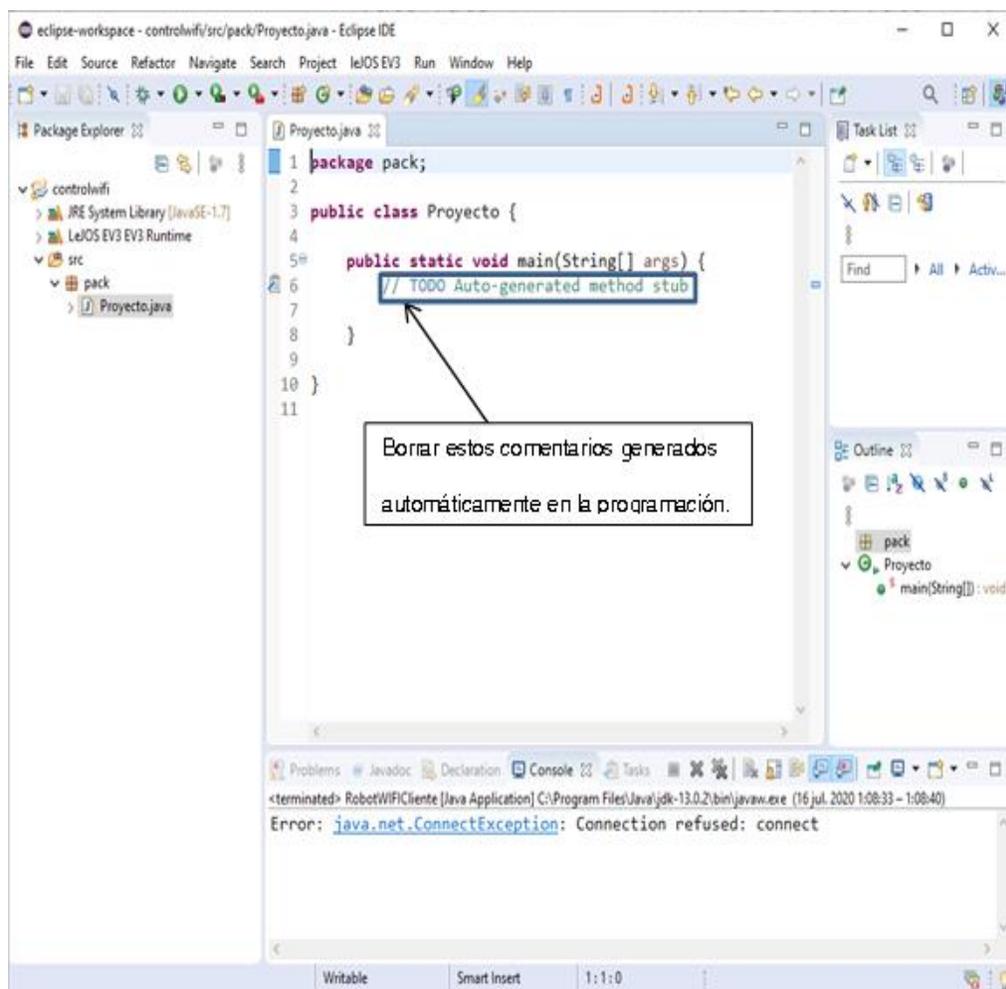
Creación de una Clase Nueva de Java.



Luego de hacer clic en finalizar se nos visualizará la creación de un proyecto básico. En el nuevo Proyecto Básico de Eclipse se genera automáticamente un comentario que por lo general viene con dos líneas laterales inclinadas “//” se borra todo este comentario y se comienza a programar. Es así como se puede realizar de forma básica diferentes tipos de líneas de instrucción para programación en Eclipse.

Figura 155.

Entorno de vista de un Proyecto Básico en Eclipse.

