

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



PROYECTO FIN DE CARRERA
Ingeniería de Telecomunicación

**DESARROLLO DE UN SISTEMA ROBÓTICO PARA EL
REPARTO PERSONALIZADO DE MANOS DE CONTRACT
BRIDGE.**

Alejandro De Agustín Urea

Julio 2016

DESARROLLO DE UN SISTEMA ROBÓTICO PARA EL REPARTO PERSONALIZADO DE MANOS DE CONTRACT BRIDGE.

**AUTOR: Alejandro De Agustín Urea.
TUTOR: Xavier Alamán.**



**AmIlab (Ambient Intelligence laboratory)
Dpto. Ingeniería Informática.
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2016**

Resumen

Este Proyecto fin de Carrera ha consistido en el estudio, diseño y construcción de un sistema robótico que es capaz de repartir manos de cartas personalizadas para el juego del Contract Bridge.

El sistema robótico constará de dos partes: una parte hardware, que atenderá al diseño y construcción del sistema robótico en sí, y de una parte software, que será capaz de controlar el robot, y de presentar un interfaz para definir la mano que se desea repartir.

Palabras clave.

Contract Bridge, sistema robótico.

Abstract

This thesis consists on the study, design and build a suitable robotic system to distribute custom hands for the game of Bridge cards.

The robotic system has two parts: software, programming language for sending commands to the robotic system; and hardware part will consist of the design and construction of the system, with its conclusions to an appropriate design.

Key Words.

API, algorithm, robot, C#, Arduino, LEGO Mindstorms EV3, Phidgets, Meccano, Fischertechnik, Raspberry pi, Bridge.

Agradecimientos

Todo es un reto, y en este proyecto he aprendido que todo es posible si uno se pone a trabajar para que las cosas funcionen y más la resolución de un problema cotidiano cómo es este proyecto.

En primer lugar, agradezco a mi tutor del proyecto Xavier Alamán por haberme dado la oportunidad de realizar este proyecto.

Agradezco a mi pareja y esposa Nohely Pacheco por haberme apoyado en todo este camino, dándome ideas y ayudándome para llevar a cabo este proyecto todos los días. También agradezco a mis padres y a mi hermano por haberme apoyado y darme ánimos en los momentos difíciles de este camino.

Agradezco a todas las personas tanto profesores como amigos y compañeros, que me han acompañado por este duro camino todos estos años. En especial: J. del Sol, S. Carrero, D. Agra, P. Sanz, Pili, R. Lara, D. Izquierdo, R. Tolosana, R. Jiménez.

Concluyendo he conseguido una cosa más que es ser Ingeniero de Telecomunicación, cerrando así una etapa y abriéndose otras más. Así que gracias a todos por todos estos años.

Alejandro De Agustín.

Julio 2016

INDICE DE CONTENIDOS

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Organización de la memoria	2
2	Estado del arte	5
2.1	Introducción	5
2.2	Sistemas orientados al gran público para el desarrollo de robots	5
2.2.1	Arduino	5
2.2.1.1	Ventajas	7
2.2.1.2	Desventajas	8
2.2.2	Raspberry pi	9
2.2.2.1	Ventajas	11
2.2.2.2	Desventajas	13
2.2.3	Phidgets	13
2.2.3.1	Ventajas	15
2.2.3.2	Desventajas	15
2.2.4	Meccano	15
2.2.4.1	Ventajas	18
2.2.4.2	Desventajas	18
2.2.5	LEGO Mindstorms EV3	19
2.2.5.1	Ventajas	20
2.2.5.2	Desventajas	21
2.2.6	Fischertechnik	21
2.2.6.1	Ventajas	23
2.2.6.2	Desventajas	23
2.3	Conclusión y justificación de la plataforma robótica adecuada	24
3	Estudio de mercado del sistema robótico	25
3.1	Introducción	25
3.1.1	Demanda	25
3.1.2	Oferta	26
4	Desarrollo e implementación del sistema robótico	31
4.1	Algoritmos de funcionamiento	31
4.1.1	Algoritmo básico de funcionamiento	31
4.1.2	Algoritmo A	33
4.1.3	Algoritmo B	35
4.2	Diseño y construcción	37
4.2.1	Modelo 1	39
4.2.2	Modelo 2	46
5	Pruebas de campo	51
5.1	Procedimiento	51
5.2	Pruebas del modelo 1	52
5.2.1	Algoritmo A	52
5.2.2	Algoritmo B	55
5.2.3	Conclusiones y posibles mejoras del modelo 1	56
5.3	Pruebas del modelo 2	57
5.3.1	Algoritmo B	57
6	Líneas de futuro y conclusiones	58

6.1 Conclusiones	58
6.2 Trabajo futuro	59
Referencias	61
Glosario.....	63
Anexos	LXV
A Manual de instalación	LXV
B Manual del programador.....	LXVIII
B.1 Formato de fichero de texto para el reparto de manos.	LXVIII
B.2 Interfaz de usuario.....	LXIX
C Tabla de resultados.....	LXIX
C.1 Modelo 1.	LXX
C.2 Modelo 2.	LXXIII
D Material entregado.	LXXVIII

INDICE DE FIGURAS

FIGURA 2-1: ENTORNO DE PROGRAMACIÓN DE ARDUINO.....	5
FIGURA 2-2: LOGO DE ARDUINO SOFTWARE.....	6
FIGURA 2-3: PLACA MODELO ARDUINO UNO.....	6
FIGURA 2-4: PRECIOS Y ESPECIFICACIONES TÉCNICAS DE ALGUNOS MODELOS PRINCIPALES DE PLACAS ARDUINO.....	8
FIGURA 2-5: LOGO DE RASPBERRY PI.....	9
FIGURA 2-6: PLACA MODELO B RASPBERRY PI 1.....	10
FIGURA 2-7: NOOBS PARA RASPBERRY PI.....	11
FIGURA 2-8: PRECIOS DE LOS MODELOS PRINCIPALES DE PLACAS RASPBERRY PI.....	12
FIGURA 2-9: LOGO PHIDGETS.....	13
FIGURA 2-10: PLACA DEL MODELO 8/8/8 PHIDGET INTERFACE KIT.....	14
FIGURA 2-11: EJEMPLO DE CONEXIÓN DE UN SENSOR ANALÓGICO Y UN LED PARA LA PLACA DEL MODELO 8/8/8 PHIDGET INTERFACE KIT.....	14
FIGURA 2-12: LOGO DE MECCANO.....	16
FIGURA 2-13: MICROCONTROLADOR MECCABRAIN DE MECCANO.....	16
FIGURA 2-14: ESQUEMA DE CONEXIÓN DE VARIOS MÓDULOS INTELIGENTES AL MICROCONTROLADOR MECCABRAIN DE MECCANO.....	17
FIGURA 2-15: CONEXIÓN DE VARIOS MÓDULOS INTELIGENTES AL MICROCONTROLADOR MECCABRAIN DE MECCANO.....	17
FIGURA 2-16: LOGO DE LEGO MINDSTORMS EV3.....	19
FIGURA 2-17: LADRILLO INTELIGENTE DE LEGO MINDSTORMS EV3.....	19
FIGURA 2-18: ENTORNO DE PROGRAMACIÓN DADO POR EL FABRICANTE PARA LEGO MINDSTORMS EV3.....	20
FIGURA 2-19: LOGO DE FISCHERTECHNIK.....	21
FIGURA 2-20: MICROCONTROLADOR ROBO LT CONTROLLER DE FISCHERTECHNIK.....	22

FIGURA 2-21: ENTORNO DE PROGRAMACIÓN DADO POR EL FABRICANTE ROBO PRO SOFTWARE PARA FISCHERTECHNIK ROBOTIC.	22
FIGURA 3-1: ESTIMACIÓN DEL NÚMERO DE JUGADORES EN EL MUNDO SEGÚN LA ORGANIZACIÓN MUNDIAL DE BRIDGE, CON DATOS DEL AÑO 2015.	25
FIGURA 3-2: SISTEMA ROBÓTICO PLAYBRIDGEDEALER4+.	27
FIGURA 3-3: FIGURA 3-2: SISTEMA ROBÓTICO DUPLIMATE.....	27
FIGURA 3-4: SISTEMA ROBÓTICO PARA USO PARTICULAR: BRIDGE PORT AUTOMATIC PLAYING CARD.....	28
FIGURA 3-5: SISTEMA ROBÓTICO PARA USO PARTICULAR, DISEÑADO POR UN USUARIO CON PIEZAS DE OTROS FABRICANTES.....	29
FIGURA 3-6: SISTEMA ROBÓTICO”DO IT YOURSELF” CON LA PLATAFORMA LEGO MINDSTORMS..	30
FIGURA 3-7: SISTEMA ROBÓTICO”DO IT YOURSELF” CON LA PLATAFORMA RASPBERRY PI CON PIEZAS DE LEGO MINDSTORMS.....	30
FIGURA 4-1: ALGORITMO BÁSICO DE FUNCIONAMIENTO.....	32
FIGURA 4-2: ALGORITMO A.....	34
FIGURA 4-3: ALGORITMO B.....	36
FIGURA 4-4: CAJA DEL MODELO 4544 LEGO MINDSTORM EV3.....	37
FIGURA 4-5: COMPONENTES DE LA CAJA DEL MODELO 4544 LEGO.	37
FIGURA 4-6: PIEZAS DE LA CAJA DEL MODELO 4544 LEGO.	38
FIGURA 4-7: CAJA DEL MODELO 31046 LEGO.....	38
FIGURA 4-8: PIEZAS DE LA CAJA DEL MODELO 31046 LEGO.....	38
FIGURA 4-9: MODELO INICIAL CON PIEZAS DEL KIT 4544 DE LEGO.	40
FIGURA 4-10: PIEZAS USADAS PARA EL ENSAMBLAR EL SOPORTE DEL SISTEMA ROBÓTICO DEL KIT 4544 DE LEGO; JUNTO CON CUATRO RUEDAS DEL KIT 31046 DE LEGO.	41
FIGURA 4-11: PIEZAS USADAS PARA ENSAMBLAR EL SISTEMA DE ROTACIÓN JUNTO CON UN SERVOMOTOR LARGO, CORRESPONDIENTES AL KIT 4544 DE LEGO.	41
FIGURA 4-12: PIEZAS USADAS PARA RAMPA CON EL KIT 4544 DE LEGO.....	42
FIGURA 4-13: PIEZAS PARA ENSAMBLAR LA BANDEJA DE REPARTO Y EL LADRILLO INTELIGENTE EV3 CON EL KIT 4544 LEGO.....	42
FIGURA 4-14: RODILLO DE REPARTO DE CARTAS USANDO UN SERVOMOTOR LARGO CON EL KIT 4544 DE LEGO.....	43

FIGURA 4-15: MODELO INICIAL 2 CORRESPONDIENTE AL MODELO 1 CON PIEZAS DE LOS KITS 4544 Y 31046 DE LEGO.	44
FIGURA 4-16: BANDEJA DE REPARTO DE CARTAS MEJORADA CON PIEZAS DE LOS KITS 4544 DE LEGO.....	44
FIGURA 4-17: RAMPA MEJORADA CON PIEZAS DE LOS KITS 4544 DE LEGO.	45
FIGURA 4-18: MODELO ÓPTIMO CORRESPONDIENTE AL MODELO 1 CON PIEZAS DE LOS KITS 4544 Y 31046 DE LEGO.	46
FIGURA 4-19: BANDEJA DÓNDE SE VA A REPARTIR LAS CARTAS CORRESPONDIENTES AL MODELO 2 CON PIEZAS DEL KIT 4544 DE LEGO.	47
FIGURA 4-20: SOPORTE DEL SISTEMA ROBÓTICO DEL MODELO 2 CON PIEZAS DEL KIT 4544 DE LEGO.....	48
FIGURA 4-21: MODELO ÓPTIMO DEL MODELO 2 CON PIEZAS DEL KIT 4544 DE LEGO.....	49
FIGURA 5-1: EXPRESIÓN DE TASA DE FALLOS.	51
FIGURA 0-1. LOGO MICROSOFT VISUAL STUDIO 2013.....	LXV
FIGURA 0-2: CREACIÓN DE UN PROYECTO.	LXV
FIGURA 0-3: OPCIONES DEL PROYECTO.	LXVI
FIGURA 0-4: INSTALACIÓN DE LA API PARA LA PLATAFORMA LEGO MINDSTORMS EV3.....	LXVI
FIGURA 0-5:	LXVII
FIGURA 0-6: INTERFAZ DE USUARIO.....	LXIX

b

INDICE DE TABLAS

TABLA 2-1: DISTINTOS TIPOS DE USO PARA LOS DISTINTOS MODELOS.	7
TABLA 2-2: TABLA COMPARATIVA DE LAS DISTINTAS PLATAFORMAS ROBÓTICAS PRESENTADAS CON LAS CARACTERÍSTICAS DE CARA AL OBJETIVO AL PROYECTO.....	24
TABLA 5-1: PARÁMETROS DEL ALGORITMO A PARA EL MODELO 1 (*).	53
TABLA 5-2: GRADOS SEGÚN LA CARTA SELECCIONADA.....	55
TABLA 5-3: PARÁMETROS DEL ALGORITMO B (**).	55
TABLA 0-1: FORMATO DE CORRESPONDENCIA DEL FICHERO CON LAS CARTAS.....	LXVIII
TABLA 0-2: PRUEBAS REALIZADAS PARA EL ALGORITMO A DEL MODELO 1 CUANDO REPARTE EN UNA ÚNICA POSICIÓN (*).	LXX
TABLA 0-3: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 1 (**).	LXXI
TABLA 0-4: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 1 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS A CADA JUGADOR (***).	LXXII
TABLA 0-5: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS A CADA JUGADOR (****).	LXXIII
TABLA 0-6: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA CADA POSICIÓN.	LXXIV
TABLA 0-7: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA LOS JUGADORES 1 A 2.	LXXV
TABLA 0-8: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA LOS JUGADORES 1 A 3.	LXXV
TABLA 0-9: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA LOS JUGADORES 1 A 4.	LXXV
TABLA 0-10: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA LOS JUGADORES 2 A 3.	LXXVI
TABLA 0-11: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA LOS JUGADORES 2 A 4.	LXXVI
TABLA 0-12: PRUEBAS REALIZADAS PARA EL ALGORITMO B DEL MODELO 2 CUANDO EL SISTEMA ROBÓTICO REPARTE LAS CARTAS CON GRADOS PARA LOS JUGADORES 3 A 4.	LXXVII

1 Introducción

1.1 Motivación

El Bridge es un juego de naipes de cuatro jugadores que se juega por parejas. En cada ronda (también denominada “mano”) una de las parejas debe ganar como mínimo un número de bazas previamente acordado en una subasta entre los jugadores. Este juego de cartas se compara en complejidad al juego del ajedrez, compartiendo con él muchas de sus circunstancias. Por ejemplo, se imparte como asignatura en muchos centros universitarios (en la UAM se imparte como asignatura de libre elección), es deporte federado y se están realizando las gestiones para que participe en las próximas olimpiadas como deporte invitado.

Uno de los problemas para empezar a jugar al Bridge es que es necesario aprender todo un conjunto de normas y convenciones, cada una de las cuales se da realmente en el juego en pocas ocasiones. Es por ello necesario preparar manualmente distribuciones de cartas para que los estudiantes puedan practicar situaciones concretas que en juego real se dan raramente. Por otro lado, cuando se realizan campeonatos, la misma mano se tiene que jugar en distintas mesas, para poder comparar los resultados de cada pareja con las demás parejas en las mismas situaciones (es lo que se denomina “Bridge duplicado”).

En ambos casos (para practicar o para jugar campeonatos) el preparar “manos” con una distribución de cartas prefijada es un requisito frecuente en la práctica del Bridge. Una primera solución es jugar “por ordenador”, ya sea sólo o con otras personas. Hay distintos sitios Web que permiten tal tipo de juego, que son relativamente populares entre los jugadores de Bridge, tanto novicios como experimentados. En este caso la preparación de manos es más sencilla, ya que una vez elegida la distribución de cartas de una mano, enseñar a los jugadores la mano escogida se puede hacer con algoritmos relativamente sencillos.

Cuando se quiere jugar con naipes reales, el problema es algo más complicado. Para el caso de clubes de Bridge y otros organismos que organizan competiciones, se suelen emplear robots de reparto de manos que automatizan el proceso. Estos robots son bastante fiables y resuelven razonablemente el problema. Sin embargo para el caso de personas particulares que quieren aprender, o de personas con mayor experiencia que quieren practicar determinadas situaciones, esta solución no suele ser factible por motivos económicos: los robots de reparto de naipes suelen costar entre 3000 y 4000 dólares. Este coste viene motivado por la necesidad de una alta fiabilidad: cuando se reparten manos para ser usadas en competición no es aceptable que haya fallos (la tasa de fallos tiene que ser extremadamente baja).

Sin embargo en el caso de personas particulares (que pueden incluir a profesores de bridge preparando manos para sus alumnos) no es necesaria una fiabilidad tan alta (la aparición ocasional de un error en una mano puede ser fácilmente reparada), y en cambio se requeriría un coste más asequible. Además sería conveniente tener un sistema “abierto” que permitiera conectar el robot de reparto con distintos tipos de software para la generación de las distribuciones de cartas.

En este Proyecto Fin de Carrera se ha diseñado, implementado y probado una posible solución económica basada en un kit de construcción asequible y fácil de usar: LEGO Mindstorm. La idea es que el robot pueda ser montado por el propio usuario final, al que se le facilitarán un programa de control y unas instrucciones de montaje y empleo. De esta manera, siguiendo la filosofía del “do-it-yourself”, personas particulares que quieran aprender o practicar el bridge podrán disponer de una solución de bajo coste para repartir manos preparadas.

1.2 Objetivos

En este Proyecto fin de Carrera se propone estudiar y resolver el siguiente problema: construir un sistema robótico para poder repartir manos de cartas personalizadas para el juego del Bridge. Para ello, este sistema tiene que ser capaz de saber qué carta está repartiendo en cada momento, y distribuirla entre cuatro mazos de cartas según haya decidido el profesor, empleando el programa asociado.

Un primer paso a la hora de desarrollar este sistema robótico es decidir cuál de las tecnologías existentes en el mercado es más económica, fiable y fácil de usar. También se debe tener en cuenta que la construcción y funcionamiento sean lo más robustos posible para su propósito final.

1.3 Organización de la memoria

- **Estudio del estado del arte**

En primer lugar se estudiará qué plataformas para el desarrollo de robots existen, sopesando sus pros y contras para las necesidades del proyecto. Para ello, se considerarán las siguientes plataformas: Arduino, Phidgets, Raspberry pi, LEGO Mindstorm EV3, Meccano y Fischertechnics.

En segundo lugar, se presentarán las conclusiones obtenidas del punto anterior, justificando la elección de la plataforma robótica para el proyecto.

- **Estudio de mercado del sistema propuesto**

Se realizará un estudio del posible mercado del sistema a desarrollar, con vistas a su posible comercialización.

- **Desarrollo e implementación del sistema robótico**

En primer lugar se presentarán los distintos algoritmos de funcionamiento que se han probado para conseguir un reparto fiable de las cartas. Se mostrará primero el algoritmo básico de funcionamiento, para luego explicar las modificaciones que se realizaron tras ser probado. Los resultados y conclusiones de los algoritmos se analizarán en el capítulo “Pruebas de campo para cada modelo”.

Posteriormente, se explicaran los distintos prototipos de robot que se han realizado para conseguir el objetivo del proyecto: repartir automáticamente una mano de cartas de Contract Bridge según se indique en un programa que actuará como interfaz de usuario.

- **Pruebas de campo**

Para cada uno de los prototipos robóticos construidos se realizará un estudio sobre la tasa de fallos y tiempo medio en repartir las cartas empleando los distintos algoritmos presentados anteriormente, con vistas a seleccionar el diseño más adecuado.

- **Líneas de Futuro y conclusión**

En este apartado se propondrán una serie de propuestas de mejora tanto sobre el sistema robótico en sí, como sobre las posibilidades de desarrollar sistemas de enseñanza e-Learning sobre dicho sistema.

2 Estado del arte

2.1 Introducción.

En este capítulo, se recoge una visión de los diferentes sistemas robóticos presentes en el mercado, orientados al gran público. No hay que olvidar que el objetivo es que el usuario final sea capaz de montarse su propio robot, a partir de unas instrucciones de montaje. El software, en cambio, lo obtendrá (ya sea gratuitamente o mediante pago) de una página web.

En la sección 2.2 se presentarán los diferentes sistemas mostrando sus ventajas y desventajas respecto al objetivo del proyecto. Se tendrán en cuenta las siguientes características: coste, entorno de programación, facilidad de montaje e interfaz usuario.

En la sección 2.3 se analizarán las conclusiones del capítulo, para posteriormente justificar la elección de la tecnología para este proyecto.

2.2 Sistemas orientados al gran público para el desarrollo de robots.

2.2.1 Arduino.

Arduino es una compañía de hardware libre que desarrolla placas en las que están integradas un microcontrolador y que pueden ser programadas mediante un entorno de desarrollo de programación (IDE) diseñado para facilitar el uso de la electrónica en distintos tipos de proyectos.

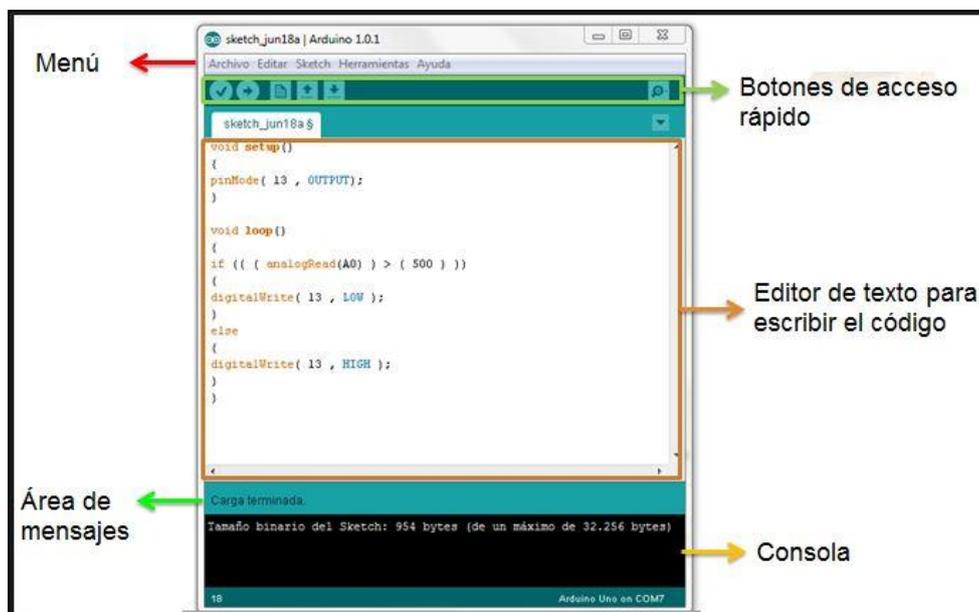


Figura 2-1: Entorno de programación de Arduino.

La primera placa Arduino fue introducida en el año 2005, con bajo coste y facilidad de uso para poder desarrollar proyectos interactivos en su entorno mediante actuadores y sensores.



Figura 2-2: Logo de Arduino Software.

A medida que fue pasando el tiempo se fueron introduciendo nuevos modelos, en los que se introdujeron nuevos microcontroladores como el CortexM3 y el ARM de 32 bits, que coexisten con los modelos originales que integran microcontroladores AVR de 8 bits.

El hardware está constituido principalmente por una placa de circuito impreso con un microcontrolador, de la marca Atmel AVR, y varios puertos digitales y analógicos de entrada y salida. A los puertos se les puede conectar sensores y actuadores para realizar un determinado propósito. Estas placas están disponibles de formas ensambladas o en forma de kits. Los esquemas de diseño del hardware están disponibles bajo licencia libre, permitiendo que cualquiera pueda crear su propia placa sin necesidad de comprar una prefabricada.

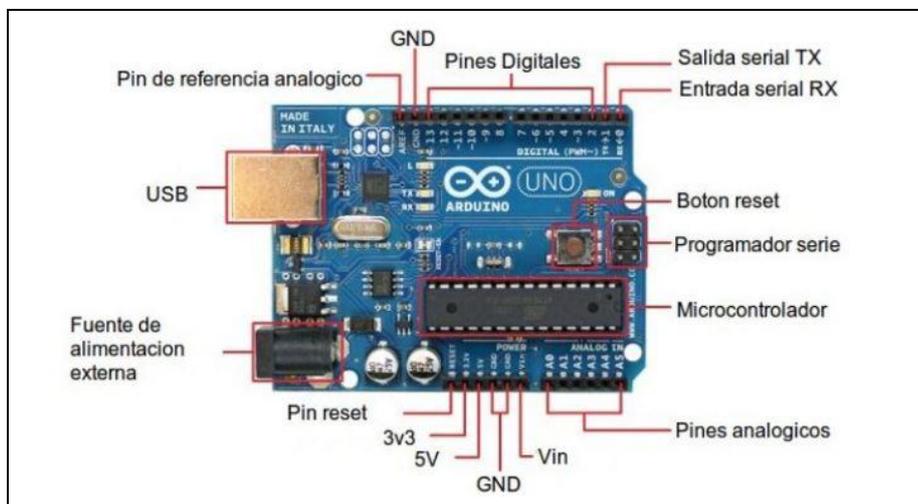


Figura 2-3: Placa modelo Arduino Uno.