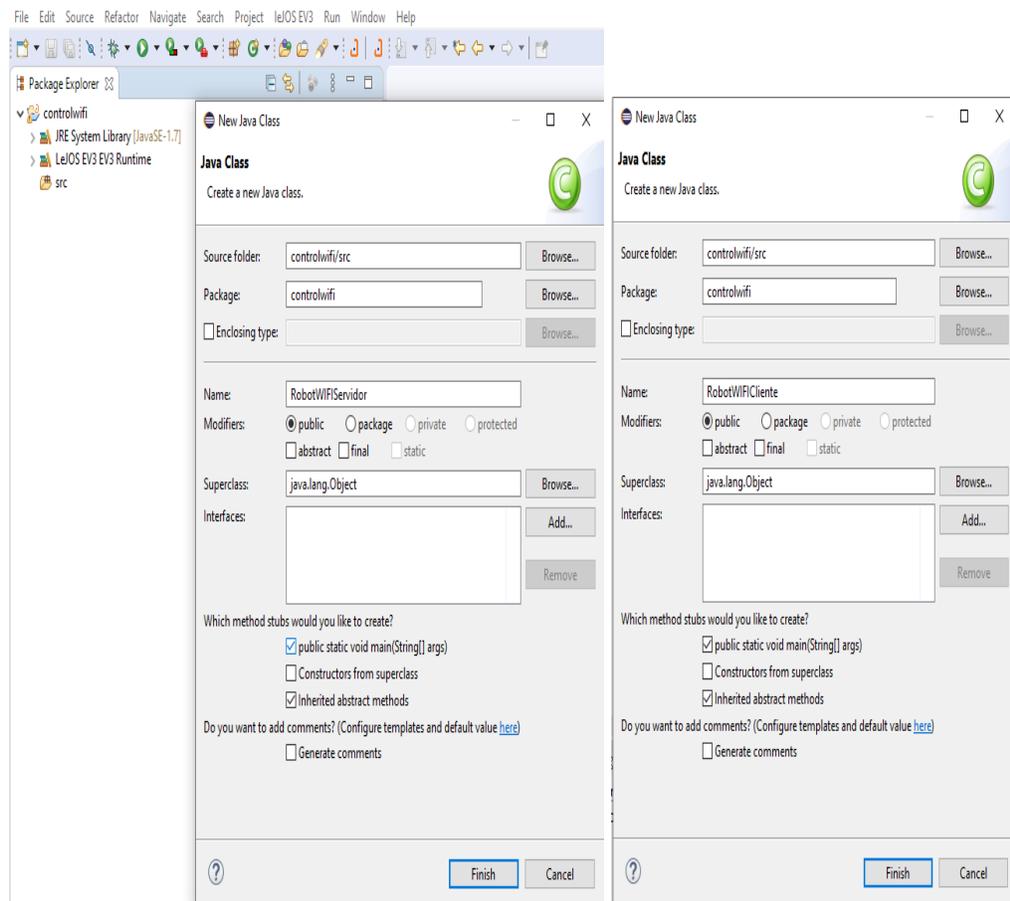


Creación del programa de Telecontrol Wifi del robot móvil EV3MEG.

Primeramente crear un nuevo proyecto con el nombre de “Control Wifi” siguiendo los pasos anteriores hasta llegar a **src** donde se usa un “New Java Package” con un nombre por default, lo que interesa es la creación de dos clases nuevas “New Java Class” con sus respectivos nombres (Figura 156).

Figura 156.

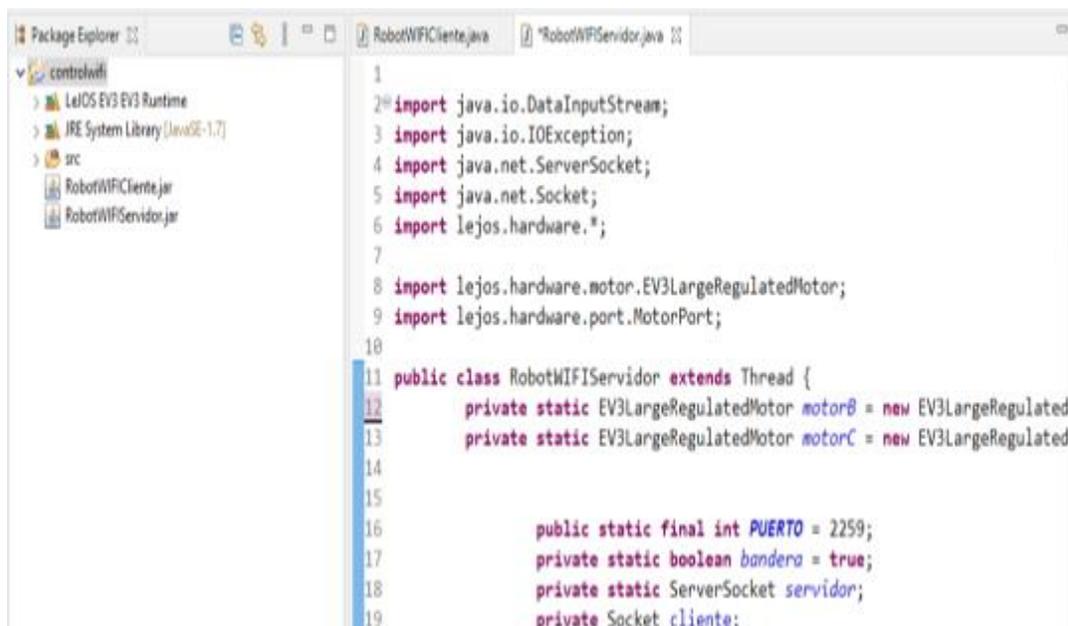
Creación de 2 New Java Class en Eclipse.