El entorno de desarrollo integrado se puede descargar de forma gratuita desde la página de Arduino. Se pueden desarrollar objetos interactivos autónomos, o conectar la placa a software tal como Adobe Flash o Processing. También disponen de una tarjeta especial para desarrollar interfaces empleando software como JAVA, Visual Basic y LabVIEW.

ENTRY LEVEL	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO UNO</div> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO PRO</div> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO PRO MINI</div> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO MICRO</div> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO NANO</div> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO STARTER KIT</div> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO BASIC KIT</div> <div style="background-color: #FFA500; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO MOTOR SHIELD</div> </div>
ENHANCED FEATURES	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO MEGA</div> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO ZERO</div> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO DUE</div> <div style="background-color: #FFA500; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO PROTO SHIELD</div> </div>
INTERNET OF THINGS	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO YÚN</div> <div style="background-color: #FFA500; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO ETHERNET SHIELD</div> <div style="background-color: #FFA500; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO GSM SHIELD</div> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO WIFI SHIELD 101</div> </div>
WEARABLE	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO GEMMA</div> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO LILYPAD</div> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO LILYPAD SIMPLE</div> <div style="background-color: #008080; color: white; padding: 2px 5px; font-size: 8px;">ARDUINO LILYPAD USB</div> </div>
3D PRINTING	<div style="display: flex; justify-content: space-between; align-items: center;"> <div style="background-color: #8B4513; color: white; padding: 2px 5px; font-size: 8px;">MATERIA 101</div> </div>

Tabla 2-1: Distintos tipos de uso para los distintos modelos.

2.2.1.1 Ventajas.

- a) Homogeneidad en el uso de los microcontroladores, debido a que los modelos son muy parecidos entre sí, por lo que es sencillo aplicar uno u otro según el propósito final.
- b) Se puede incluir cualquier componente electrónico (sensores, actuadores, etc.) que soporte las especificaciones técnicas.
- c) El entorno de programación es válido para cualquier sistema operativo (Windows, Mac OS X, Linux) de cualquier arquitectura (32 y 64 bits) y gratuito: se puede descargar de la página oficial de Arduino.
- d) Existen en internet manuales, foros y librerías extras del entorno de programación.
- e) El precio para las placas está en torno a 20€ si es una versión oficial, y de 8€ en otros fabricantes. El precio medio de las placas de expansión, los kits, accesorios e impresoras 3d puede rondar entre los 10 a 700 € dependiendo del propósito final.

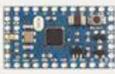
								
Fabricante	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Arduino	Netduino
Modelo	Pro Mini	Nano	Uno	Mega / Mega 2560	Leonardo	Micro	Due	Netduino 2
Microcontrolador	AVR Atmega 168 ó 328 8bits	AVR ATmega 168 ó 328 8bits	AVR ATmega 328 8bits	AVR ATmega2560 8bits	AVR ATmega 32u4 8bits	AVR ATmega 32u4 8bits	ARM SAM3X8E Cortex-M3 32bits	ARM STMicro STM32F2 Cortex-M3 32bits
Frecuencia	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	16Mhz	84Mhz	120Mhz
Voltaje de alimentación por el USB	3.3v ó 5v (sin usb)	5v	5v	5v	5v	5v	5v	5v
Voltaje de alimentación recomendado por el Jack	3.35 -12 V (modelo 3.3V) ó 5 - 12 V (modelo 5V)	7~12v	7~12v	7~12v	7~12v	7~12v	7~12v	7.5~9v
Voltaje de alimentación limite por el Jack	-	6~20v	6~20v	6~20v	6~20v	6~20v	6~20v	-
Precio oficial	15+gi	-	20€+gi	40€+gi	18€+gi	18€+gi	39€+gi	~35\$+gi
Precio BBB	~4€	~9€	~10€	~12€	11€~	~16€	~38€	25~30€

Figura 2-4: Precios y especificaciones técnicas de algunos modelos principales de placas Arduino.

2.2.1.2 Desventajas.

- Es necesario tener conocimientos básicos de electrónica digital y analógica, para conectar los sensores y actuadores.
- Es necesario tener conocimientos de programación, especialmente en el lenguaje de programación C++, basado en Processing y Wiring.
- Puede ser necesario tener conocimientos básicos de comunicaciones e incluso de procesado de señal para poder conectar y hacer funcionar los diferentes componentes conectados a la placa Arduino.
- Para poder construir el robot es necesario utilizar piezas adicionales que no son las oficiales de Arduino.
- Se necesita una cierta formación técnica para buscar información acerca de los componentes electrónicos que sean acordes a las especificaciones técnicas.

2.2.2 Raspberry pi.

Raspberry pi es un ordenador de placa única u ordenador de placa simple (SBC) de bajo coste desarrollado en Reino Unido por la Fundación Raspberry Pi, con objetivo de ayudar en la enseñanza de ciencias de la computación en las escuelas.

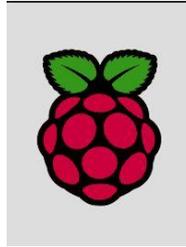


Figura 2-5: Logo de Raspberry pi.

Los primeros diseños fueron lanzados el año 2006 basados en el microcontrolador Atmel ATmega644. En el año 2009, se crea la Fundación Raspberry Pi en Reino Unido, como una asociación caritativa regulada por la comisión de Caridad de Inglaterra y Gales.

Las primeras placas fueron introducidas a lo largo del año 2012; pero debido a problemas en la provisión de CPUs y problemas burocráticos, no se pusieron en venta con mejores prestaciones hasta el año 2015.

El entorno de desarrollo de programación es open source, mientras que su hardware es un producto con propiedad registrada pero de uso libre: la fundación mantiene el control de la plataforma pero permite el uso particular con fines educativos.

El hardware está basado en el Broadcom BCM2835 Soc (System on chip), en el que se incluye un procesador ARM1176JZF-S de 700 MHz, una unidad de procesamiento de gráfico (GPU) de la versión VideoCore IV y una memoria RAM.

Hay varias versiones del hardware, con variaciones en la capacidad de memoria y en el soporte de periféricos.

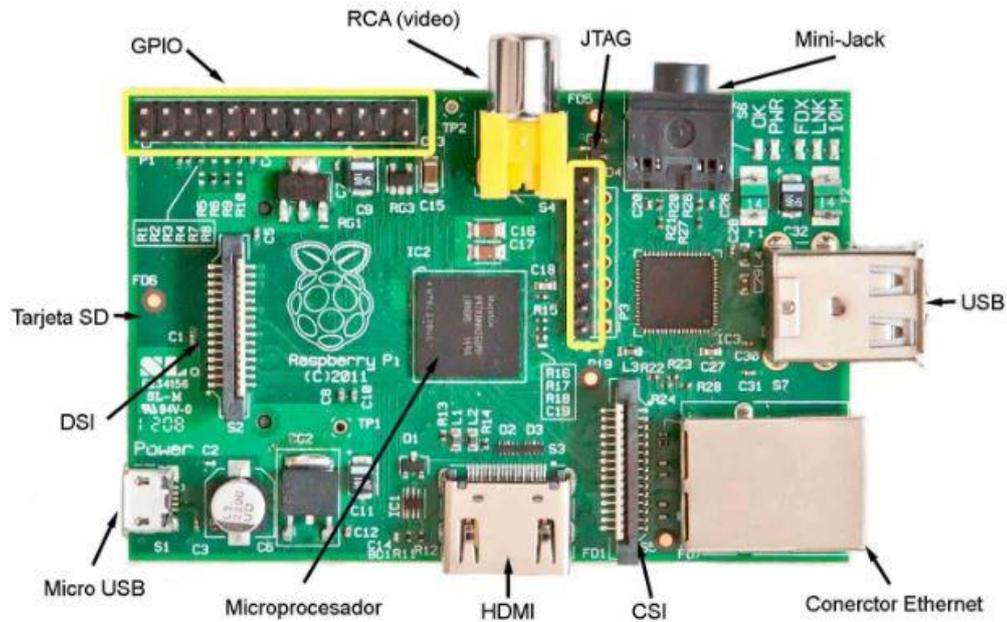


Figura 2-6: Placa modelo B Raspberry pi 1.

El software está basado en sistemas operativos basados en el núcleo de Linux: Raspbian; distribución derivada de Debian optimizada para estas placas.

Para poder instalar los diferentes entornos de desarrollo es necesario NOOBS (New Out of the Box) una interfaz de usuario gratuito para PC, que permite instalar distintos tipos de sistemas operativos para guardarlo en la tarjeta SD. Este permite instalar los siguientes sistemas operativos: Raspbian (Debian para Raspberry Pi), Open ELEC (liberación Optimizada de XBMC), RISC OS, Arch (Linux para Raspberry Pi), RaspBMC (Raspbian con XBMC) y Pildora (versión de Fedora para procesadores ARM).

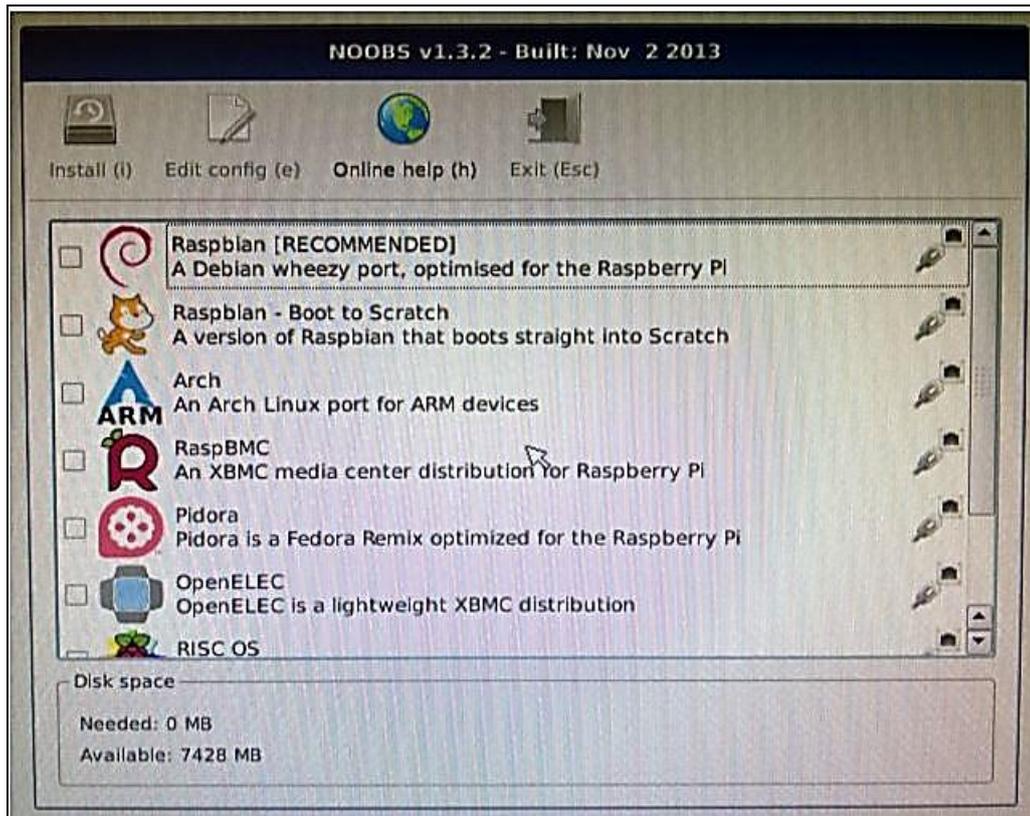


Figura 2-7: NOOBS para Raspberry pi.

Los lenguajes de programación que soporta son: Python, muy parecido al lenguaje de programación C/C++, y Scratch, orientado a la enseñanza mediante juegos.

2.2.2.1 Ventajas.

- Homogeneidad en el uso de los microcontroladores, debido a que los modelos son muy parecidos entre sí, por lo que es sencillo aplicar uno u otro según el propósito final.
- Se puede añadir cualquier componente electrónico (sensores, actuadores, etc.) que soporte las especificaciones técnicas. Se le pueden añadir packs de expansión oficiales (Cámara, Cámara infrarroja, HAT para expansión de placas y Gertboard) como no oficiales (sensores, actuadores).
- Se puede diseñar y construir para un amplio número de sistemas operativos: todas las derivaciones de Linux (entre las que se incluye Android y Raspbian), todas las derivaciones de UNIX, y Windows 10 (Windows CE).
- Ofrece otras funcionalidades: conexión a un MP3 y smartphones para escuchar música mediante el conector RCA y conector mini-jack, conexión a un monitor mediante un cable HDMI, conexión a Internet así como a otros dispositivos mediante el conector de Ethernet.

- e) El precio asequible en torno 20 a 35 dólares, mientras que los accesorios pueden ser cualquier componente electrónico cuyo precio varía según su propósito final.

	RPI MODEL A	RPI MODEL A+	RPI MODEL B	RPI MODEL B+	RPI 2 MODEL B
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv6 700 MHz.	ARM11 ARMv7 ARM Cortex-A7 4 núcleos @ 900 MHz.
GPU	Broadcom VideoCore IV 250 MHz. OpenGL ES 2.0				
RAM	256 MB LPDDR SDRAM 400 MHz.	256 MB LPDDR SDRAM 400 MHz.	512 MB LPDDR SDRAM 400 MHz.	512 MB LPDDR SDRAM 400 MHz.	1 GB LPDDR2 SDRAM 450 MHz.
USB 2.0	1	1	2	4	4
Salidas de video	HDMI 1.4 @ 1920x1200 píxeles				
Almacenamiento	SD/MMC	microSD	SD/MMC	microSD	microSD
Ethernet	No	No	Sí, 10/100 Mbps	Sí, 10/100 Mbps	Sí, 10/100 Mbps
Tamaño	85,60x56,5 mm	65x56,5 mm.	85,60x56,5 mm	85,60x56,5 mm	85,60x56,5 mm
Peso	45 g.	23 g.	45 g.	45 g.	45 g.
Precio	25 dólares	20 dólares	35 dólares	35 dólares	35 dólares

Figura 2-8: Precios de los modelos principales de placas Raspberry pi.

- f) Existen en Internet manuales, foros y librerías extras del entorno de programar para poder diseñar el robot.

2.2.2.2 Desventajas.

- a) Se emplean piezas adicionales no oficiales para el diseño y construcción del robot.
- b) Es necesario tener conocimientos básicos de electrónica digital y analógica, para conectar sensores y actuadores así como motores a la placa.
- c) Es necesario tener conocimientos sobre lenguajes de programación C/C++ y Python para poder enviar las acciones del robot.
- d) Es necesario tener conocimientos sobre distintos sistemas operativos para poder instalar el entorno de software.
- e) Hay que tener una formación técnica para poder buscar información acerca de los componentes electrónicos que sean acordes a las especificaciones técnicas.

2.2.3 Phidgets.

Phidgets es una compañía de hardware libre creada por Bernard Rousseau en el año 2010. Se encarga de crear sistemas de componentes electrónicos de bajo coste y sensores, controlados por un microcontrolador.



Figura 2-9: Logo Phidgets.

El hardware está basado en una placa microcontrolador al que se le conectan sensores analógicos (motores, giroscopios, etc.), actuadores, interruptores de control y un registro de pulsadores. En una placa se le puede conectar hasta 24 dispositivos: para saber si están conectados correctamente el fabricante proporciona una interfaz llamada Phidget Control Panel. Hay distintos modelos y evoluciones según su propósito final.

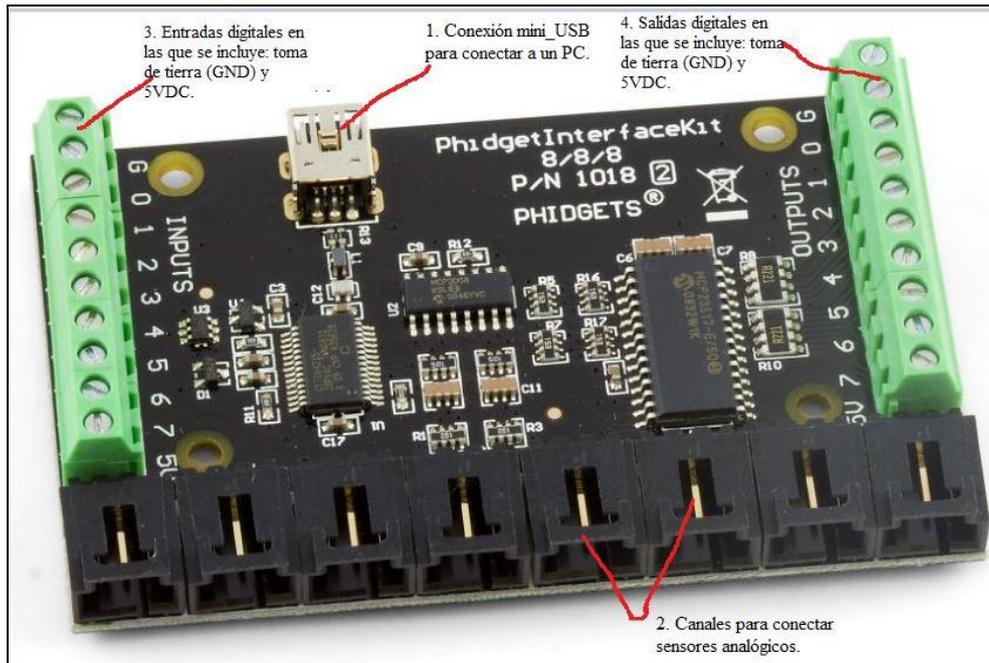


Figura 2-10: Placa del modelo 8/8/8 Phidget Interface Kit.

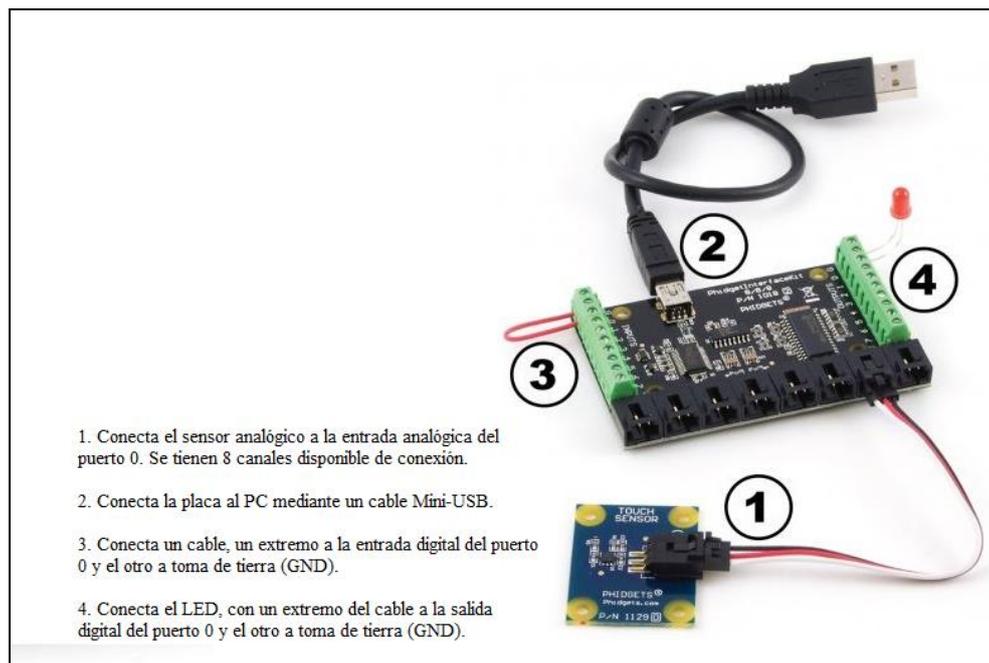


Figura 2-11: Ejemplo de conexión de un sensor analógico y un LED para la placa del modelo 8/8/8 Phidget Interface Kit.

El software permite gran variedad de sistemas operativos (Windows, Mac, Linux, Android, IOS) con distintas arquitecturas (32 bits, 64 bits); que se pueden bajar desde la página del fabricante de forma gratuita. Se pueden programar las acciones del robot en gran variedad de lenguajes de programación: C, C#, Java, LabVIEW, Python, Matlab, Visual Basic, Delphi.

2.2.3.1 Ventajas.

- a) En la página del fabricante existen foros de desarrolladores, videos demostrativos de montajes, y librerías adicionales para programar en cualquier lenguaje de programación y entorno.
- b) El fabricante proporciona distintos elementos como motores, sensores, etc. No es necesario emplear componentes electrónicos no oficiales.
- c) Funciona con gran variedad de lenguajes de programación.
- d) La instalación del software básico es sencilla.

2.2.3.2 Desventajas.

- a) Es necesario tener conocimientos básicos de electrónica digital y analógica, para conectar sensores y actuadores a la placa.
- b) Es necesario tener conocimientos sobre lenguajes de programación C/C++ y Python para poder enviar las acciones del robot.
- c) Es necesario tener conocimientos sobre distintos sistemas operativos para poder instalar el entorno de software.
- d) El precio es algo más elevado: en torno a 90 a 100 dólares para las placas de desarrollo.

2.2.4 Meccano.

Meccano es un sistema de construcción de modelos consistente en piezas de distintos tamaños, formas y colores construidas en metal con filas de barrenos (agujeros) para sujetarlas a otras piezas por medio de tornillos, orientado principalmente a niños. Dependiendo del modelo, puede contener motores eléctricos, ruedas, etc.

La idea de Meccano, fue concebida en 1898 por Hornby, que desarrollaron y patentaron el kit de construcción como “Mecánica fácil” en 1901. El nombre fue cambiado más adelante por “Meccano”, fabricado por la compañía británica Meccano Ltd, entre 1908 y 1980.

Actualmente se fabrica en Francia y China por Meccano SN de Francia, que forma parte de la empresa de juguetes Spin Master canadiense. En los EEUU, Meccano se vende bajo la marca Erector Set.



Figura 2-12: Logo de Meccano.

De cara al objetivo del proyecto sólo se va considerar el modelo Meccano Tech (Robot Personal Meccanoid G15, Robot personal Meccanoid G15Ks), lanzado a finales del 2015. En este modelo el robot es controlado por un microcontrolador al que se conectan motores y LEDs mediante módulos inteligentes.

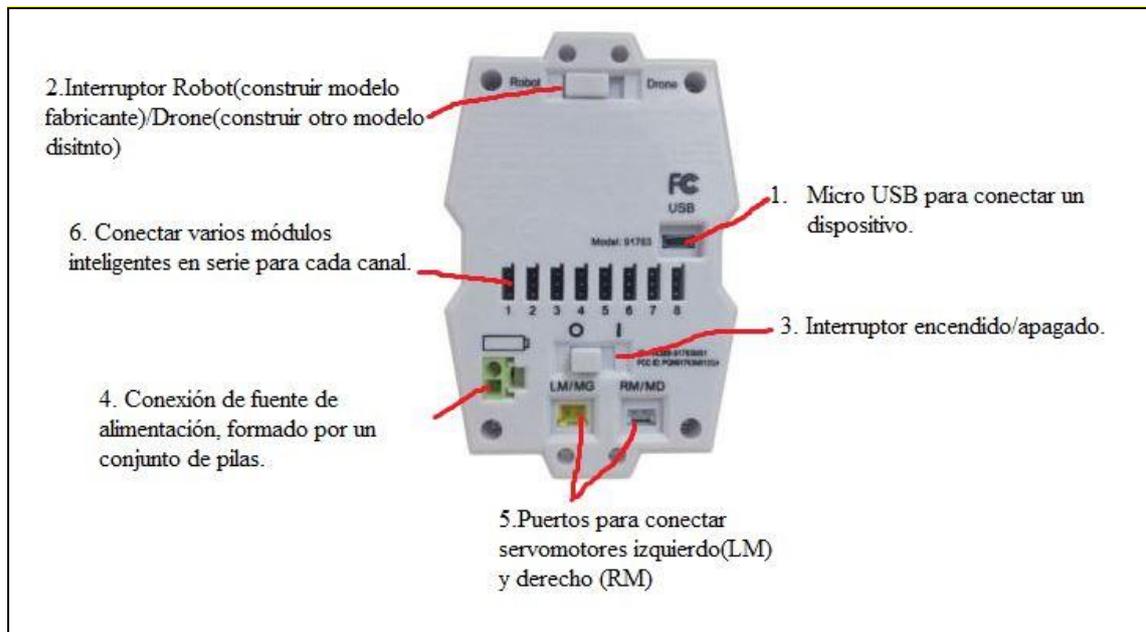


Figura 2-13: Microcontrolador MeccaBrain de Meccano.

Un módulo inteligente se refiere a un módulo electrónico capaz de comunicar datos bidireccionalmente a través de una sola línea de datos. Todos los módulos conectados son detectados; es decir, cuando se conecta un nuevo módulo el microcontrolador MeccaBrain le contesta. Existen dos tipos:

- i. Smart Servo, usado para realizar el movimiento de motores mediante la especificación de la posición angular; de forma que se puede girar ángulos específicos. También se le puede desenganchar el motor y sólo enviar los datos de su posición angular. Este módulo es la base para Meccanoid LIM (Learned Intelligent Movement), dónde el robot tomará los datos y el usuario puede grabar una animación en la memoria del movimiento las partes del robot. También disponen de un embrague mecánico y de detección de parada de corriente, para proteger el módulo de niveles nocivos de giro.
- ii. Smart LED, usado como indicador luminoso que dispone de dos entradas RGB (Red Green Blue) LEDs comunicándose con el microcontrolador

MeccaBrain. Este módulo sólo dispone de un conector de entrada de modo que no puede ser usado en modo serie como otros módulos; sólo se puede conectar con el extremo de la cadena.