Creados las dos nuevas Clases de Java proceder con las líneas de programación para el Servidor del Robot. Primero se definen las variables a utilizar creando además dos objetos de la clase de EV3LargeRegulatedMotor dentro la programación para controlar nuestros motores B y C respectivamente. Usar un int (Entero) para definir el puerto de la conexión y un (private static boolean) que se usará como bandera para la manipulación del ciclo (mientras que “while”); el socket en el que el cliente se conectara será el objeto (ServerSocket), además de un objeto socket que nos servirá para obtener la información de la programación de la otra clase creada “cliente” después de la conexión.

Figura 157.

Inicio de Programación del "Servidor".



```

1
2 import java.io.DataInputStream;
3 import java.io.IOException;
4 import java.net.ServerSocket;
5 import java.net.Socket;
6 import lejos.hardware.*;
7
8 import lejos.hardware.motor.EV3LargeRegulatedMotor;
9 import lejos.hardware.port.MotorPort;
10
11 public class RobotWiFiServidor extends Thread {
12     private static EV3LargeRegulatedMotor motorB = new EV3LargeRegulated
13     private static EV3LargeRegulatedMotor motorC = new EV3LargeRegulated
14
15
16     public static final int PUERTO = 2259;
17     private static boolean bandera = true;
18     private static ServerSocket servidor;
19     private Socket cliente;

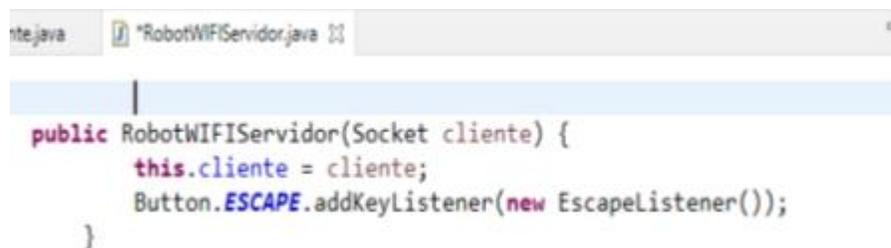
```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

La programación recibe un objeto Socket con la información del cliente de nuestra clase "Servidor" y lo guarda en el objeto "cliente" propio de la nueva clase creada. Además añade un KeyListener al botón de salida del Bloque EV3, para que de manera más óptima termine o cancele el programa que está corriendo.

Figura 158.

Líneas de programación para cancelar el programa.



```

public RobotWiFiServidor(Socket cliente) {
    this.cliente = cliente;
    Button.ESCAPE.addKeyListener(new EscapeListener());
}

```

El “main” crea un objeto `ServerSocket` con el puerto definido al inicio y repetidamente manda a llamar al método “`accept ()`” del objeto, hasta que una conexión se haya establecido. Una vez que se establece la conexión, se hace llamar al método “`start()`”, que ejecuta el método “`run()`” de la clase.

Figura 159.

Líneas de programación con main.

```

public static void main(String[] args) throws IOException {
    servidor = new ServerSocket(PUERTO);
    while(bandera) {
        System.out.println("Escuchando..");
        new RobotWiFiServidor(servidor.accept()).start();
    }
}

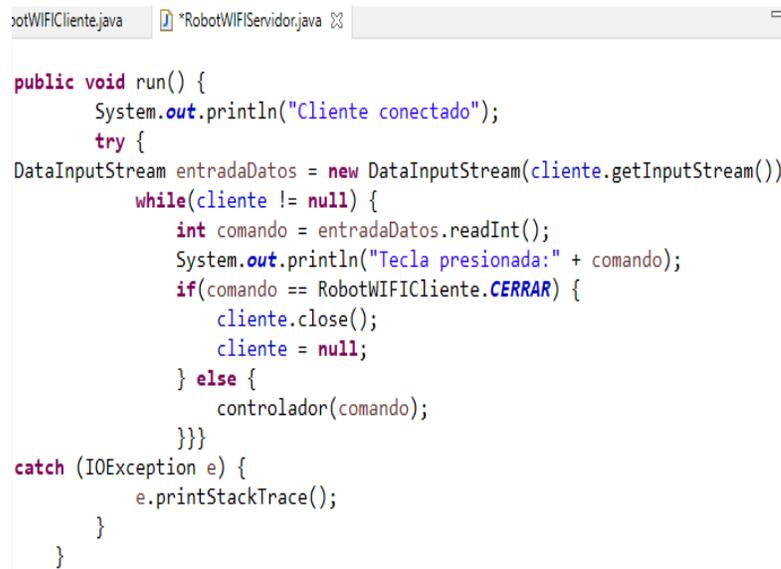
```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

Para estas líneas de programación del bloque se empieza por anunciar en la pantalla nuestra conexión exitosa. Luego se crea un objeto de la clase *DataInputStream* que nos servirá para extraer la salida de datos de nuestro cliente. La estructura de control *while* de las líneas de programación siguientes se encargará de traducir los datos a entero, imprimirlo en pantalla y mandar llamar al método que convertirá ese entero en una acción del robot. En caso de recibir un 0, el *while* termina y nuestro programa acaba.