Para comprender el funcionamiento de los módulos inteligentes con el microcontrolador se muestra un esquema de conexión:

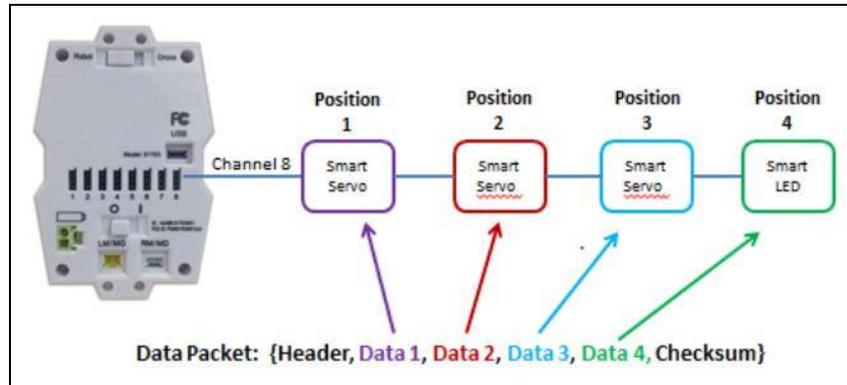


Figura 2-14: Esquema de conexión de varios módulos inteligentes al microcontrolador MeccaBrain de Meccano.

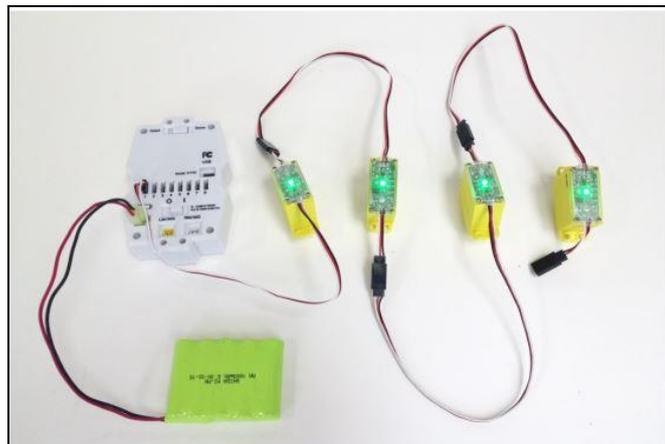


Figura 2-15: Conexión de varios módulos inteligentes al microcontrolador MeccaBrain de Meccano.

El software permite gran variedad de sistemas operativos (Windows, Mac) así como distintos dispositivos (PC, Smartphone, tablet), y se puede descargar de forma gratuita desde la página del fabricante. Al ser un modelo reciente, no existen otras maneras en las que se pueda programar; pero sí existen librerías adicionales para complementarlo con Arduino.

2.2.4.1 Ventajas.

- a) El entorno de desarrollo es amplio ya que existen dos maneras: el dado por el fabricante así como librerías adicionales para complementarlo con otras plataformas robóticas adicionales como Arduino.
- b) Para construir el modelo no es necesario buscar piezas adicionales, ya que el propio modelo ya las incluye.
- c) No es necesario tener conocimientos de electrónica digital así como de programación: cualquier usuario puede construir y diseñar el robot.
- d) Es fácil de usar: viene con un manual de instrucciones en el que se indica paso a paso cada construcción.
- e) Se puede usar para otro tipo de funcionamiento que no sea el mostrado en el manual de instrucciones.

2.2.4.2 Desventajas.

- a) Es difícil encontrar entornos de programación distintos del programa que da el fabricante, debido a que es un modelo recién lanzado. Esto causa una falta de flexibilidad.
- b) Las piezas que da el fabricante para poder construir el robot se unen con tornillos, que con el paso del tiempo suelen aflojarse causando mal funciones.
- c) El precio es relativamente más alto, entre 300 a 400 €; aunque el precio incluye varias piezas para la construcción: LEDs y servomotores (además del microcontrolador).
- d) No se venden piezas sueltas como motores, LEDs; en caso de necesitar alguna extra es necesario comprar otro kit completo.

2.2.5 LEGO Mindstorms EV3.

LEGO Mindstorms EV3 es la tercera generación del kit de robótica de la línea Mindstorms de LEGO, lanzado oficialmente para su venta a finales del 2013. En este proyecto se va a analizar la edición de educación, lanzada a mediados del 2013.



Figura 2-16: Logo de LEGO Mindstorms EV3.

En este modelo el robot es controlado por un microcontrolador, llamado ladrillo inteligente EV3, desde el que se controlan funciones como mover motores o leer sensores.

El hardware está basado en un microprocesador ARM9 y en el sistema operativo Linux, incluyendo un conector USB para PC, un micro USB para conectar a otros dispositivos, y una ranura para introducir un micro SD (32 GB) para almacenamiento extra. También dispone de Bluetooth y de Wifi para conectar. Se puede complementar con otras plataformas ya presentadas como Arduino o Raspberry pi.

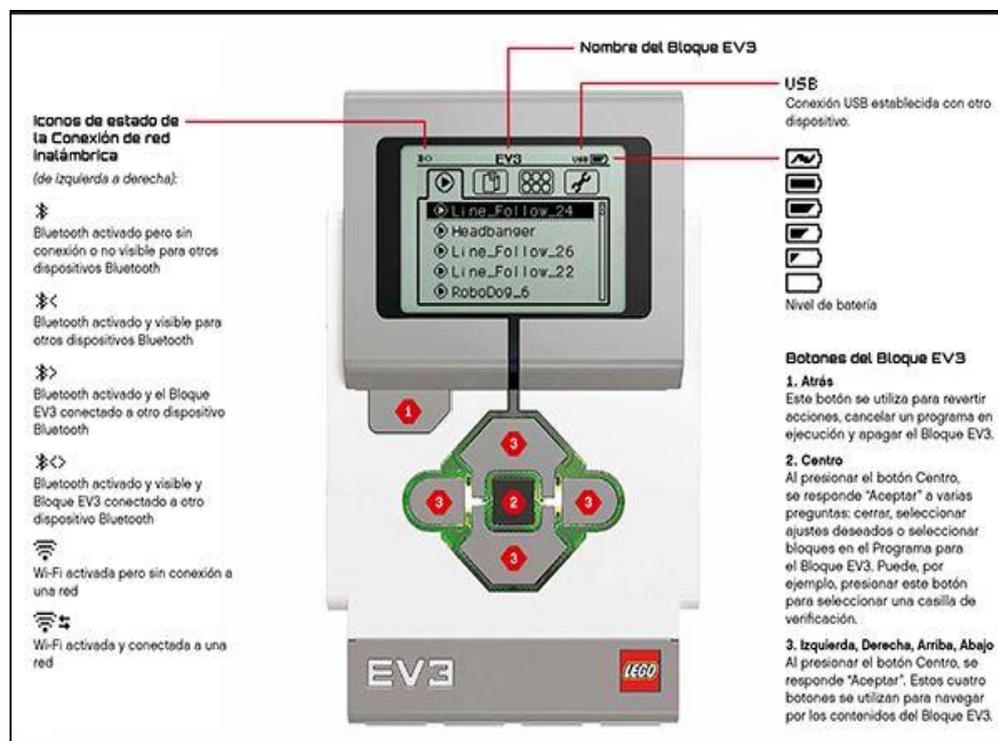


Figura 2-17: Ladrillo inteligente de LEGO Mindstorms EV3.

El entorno de programación viene dado por el fabricante y está basado en programación por bloques, válido para cualquier sistema operativo (Windows, Mac OSX) y se puede descargar de forma gratuita. También existen plataformas de programación open source que incluyen librerías para distintos lenguajes de programación: C, C++, C#, y Visual Basic.

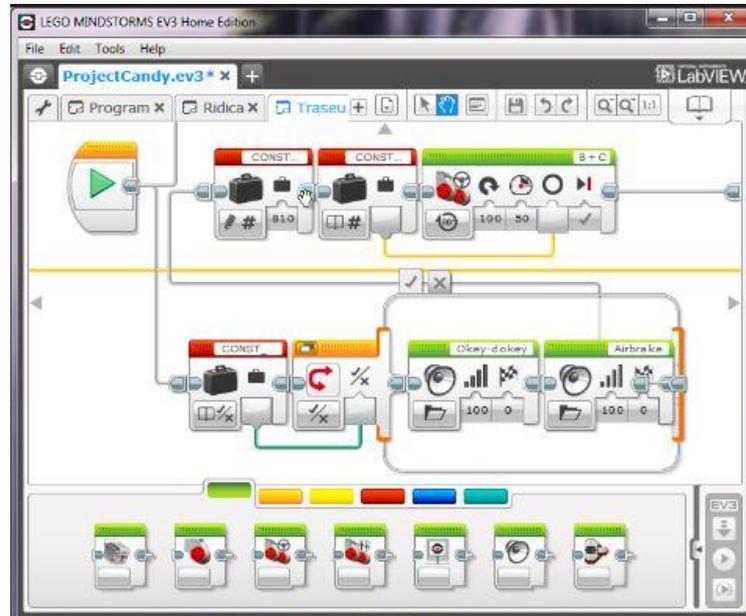


Figura 2-18: Entorno de programación dado por el fabricante para LEGO Mindstorms EV3.

2.2.5.1 Ventajas.

- a) Es fácil de usar ya que viene un manual de instrucciones en el que se indica paso a paso cada una de las construcciones. Sin embargo, es posible construir cualquier diseño propio con las piezas del fabricante.
- b) En Internet existen foros, videos demostrativos de montajes, librerías adicionales para programar en un entorno distinto al dado por el fabricante; y distintas plataformas open source con sus Apis, explicadas para su implementación.
- c) Los diseños son bastante sólidos y duraderos, debido a que las piezas se encajan perfectamente. No es necesario ajustarlas con el paso del tiempo.
- d) Es fácil de desmontar y montar, de tal manera que es fácil su transporte y ocupa poco espacio.
- e) Está orientado al gran público, no es necesario tener conocimientos técnicos para poder diseñar y construir el robot.

2.2.5.2 Desventajas.

- a) El precio es relativamente alto: entre 300 a 400 €. Sin embargo en este precio ya vienen incluidas varias piezas para la construcción, LEDs, servomotores, etc. (además del microcontrolador).
- b) La poca durabilidad de la batería o pilas, si se mantiene encendido siempre y con uso prolongado.
- c) Es relativamente frágil.
- d) El mecanismo de bluetooth presenta a veces interrupciones en la señal.

2.2.6 Fischertechnik.

Fischertechnik es una marca de juguetes de construcción, creada por Artur Fischer en 1965 y producida por fischertechnik GmbH en Alemania.

Fischertechnik se basa en sistemas de construcción de numerosos componentes que intentan imitar los elementos utilizados en máquinas y dispositivos reales, permitiendo que se pueda orientar a la enseñanza; tanto para enseñanza básica como superior.



Figura 2-19: Logo de Fischertechnik.

De cara al objetivo del proyecto sólo se va a analizar el modelo COMPUTING, controlado por un microcontrolador ROBO LT Controller.

El hardware está basado en un procesador de doble núcleo, con un procesador principal ARM Cortex A8 (32 bits/500 Mhz) y Cortex M3 empleando el sistema operativo Linux. Incluye una capacidad de memoria de 64 MB Flash y 128 MB DDR3 de RAM; con tarjeta SD para ampliación de memoria.

También dispone de Bluetooth, Wifi o entrada USB para conectar a cualquier dispositivo electrónico (Smartphone, PC, Tablet); o receptor infrarrojo, si se quiere conectar a algún dispositivo del fabricante como motor o sensor. Se puede complementar con otras plataformas ya presentadas como Arduino o Raspberry pi.



Figura 2-20: Microcontrolador ROBO LT Controller de Fischertechnik.

El entorno de programación ROBO PRO Software viene dado por el fabricante y está basado en programación por bloques, siendo válido para cualquier sistema operativo (Windows, Mac OSX). También existen plataformas de open source que incluyen varias librerías para distintos lenguajes de programación: C, Java, LabView.

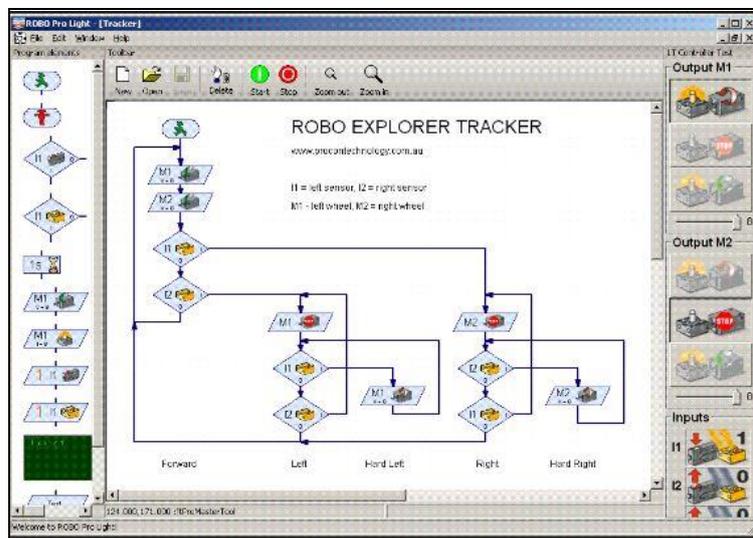


Figura 2-21: Entorno de programación dado por el fabricante ROBO PRO Software para Fischertechnik Robotic.

2.2.6.1 Ventajas.

- a) Es fácil de usar ya que viene un manual de instrucciones en el que se indica paso a paso cada una de las construcciones. Sin embargo, es posible construir cualquier diseño propio con las piezas del fabricante.
- b) En internet existen foros, videos demostrativos de montajes, y librerías adicionales para programar en entornos distintos al dado por el fabricante.
- f) Los diseños son bastante sólidos y duraderos, debido a que las piezas se encajan perfectamente. No es necesario ajustarlas con el paso del tiempo.
- g) Está orientado al gran público, no es necesario tener conocimientos técnicos para poder diseñar y construir el robot.

2.2.6.2 Desventajas.

- a) El precio es relativamente elevado. Por ejemplo el microcontrolador por separado cuesta en torno 230 euros.
- e) Es relativamente frágil.
- b) Es necesario comprar productos del fabricante para construir el robot, no es válido con otras piezas de otros fabricantes como LEGO.
- c) No se le pueden añadir otros dispositivos electrónicos (motores, sensores) que no sean los que vengan dados por el fabricante.

2.3 Conclusión y justificación de la plataforma robótica adecuada.

Plataformas presentadas	Precio	Conocimientos previos	Durabilidad de los diseños	Permite Piezas y componentes oficiales del fabricante.	Instalación De dispositivos.
Arduino	Asequible	Sí	No	Sí	Media
Raspberry pi	Asequible	Sí	No	Sí	Media
Phidgets	Elevado	Sí	No	Sí	Media
Meccano	Elevado	No	No	No	Rápida
Legó Mindstorm EV3	Elevado	No	Sí	No	Rápida
Fischertechnik	Elevado	No	Sí	No	Rápida

Tabla 2-2: Tabla comparativa de las distintas plataformas robóticas presentadas con las características de cara al objetivo al proyecto.

A la vista de la tabla comparativa, se sacan las siguientes conclusiones:

- i. Meccano, Legó Mindstorms EV3 o Fischertechnik son recomendables, porque el usuario no necesita conocimientos previos para poder diseñar el robot.
- ii. Meccano, Legó mindstrom EV3 o Fischertechnik son recomendables, porque la instalación de los componentes electrónicos es rápida.
- iii. Legó Mindstorms EV3 o Fischertechnik son recomendables, porque las piezas que se utilizan para construir el robot tienen mayor durabilidad en la construcción de sus diseños.
- iv. Meccano, Legó Mindstorms EV3 o Fischertechnik son recomendables, porque el usuario no necesita buscar piezas o dispositivos adicionales para poder diseñar el robot; debido a que el fabricante dispone de piezas adicionales frente a las otras plataformas robóticas.
- v. Arduino, Raspberry pi son recomendables, porque el precio no es tan elevado frente a otras plataformas robóticas.

De todo lo anterior se puede inferir que las plataformas más convenientes para diseñar y construir el robot para este proyecto son Legó Mindstorms EV3 o Fischertechnik. Para este proyecto, se ha elegido la plataforma Legó Mindstorms EV3, ya que posiblemente es un poco más conocida para el gran público.

3 Estudio de mercado del sistema robótico

3.1 Introducción.

En este capítulo, se realizará un estudio de mercado del sistema robótico enfocado al objetivo del proyecto.

Para ello, en la sección 3.1.1 se realizará una estimación de la demanda que puede tener el desarrollo propuesto, teniendo en cuenta el número de jugadores, tanto noveles como experimentados, en todo el mundo. A partir de estos números se estima el porcentaje de jugadores que podrían estar interesados en emplear este sistema; así como lo que estos jugadores podrían esperar del mismo.

Por último, en la sección 3.1.2 se analizará la oferta de otros sistemas robóticos que existen en el mercado, especializados en el problema de repartir manos preparadas de Contract Bridge.

3.1.1 Demanda.

Para estimar el número de jugadores se han empleado los datos facilitados por la Federación Mundial de Bridge (WBF: WORLD BRIDGE FEDERATION). Los jugadores reportados corresponden a aquellos que están federados y apuntados en un club de Bridge. Posiblemente habrá otro cierto número de jugadores no federados que podrían estar interesados en el robot propuesto. Estimemos que estos sean un 10% de los federados.

Zone	Organization	Area	Member Countries	Total Membership
1	 EBL European Bridge League	Europe	46	384,186
2	 ACBL American Contract Bridge League	North America	3	170,606
3	 CSB Confederacion Sudamericana de Bridge	South America	10	3,575
4	 BFAME Bridge Federation of Asia & the Middle East	Asia	11	7,307
5	 CAC Central American & Caribbean Bridge Federation	Central America	14	895
6	 APBF Asia Pacific Bridge Federation	Pacific Asia	12	59,242
7	 SPBF South Pacific Bridge Federation	South Pacific	4	51,420
8	 ABF African Bridge Federation	Africa	12	6,086
2015 information, unless noted with *			Total:	112 683,317

Figura 3-1: Estimación del número de jugadores en el mundo según la Organización Mundial de Bridge, con datos del año 2015.

Por tanto tendremos una estimación de 750.000 jugadores de Contract Bridge en el mundo.

Ahora querríamos estimar el número de profesores en el mundo. Para ello analizamos, por ejemplo, el número de profesores en el Reino Unido, donde sólo hay 45 profesores registrados. Si estimamos que puede haber otros tantos profesores no registrados, estimamos un total de 100 profesores en Reino Unido, para un conjunto de 38.000 jugadores federados. Esto daría una estimación de 0,2% de profesores respecto al número de jugadores federados. Si hay 680.000 jugadores federados en el mundo, podemos estimar un total de 1800 profesores de Contract Bridge en el mundo.

También interesaría estimar el número de jugadores noveles en el mundo. Si cada profesor tiene una media de 25 alumnos aprendiendo, esto da una estimación de 50.000 jugadores noveles en el mundo. Considerando que el ciclo de aprendizaje sea de unos dos años, esto significa que cada año se incorporan unos 25.000 jugadores noveles en todo el mundo.

De esos 25.000 jugadores noveles, es muy probable que un muy alto porcentaje esté interesado en un sistema de reparto como el que se presenta en este proyecto, si estuviera disponible a un precio razonable. Estimemos en un 80% esta fracción (20.000 jugadores). A pesar de que el sistema que se propone en este Proyecto está diseñado para el gran público, empleando un sistema de construcción extremadamente sencillo (LEGO), sin embargo no todos estos jugadores (o personas de su entorno) tendrán la capacidad de montar el robot, cargar el software en su portátil y emplearlo para distribuir manos. Estimemos que sólo un 25% de los jugadores sería capaz (ellos o alguien de su entorno) de construir y conectar el robot. Eso da un total de 5.000 ventas anuales estimadas.

A las anteriores ventas en estado “estacionario” habría que añadir las ventas iniciales sobre el total de jugadores que no disponen hoy en día de un sistema de reparto automatizado. Si del total de 750.000 jugadores actuales estimamos que al menos un 1% podrían estar interesados en un sistema de reparto robótico a precio asequible, eso da un total de 75.000 personas interesadas. Estimando de nuevo que sólo un 25% de ellos tienen (ellos o su entorno) la capacidad de construir y operar un robot LEGO, eso da un número inicial de ventas de 20.000 sistemas.

3.1.2 Oferta.

La oferta de sistemas robóticos especializados para repartir cartas de Bridge es muy variada, según a qué usuarios va enfocada:

- a) Sistemas robóticos para clubes.

Estos sistemas robóticos dependen del número de integrantes de cada club. Debido a que a lo largo del día tienen que preparar muchas manos y de manera muy fiable, estos sistemas son un poco más caros, teniendo un precio entre 3000 y 4000 dólares.

De los sistemas más usados son:

- i. PlayBridgeDealer4+: Permite en meter las cartas que mediante un display en el que se le indica qué manos se quieren para jugador y automáticamente el sistema las reparte en una bandeja. Este sistema tiene un precio de 4000 dólares.



Figura 3-2: Sistema robótico PlayBridgeDealer4+.

- ii. Duplimate, su funcionamiento es exactamente igual que PlayBridgeDealer4+; pero menos sofisticado. Este sistema tiene un precio entre de 2200 libras.



Figura 3-3: Sistema robótico Duplimate.

b) Sistemas robóticos para particulares.

Aunque hay referencias en Internet a sistemas robóticos que en teoría deberían ser menos costosos que los usados en clubs, debido a que están pensados para un uso particular con menor velocidad de reparto y fiabilidad sin embargo no hemos encontrado dónde poder comprarlos (y por tanto tampoco hemos encontrado precios). Posiblemente son sistemas todavía no en el mercado, que podrían ser competidores del que se propone en este Proyecto.

Un ejemplo es el “*Bridge Port Automatic Playing Card*”, que se muestra en la siguiente figura.

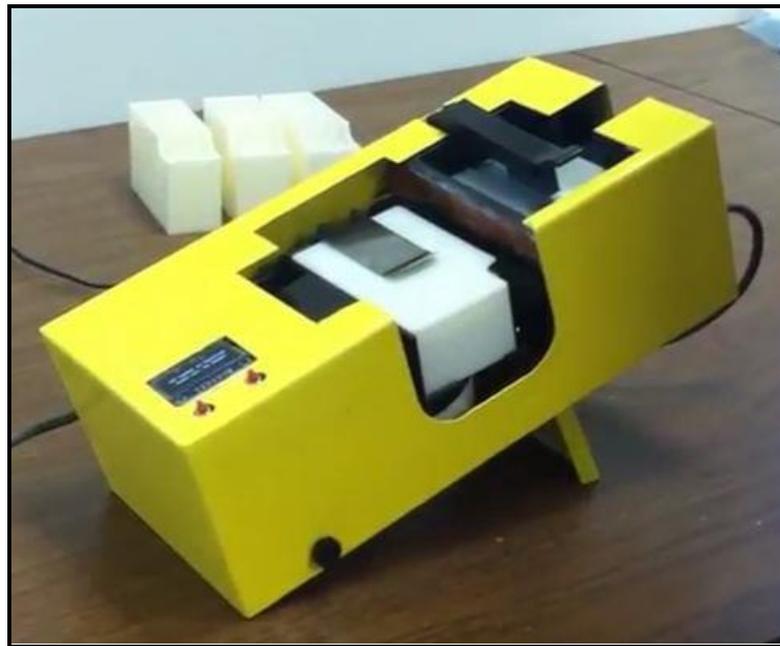


Figura 3-4: Sistema robótico para uso particular: Bridge Port Automatic Playing Card.

También existen otros sistemas robóticos diseñados por usuarios con piezas de otros fabricantes, aunque tampoco se encuentra información en Internet de cómo construirlos o comprarlos. Un ejemplo es el que se muestra en la siguiente figura.



Figura 3-5: Sistema robótico para uso particular, diseñado por un usuario con piezas de otros fabricantes.

c) Sistemas robóticos “do it yourself”.

Estos sistemas robóticos son los que menor coste tienen, con respecto a los anteriormente explicados. Se orientan a usuarios con distintos niveles de conocimiento técnico, utilizando las diferentes plataformas robóticas explicadas en el capítulo 2. En general sólo se pueden encontrar fotografías o videos de los sistemas funcionando, no instrucciones de cómo montarlos, y aún menos el software que se tiene que emplear.

En algunos de ellos el diseñador ha utilizado diferentes piezas de cada fabricante o ha realizado ingeniería inversa para adecuarlo a la construcción de estos sistemas robóticos. A continuación se muestran algunos ejemplos:

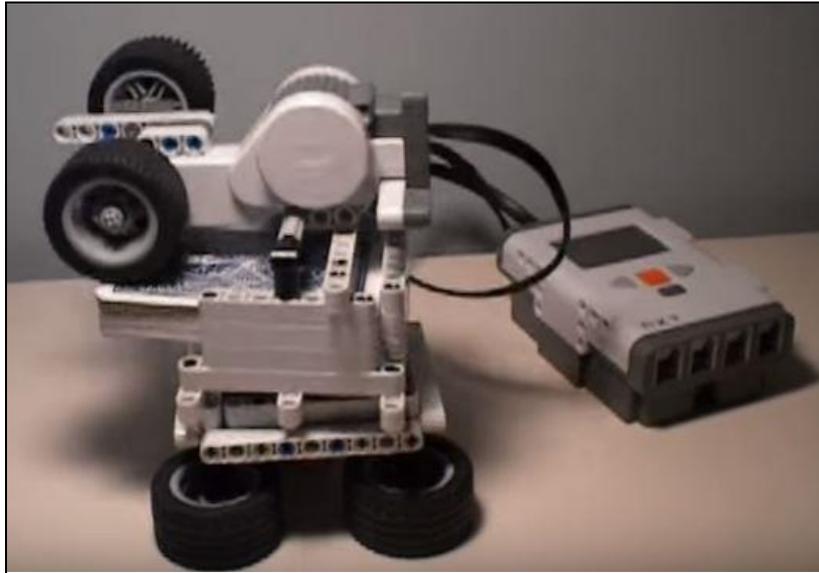


Figura 3-6: Sistema robótico "do it yourself" con la plataforma LEGO Mindstorms.