Figura 160.

Líneas de programación para ejecución de datos “Servidor”.



```

RobotWIFICliente.java  *RobotWIFIServidor.java
public void run() {
    System.out.println("Cliente conectado");
    try {
        DataInputStream entradaDatos = new DataInputStream(cliente.getInputStream())
        while(cliente != null) {
            int comando = entradaDatos.readInt();
            System.out.println("Tecla presionada:" + comando);
            if(comando == RobotWIFICliente.CERRAR) {
                cliente.close();
                cliente = null;
            } else {
                controlador(comando);
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

Estas siguientes líneas de programación utilizan una estructura de control *switch* que de acuerdo al entero que nuestro Cliente haya enviado entrará al caso específico que dará movilidad al robot EV3MEG mediante los objetos que controlan los motores.

Figura 161.

Programación para ejecución de datos que controlan a los motores del robot.

```

public void controlador(int comando) {
    switch(comando) {
        case RobotWIFICliente.ATRAS:
            motorB.rotate(360, true);
            motorC.rotate(360);
            break;
        case RobotWIFICliente.ADELANTE:
            motorB.rotate(-360, true);
            motorC.rotate(-360);
            break;
        case RobotWIFICliente.DERECHA:
            motorC.rotate(315);
            break;
        case RobotWIFICliente.IZQUIERDA:
            motorB.rotate(315);
            break;
    }
}

```

Se crea una clase que implemente el interface del ciclo *while con KeyListener* con la información del cliente de nuestra clase “Servidor” y lo guarda en el objeto “cliente” propio de la nueva clase creada. Además añade un KeyPressed para el botón de salida del Bloque EV3, para que de manera que en caso de recibir un 0, el *while* termina y nuestro programa acaba.

Figura 162.

Programación para ejecución de datos que controlan a los motores del robot.

```
private class EscapeListener implements KeyListener {
    public void keyPressed(Key k) {
        bandera = false;
        System.exit(0);
    }
    public void keyReleased(Key k) {}
}
```

Al finalizar con las líneas de programación correspondiente a la clase “Servidor WIFI” verificar que no exista ningún error en la misma. Esto se puede observar en la parte inferior de la programación (Figura 162).

Figura 163.

Verificador de problemas y advertencias.

The screenshot shows a code editor with the following Java code:

```
75 private class EscapeListener implements KeyListener {
76     public void keyPressed(Key k) {
77         bandera = false;
78         System.exit(0);
79     }
80     public void keyReleased(Key k) {}
81 }
82
83 }
84
```

Below the code editor, the IDE's 'Problems' window is open, displaying the status: '0 errors, 2 warnings, 0 others'. A table below this status shows the details of the warnings:

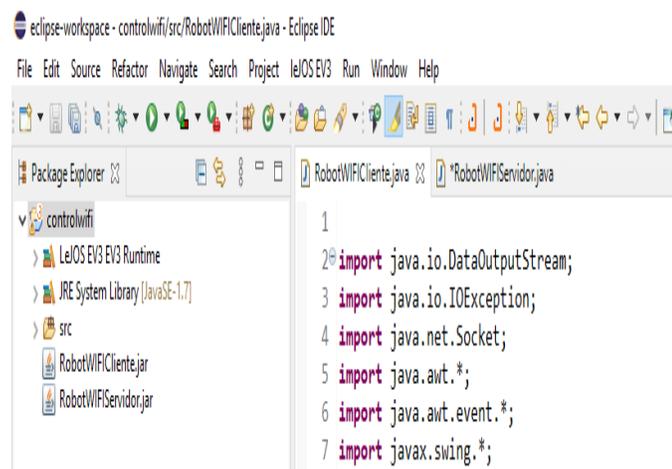
Description	Resource	Path	Location	Type
> Warnings (2 items)				

Two callout boxes with arrows point to the 'Problems' window: one points to the '0 errors, 2 warnings, 0 others' text, and another points to the 'Warnings (2 items)' row in the table.

Luego de revisar que no exista ningún error o problema en la programación de la clase de “Servidor” se procede con las líneas de programación de la otra clase de Java creada “Cliente”. Primero se define las variables a utilizar un `DataOutputStream` la cual es un flujo de salida de datos permite a una aplicación escribir tipos de datos primitivos de Java en un flujo de salida de forma portátil.

Figura 164.

Inicio de programación con definición de variables.



```

eclipse-workspace - controlwifi/src/RobotWiFiCliente.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project leOS EV3 Run Window Help
Package Explorer
controlwifi
  LeOS EV3 EV3 Runtime
  JRE System Library [JavaSE-1.7]
  src
  RobotWiFiCliente.jar
  RobotWiFiServidor.jar
RobotWiFiCliente.java
*RobotWiFiServidor.java
1
2 import java.io.DataOutputStream;
3 import java.io.IOException;
4 import java.net.Socket;
5 import java.awt.*;
6 import java.awt.event.*;
7 import javax.swing.*;

```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

Para que nuestra clase “Cliente” pueda usar interface con la anterior programación, se debe usar un “`extends JFrame`” que es el panel de contenido proporcionado por el panel raíz donde debe contener todos los componentes que no son de menú es decir usare un `int` (Entero) para definir el puerto, y los controles generales del robot en la conexión.

Figura 165.

Uso de la extensión para la acción del Key.

```

*RobotWIFICliente.java  *RobotWIFIServidor.java
10
11 public class RobotWIFICliente extends JFrame implements ActionListener, Key
12     /**
13     *
14     */
15     private static final long serialVersionUID = 2L;
16     public static final int PUERTO = RobotWIFIServidor.PUERTO;
17     public static final int CERRAR = 0;
18     public static final int ADELANTE = 38, // ^ = Adelante
19         IZQUIERDA = 37, // < = Izquierda
20         ATRAS = 40, // \ = Atrás
21         DERECHA = 39; // > = Derecha

```

Bajo las líneas de programación de los controles enteros usar un (private Socket) que es un punto final para la comunicación entre dos máquinas del ciclo (“while”); este enlazará con el socket que nos servirá para obtener la información de la programación de la otra clase creada “servidor” después de la conexión.

Además usar un boolean que nos dará la escritura e interpretación de las líneas de contexto que por lo general vendrán en su inicio con una J.

Figura 166.

Uso de private Socket para la comunicación entre máquinas.

```

*RobotWIFICliente.java  *RobotWIFIServidor.java
23
24     private Socket socket;
25     private DataOutputStream salidaDatos;
26     boolean conectado=false;
27     JLabel lblinstrucciones=new JLabel("^, \ Avanzar y Retroceder | <, > Izquierda
28     JTextField txtIPAddress =new JTextField();
29     JButton btnconectar=new JButton("Conectar");
30     JButton btnsalir=new JButton("Salir");
31     JLabel lblestado=new JLabel("Estado: Espera.");
32

```

Luego crear un *String* conjuntamente con el constructor de la nueva clase creada “Cliente”, esta recibe la dirección de *ip* como argumento y se encargará de

realizar la vinculación, usando esa dirección recibida como default en nuestro cuadro de texto.

Figura 167.

Uso del String para la dirección ip de vinculación.

```

*RobotWIFICliente.java  *RobotWIFIServidor.java
32
33 public RobotWIFICliente(String ip) {
34     this.setBounds(0,0,400,200);
35     this.setResizable(false);
36     btnconectar.setFont(new java.awt.Font("Arial", 0, 15));
37     lblinstrucciones.setBounds(20,60,400,20);
38     lblestado.setBounds(20,80,200,20);
39     txtIPAddress.setBounds(180,20,200,20);
40     btnsalir.setBounds(150,130,100,20);
41     btnconectar.setBounds(20,20,140,20); //X,Y,Largo,Ancho
42     txtIPAddress.setText(ip);
43     btnsalir.setForeground(Color.red);
44     setLayout(null);
45     this.add(lblinstrucciones);
46     this.add(txtIPAddress);
47     this.add(btnconectar);
48     this.add(btnsalir);
49     this.add(lblestado);
50     this.setVisible(true);
51     btnconectar.addActionListener(this);
52     btnsalir.addActionListener(this);
53 }

```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots*.

En la siguiente línea de instrucción figura 168 definir el main a la dirección IP determinada que usará el programa, y mandar a llamar al “Cliente” con la similar dirección IP. Para configurar el wifi en el Bloque EV3 y se nos muestre la dirección del *Internet Protocol (IP)*. Dirigirse al **(Anexo F)**.

Figura 168.

Definición del main con la dirección IP.

```

*RobotWIFICliente.java  *RobotWIFIServidor.java
32
56 public static void main(String args[]) {
57     String ip = "192.168.100.42";
58     if(args.length > 0) ip = args[0];
59     new RobotWIFICliente(ip);
60 }

```

En las siguientes líneas del programa se crea una instancia del objeto `Socket`, con el nombre del método `conectar`, mandando como argumento la dirección IP introducida en el campo de texto y el puerto del servidor. Además se solicita el objeto `DataOutputStream`, que guardará las teclas presionadas para que nuestro servidor pueda acceder a ellas. Se hace saber el estado de conexión en nuestra interfaz y en caso de “conexión exitosa”, se añade el `KeyListener` que estará escuchando las teclas presionadas en nuestro teclado de la pc.

Figura 169.

Creación de instancia del objeto Socket.