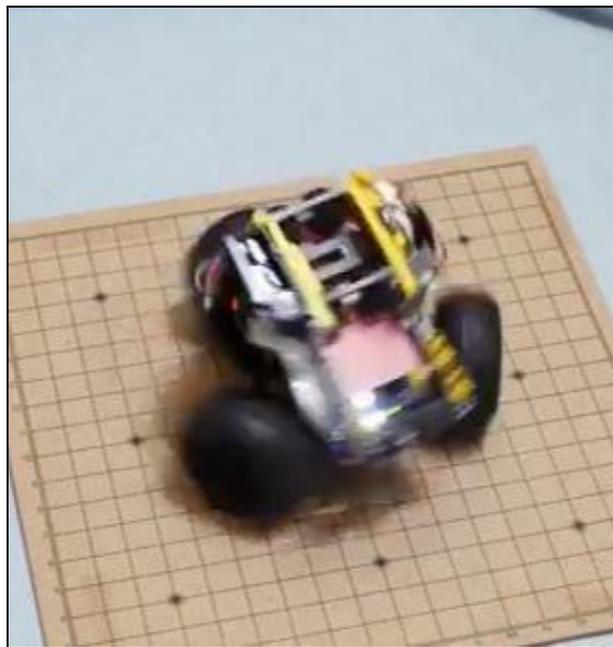


Figura 3-7: Sistema robótico "do it yourself" con la plataforma Raspberry Pi con piezas de LEGO Mindstorms.

4 Desarrollo e implementación del sistema robótico.

En este capítulo, se explicará el desarrollo e implementación del sistema robótico. Para ello, se presentarán dos modelos distintos que comparten el sistema de lanzamiento de cartas. A su vez se experimentó con varias versiones de este mecanismo de lanzamiento de cartas, que es una de las partes más sensibles a fallos del sistema.

En la sección 4.1, se presentarán una serie de algoritmos para el sistema de reparto y lanzamiento de cartas, explicando su diseño y funcionamiento y cuyos resultados se presentarán en el capítulo 5.

Posteriormente en las secciones 4.2 y 4.3 se explicará la construcción y diseño de cada modelo, mostrando la evolución de cada modelo a lo largo del desarrollo del proyecto para conseguir un reparto óptimo de cartas.

4.1 Algoritmos de funcionamiento.

Para agilizar el entendimiento del funcionamiento para repartir cartas, se explicarán todos los algoritmos con sus diagramas de flujo.

Primero, se explicará el algoritmo básico de funcionamiento del reparto de cartas, y a partir de éste se explican las sucesivas evoluciones realizadas para conseguir un reparto más robusto.

4.1.1 Algoritmo básico de funcionamiento.

El algoritmo básico de funcionamiento sigue los siguientes pasos:

- i. Lee de un fichero (formato .txt) la carta correspondiente y a que jugador le corresponde. El formato, se explicará en el Anexo B (guía del programador).
- ii. Mueve primer el motor del robot; empleando un número prefijado de grados a mover y una velocidad angular prefijada, llegando a la posición indicado por el fichero dado explicado anteriormente.
- iii. Una vez que haya detectado la posición, reparte la carta.
- iv. Si el robot ha finalizado en repartir las cartas, volverá a la posición inicial y restaura sus valores; por si se tuviese que ejecutar en otro momento. Si no ha terminado de repartir indicado por el fichero dado, volverá al punto i y continuará hasta que haya finalizado.

Se muestra un diagrama de flujo para comprender mejor su funcionamiento, de los pasos explicados anteriormente:

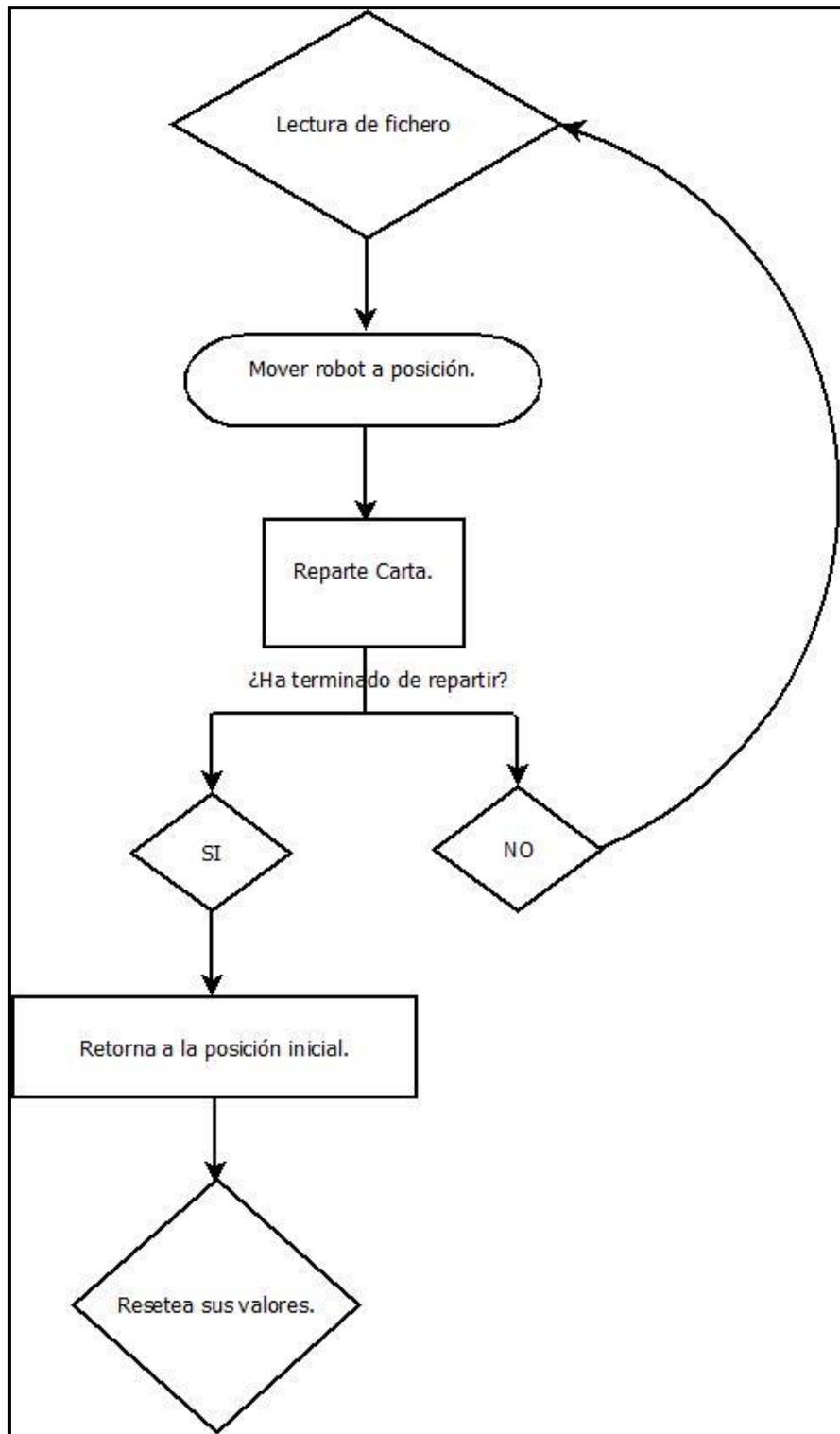


Figura 4-1: Algoritmo básico de funcionamiento.

4.1.2 Algoritmo A.

El objetivo principal es conseguir que se reparta exactamente una carta del mazo en cada posición. Uno de los problemas más recurrentes en los sistemas de reparto de naipes es que, dependiendo del estado de los mismos (nuevos, seminuevos, muy usados), puede ocurrir que se repartan (de manera errónea) dos naipes, o ningún naipe, en una posición.

Se explica su funcionamiento:

- i. Lee de un fichero (formato .txt) la carta correspondiente y a que jugador le corresponde. El formato, se explicará en el Anexo B (B.1 Formato de fichero de texto para el reparto de manos.).
- ii. Mueve el motor del robot, empleando un número prefijado de grados a mover y velocidad angular indicada, a la posición dónde va a repartir la carta, indicado por el fichero dado explicado anteriormente.
- iii. Una vez que haya detectado la posición, reparte la carta empleando un número prefijado de grados a mover y velocidad angular.
- iv. Retrocede el motor, empleando un número prefijado de grados a mover y velocidad angular, para hacer retroceder a una posible segunda carta.
- v. Retrocede el motor por segunda vez, con otros parámetros que en el paso iv.
- vi. Si el robot ha finalizado en repartir las cartas, volverá a la posición inicial y restaura sus valores; por si tuviese que ejecutar en otro momento. Si no ha terminado de repartir indicado por el fichero dado, volverá al punto i y continuará hasta que haya finalizado.

Se muestra un diagrama de flujo para comprender mejor su funcionamiento, de los pasos explicados anteriormente:

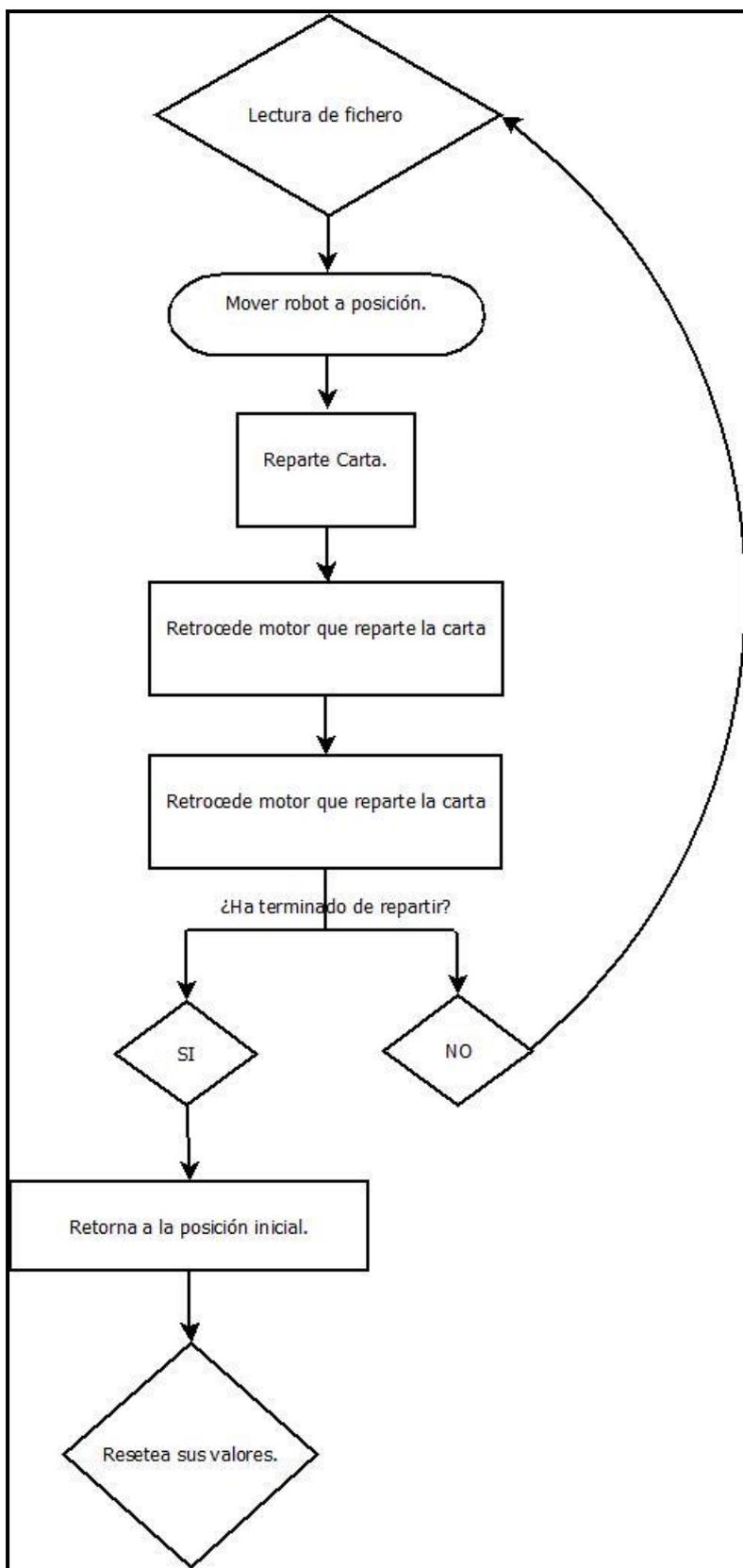


Figura 4-2: Algoritmo A.