```
*RobotWIFICliente.java  *RobotWIFIServidor.java
63 public void conectar(){
64     try {
65         conectado=true;
66         btnconectar.setText("Desconectar");
67         socket = new Socket(txtIPAddress.getText(), PUERTO);
68         salidaDatos = new OutputStream(socket.getOutputStream());
69         lblestado.setText("Estado: Conexión exitosa.");
70         btnconectar.setText("Desconectar");
71         btnconectar.addKeyListener(this);
72     } catch (Exception exc) {
73         lblestado.setText("Estado: Conexión fallida.");
74         System.out.println("Error: " + exc);
75     }
76 }
```

El método que actualiza el estado en la interfaz es el `private void enviarComando`, este método sirve además para saber que comando se está enviando al servidor, donde convierte la tecla presionada en la PC a un entero (`int`) que va guardando en el `DataOutputStream` el cual contendrá el registro de las teclas presionadas.

Figura 170.

Método realizado para enviar Comando.

```
*RobotWiFiCliente.java  *RobotWiFiServidor.java
/o
79 private void enviarComando(int comando){
80     lblestado.setText("Estado: Enviando comando.");
81     try {
82         salidaDatos.writeInt(comando);
83     } catch(IOException io) {
84         lblestado.setText("Estado: Fallo al enviar el comando.");
85     }
86     lblestado.setText("Estado: Comando enviado exitosamente.");
87 }
```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

Las siguientes líneas de programación se encargarán de enviar el comando definido anteriormente para terminar la conexión. También cierra el socket y actualiza la interfaz, para saber que la conexión ha sido terminada.

Figura 171.

Actualización del estado en la interfaz.

```
*RobotWiFiCliente.java  *RobotWiFiServidor.java
89 public void desconectar(){
90     try{
91         enviarComando(CERRAR);
92         socket.close();
93         btnconectar.setText("Conectar");
94         lblestado.setText("Estado: Desconexión exitosa.");
95         conectado=false;
96         btnconectar.setText("Conectar");
97         btnconectar.removeKeyListener(this);
98     } catch (Exception exc){
99         lblestado.setText("Estado: Fallo en la desconexión.");
100         System.out.println("Error: " + exc);
101     }
102 }
```

El *actionPerformed* se encarga de escuchar qué botones de la interfaz son presionados, y manda a llamar al comando y método apropiado de acuerdo a cada botón propiamente presionado.

Figura 172.

Actualización del estado en la interfaz.

```

*RobotWIFICliente.java  *RobotWIFIServidor.java
103 public void actionPerformed(ActionEvent evt){
104     Object presionado=evt.getSource();
105     if(presionado==btnsalir){
106         System.exit(0);
107     }
108     if(presionado==btnconectar){
109         if(conectado==false){
110             conectar();
111         }
112         else if(conectado=true){
113             desconectar();
114     } }
115 }

```

Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

Luego de realizar las líneas de programación para el método de actionPerformed. Finalmente se tiene los métodos que nos servirán para interpretar las teclas del teclado presionadas y procesar el código de esa tecla al método enviarComando.

Figura 173.

Métodos para la interpretación del teclado.

```

*RobotWIFICliente.java  *RobotWIFIServidor.java
116 public void keyPressed(KeyEvent e) {
117     enviarComando(e.getKeyCode());
118 }
119 public void keyReleased(KeyEvent e) {}
120 public void keyTyped(KeyEvent arg0) {}
121
122 }

```

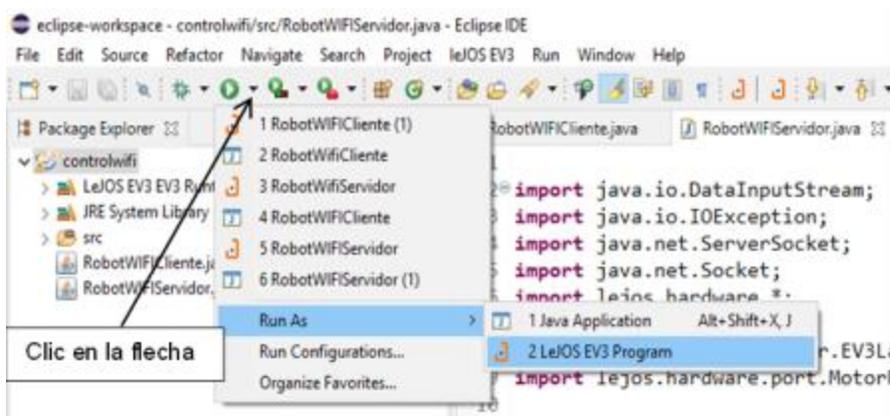
Nota: Es importante recalcar que la programación es tomada previamente de Zaldívar, D., Cuevas, E., Pérez-Cisneros, M., Esquivel Torres, S., & Oliva Navarro, D. A. (2018). *LEGO EV3: Programación de robots.*

Luego de culminar con las líneas de programación tanto para la clase “cliente” como para el “servidor” revisar que no exista ningún error en cada una de ellas y posteriormente enviar a correr por separado los programas respectivamente desde Eclipse.

Para enviar a correr la programación del “Servidor”, dirigir el cursor hacia la flecha que se encuentra al lado del *Run* dar clic y arrastrar el cursor hacia “*Run As*” donde inmediatamente se abrirá una ventana junto a la misma. La primera opción es para mandar a correr como una aplicación de Java y la segunda opción es la de correr como una aplicación o programa de leJOS EV3, elegir esta última, ya que se necesita enviar esta programación hacia al robot.

Figura 174.

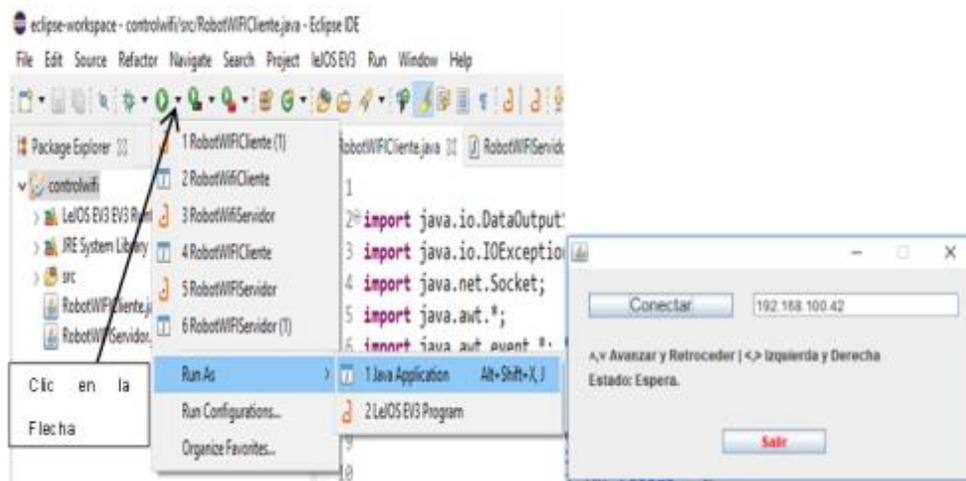
Explicación para el Envío del programa “Servidor”.



Por último para enviar a correr la programación del “Cliente”, dirigir el cursor hacia la flecha que se encuentra al lado del *Run* dar clic y arrastrar el cursor hacia “*Run As*”. Escoger la primera opción para mandar a correr como una aplicación de Java y se nos visualice los controles de mando en la pantalla de la PC.

Figura 175.

Explicación para el Envío del programa “Cliente”.



Nota: Dar clic en conectar y posteriormente nos saldrá en el estado “*conexión exitosa*” caso contrario saldrá “*conexión fallida*”, si es así, volver hacer los pasos anteriores.

CAPÍTULO IV

CONCLUSIONES

- Se logró implementar el sistema de telecontrol mediante dispositivos android y computadores tras haber recopilado información de libros, trabajos de titulación, sitios web entre otros del funcionamiento y las aplicaciones que se pueden realizar en el robot LEGO MINDSTORMS EV3.
- Utilizar el Software LEGO MINDSTORMS EV3 Home Edition que tiene un lenguaje de programación basado en LabVIEW además de un sistema de adquisición de datos que permite realizar la programación instintiva del robot usando la interfaz de programación del software para programar y ajustar los sensores (táctil ,infrarrojo, de color o luminosidad) además de los motores del robot.
- Se desarrolló una aplicación usando el entorno de desarrollo de software de App Inventor que permite mover los motores del robot hacia adelante, atrás, izquierda y derecha.
- Se ejecutaron diferentes tipos de pruebas de telecontrol usando el entorno de desarrollo de software de App Inventor hasta crear una aplicación que permitió manipular y controlar los motores del robot.
- Se realizó una interfaz de programación de aplicación (API) mediante el software leJOS EV3 mediante el acceso al control Wifi del robot tanto para la lectura y manipulación de sus motores.
- Se desarrollaron pruebas de conexión inalámbrica que permitieron comprobar el acceso remoto Wifi del robot para el control de movimiento de sus motores.

RECOMENDACIONES

- Instalar el software propio de LEGO en computadoras y dispositivos android ya que al momento de la instalación existen dos versiones de software (Home-Education), el más factible es el Education porque posee de muchas más herramientas que ayudan al desarrollo de programas más complejos.
- Con la ayuda del brick USB del bloque EV3 es posible aumentar los recursos de procesamiento con un mejor MICRO, se recomendaría que este tenga mayor cantidad de memoria, la cual ayudará mucho para aumentar la cantidad de órdenes de trabajo que deba realizar el robot sin ninguna dificultad en actividades complejas.
- Las pruebas que se realizaron estuvieron fructíferas puesto que se obtuvo el kit del robot en su versión doméstica #31313, y los estudiantes pueden realizar programas y controles más complejos ya sea bien por vía inalámbrica o alámbrica.
- En vista de que se utilizan motores y sensores que demandan gran cantidad de corriente se recomienda realizar las pruebas con las baterías recargadas al 100%.

REFERENCIAS BIBLIOGRÁFICAS

- Acosta, A. S. (09 de 2016). *https://programamos.es/*. Recuperado el 23 de 02 de 2020, de *https://programamos.es/*: *https://programamos.es/la-interfaz-de-app-inventor-a-fondo/*
- Boogaarts, M. (2013). *www.lego.com*. Recuperado el 20 de 12 de 2019, de *https://www.lego.com/*:
https://www.lego.com/cdn/cs/set/assets/bltc79715c0087bf480/EV3MEG.pdf
- España, L. (5 de 10 de 2015). *EsMindstorms.com*. Recuperado el 21 de 12 de 2019, de *https://www.esmindstorms.com*: *https://www.esmindstorms.com/sensor-infrarrojo-ev3/*
- Gallegos, E. G., & Mayorga, E. G. (2015). *Repositorio Digital UCSG*. Recuperado el 17 de 12 de 2019, de *http://repositorio.ucsg.edu.ec/*:
http://repositorio.ucsg.edu.ec/bitstream/3317/3926/1/T-UCSG-PRE-TEC-ITEL-87.pdf
- Heras, G. H. (20 de 11 de 2017). *http://haciaespacio.aem.gob.mx*. Recuperado el 11 de 10 de 2019, de *http://haciaespacio.aem.gob.mx*:
http://haciaespacio.aem.gob.mx/revistadigital/articul.php?interior=733
- Köhler, S., Shaw, A., & Griffiths, L. (28 de 01 de 2016). *SourceForge.net*. Recuperado el 20 de 04 de 2020, de *https://sourceforge.net/*:
https://sourceforge.net/p/lejos/wiki/Getting%20started%20with%20leJOS%20EV3/
- LEGO®MINDSTORMS®. (2013). *LEGOeducation.com*. Recuperado el 22 de 10 de 2019, de *LEGOeducation.com/MINDSTORMS*: *https://le-www-live-*

s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_esmx-6ac740d3cdd578cc6a52d10d7d173da9.pdf?la=en-au

Navarro, D. Z., Cuevas, E., Pérez, M., Esquivel, S., & Oliva, D. (2018). *LEGO EV3. Programación de Robots*. Grupo Editorial RA-MA.

Nino, L. B. (16 de 08 de 2013). *blog.electricBricks*. Recuperado el 25 de 11 de 2019, de <http://blog.electricbricks.com>: <http://blog.electricbricks.com/2013/08/comparativa-entre-nxt-y-ev3-2/?fbclid=IwAR3uAWg59zeDLzoEg4SXGvYeubWUyC-CJQ9UHRFFqqW82aTVK-ZVfBf3Wik>

Pérez, J. (10 de 07 de 2015). *SlideShare.net*. Recuperado el 10 de 10 de 2019, de <https://es.slideshare.net>: <https://es.slideshare.net/JosePerez382/robotica-con-mindstrom-ev3>

Presilla, Á. V. (09 de 2017). *Archivo Digital UPM*. Recuperado el 20 de 12 de 2019, de http://oa.upm.es/48722/1/PFC_ANGEL_VAZQUEZ_PRESILLA.pdf: http://oa.upm.es/48722/1/PFC_ANGEL_VAZQUEZ_PRESILLA.pdf

Prieto, F. P. (01 de 2015). <http://canaltic.com/rb>. Recuperado el 25 de 11 de 2019, de <http://canaltic.com/rb>: http://canaltic.com/rb/legoev3/2_el_software_lego_ev3.html

Solórzano, S. (2012). *Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE*. Recuperado el 21 de 12 de 2019, de Repositorio Dspace: <http://repositorio.espe.edu.ec/handle/21000/7436>

ANEXOS