4.1.3 Algoritmo B.

Éste es la modificación del anterior, para evitar los problemas que se explican en el capítulo 5.

Se explica su funcionamiento mediante los siguientes pasos:

- i. Lee de un fichero (formato .txt) la carta correspondiente y a que jugador le corresponde. El formato, se explicará en el Anexo B (B.1 Formato de fichero de texto para el reparto de manos.).
- ii. Mueve el motor del robot; tantos grados como la velocidad indicada, a la posición dónde va a repartir la carta, indicado por el fichero dado explicado anteriormente.
- iii. Una vez que haya detectado la posición, reparte según los grados dependiendo de la carta que haya repartido, que se explicará en el capítulo 5.
- iv. Retrocede el motor, tantos grados como la velocidad que se le haya indicado, que se encarga de repartir la carta, para evitar que ningún jugador tenga más cartas que otros.
- v. Si el robot ha finalizado en repartir las cartas, volverá a la posición inicial y restaura sus valores; por si tuviese que ejecutar en otro momento. Si no ha terminado de repartir indicado por el fichero dado, volverá al punto i y continuará hasta que haya finalizado.

Se muestra un diagrama de flujo para comprender mejor su funcionamiento, de los pasos explicados anteriormente:

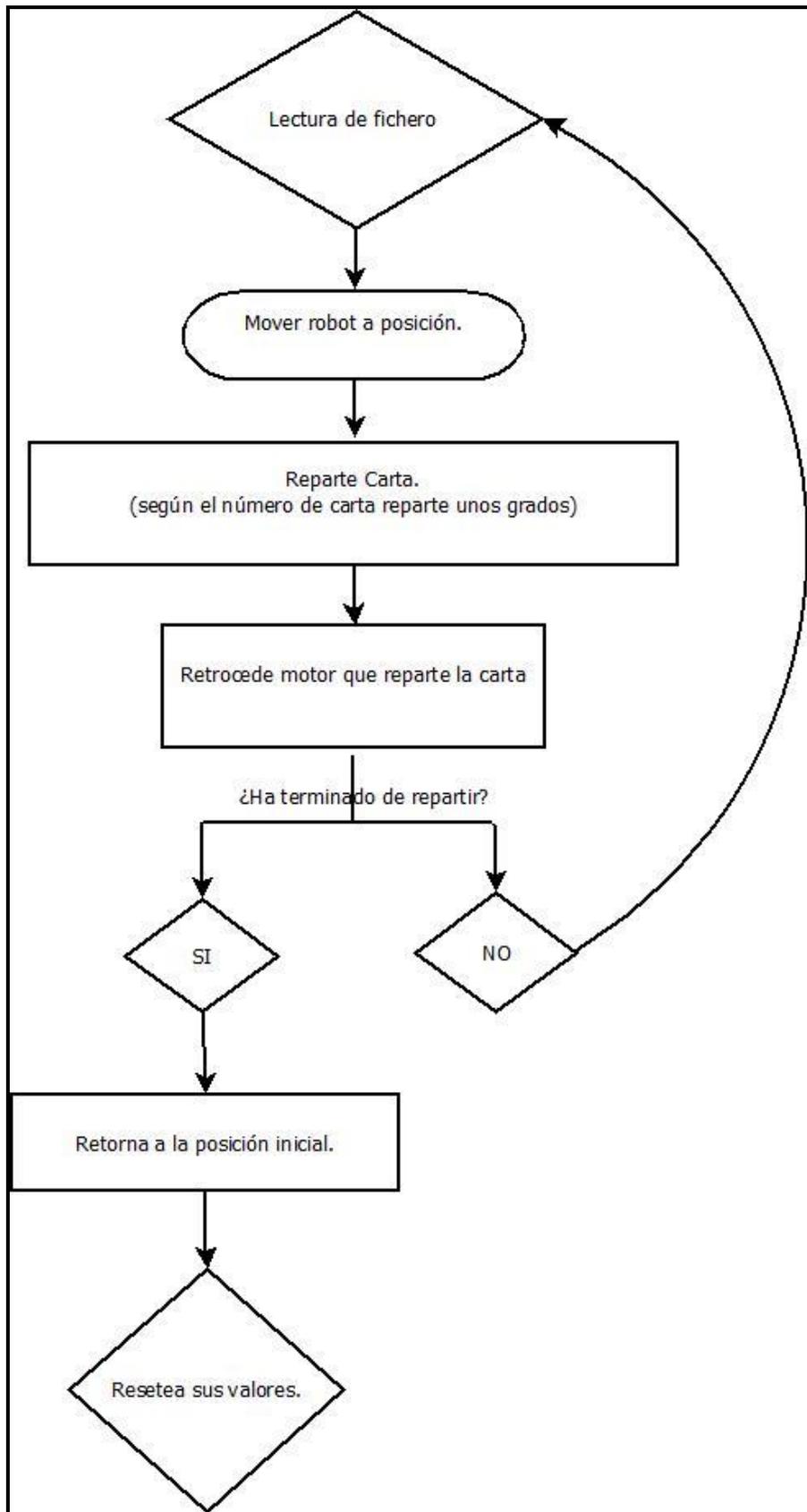


Figura 4-3: Algoritmo B.

4.2 Diseño y construcción.

Para la construcción de ambos modelos, se han empleado los siguientes kits:

- i. Caja del modelo 4544 de LEGO Mindstorm education EV3 en las que se incluye: ladrillo inteligente de EV3, tres servomotores interactivos, sensor de color, sensor giroscópico, sensor ultrasónico y dos sensores de contacto, batería recargable, rueda de bola, cables de conexión, instrucciones de montaje, piezas LEGO Technic para crear una gran variedad de modelos.



Figura 4-4: Caja del modelo 4544 LEGO Mindstorm EV3.

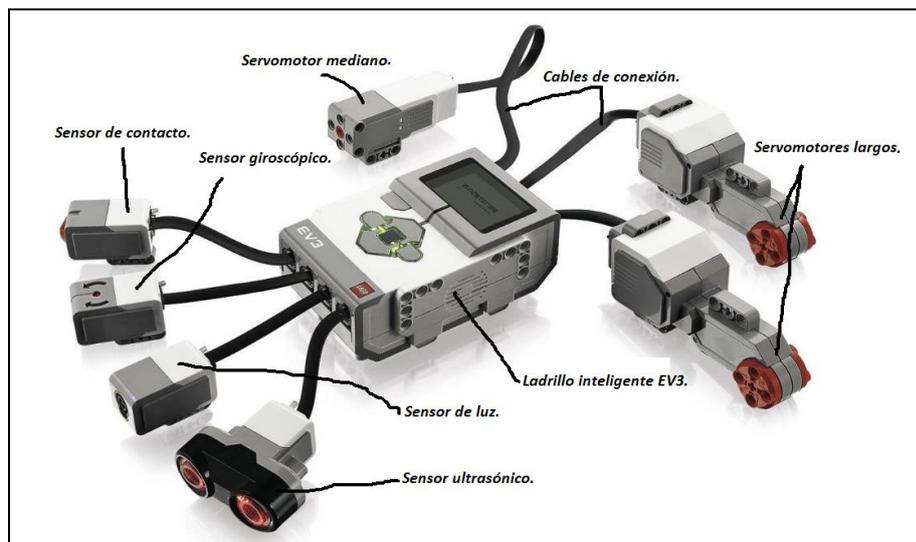


Figura 4-5: Componentes de la caja del modelo 4544 LEGO.

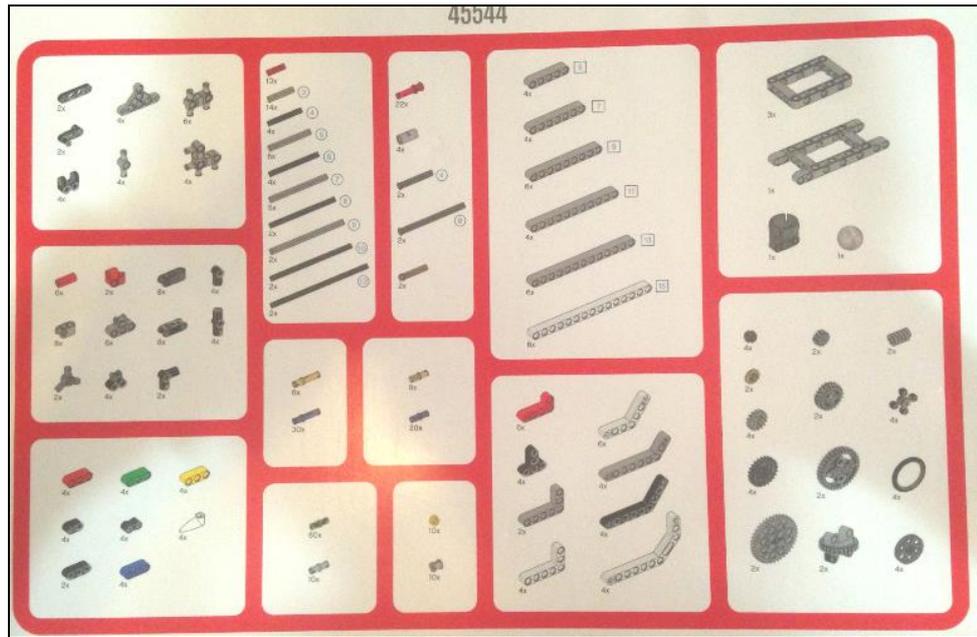


Figura 4-6: Piezas de la caja del modelo 4544 LEGO.

- ii. Caja adicional de piezas del modelo 31046 de LEGO, para poder construir y diseñar ambos modelos.



Figura 4-7: Caja del modelo 31046 LEGO.

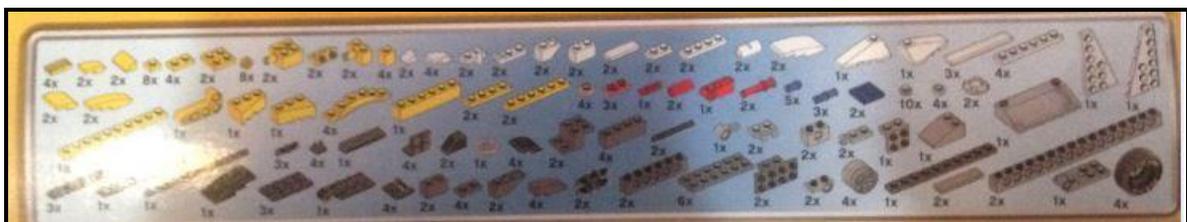


Figura 4-8: Piezas de la caja del modelo 31046 LEGO.

Para la construcción de ambos modelos, se han tomado para estos casos los siguientes elementos de los kits:

- a) 2 servomotores largos del modelo 4544 de LEGO.
- b) Ladrillo inteligente EV3, que se le conectará mediante USB al PC para programar sus órdenes de funcionamiento, tal como se explica en el manual de programación en el Anexo B. Viene incluido en la caja del modelo 4544 de LEGO Mindstorms EV3.
- c) 2 cables de conexión para conectar los dos servomotores al Ladrillo inteligente EV3, y que viene incluido en la caja del modelo 4544 de LEGO Mindstorms EV3.
- d) Piezas del kit de la caja del modelo 4544 y 31046 de LEGO.

Para entender mejor el diseño y construcción de ambos modelos, se adjuntarán junto con la memoria videos de explicación para su construcción. En ambos modelos, se parte de uno inicial en el que se ha ido realizando una serie de mejoras hasta obtener el óptimo.

4.2.1 Modelo 1.

El objetivo de este modelo es repartir cada carta a un jugador concreto, mediante un sistema de rotación para mover el brazo de reparto de cartas a la posición de cada jugador.

Para ello, son necesarias:

- i. Bandeja para colocar las cartas, usando las piezas del kit de 4544 para conseguir que entren las cartas con su tamaño correspondiente.
- ii. Sistema de rotación para el reparto de las cartas a cada jugador, necesitando un servomotor largo del kit 4544 junto con las piezas de este kit.
- iii. Rodillo parecido a las máquinas de imprenta, colocado encima de las cartas para su reparto a cada jugador, necesitando otro servomotor largo del kit 4544 para que realice esta función junto con piezas de este kit.
- iv. Soporte sobre el que va montado todo el sistema robótico: bandeja de reparto, sistema de rotación y rodillo para el reparto de cartas, utilizando piezas de los kits 4544 y 31046 para su construcción.

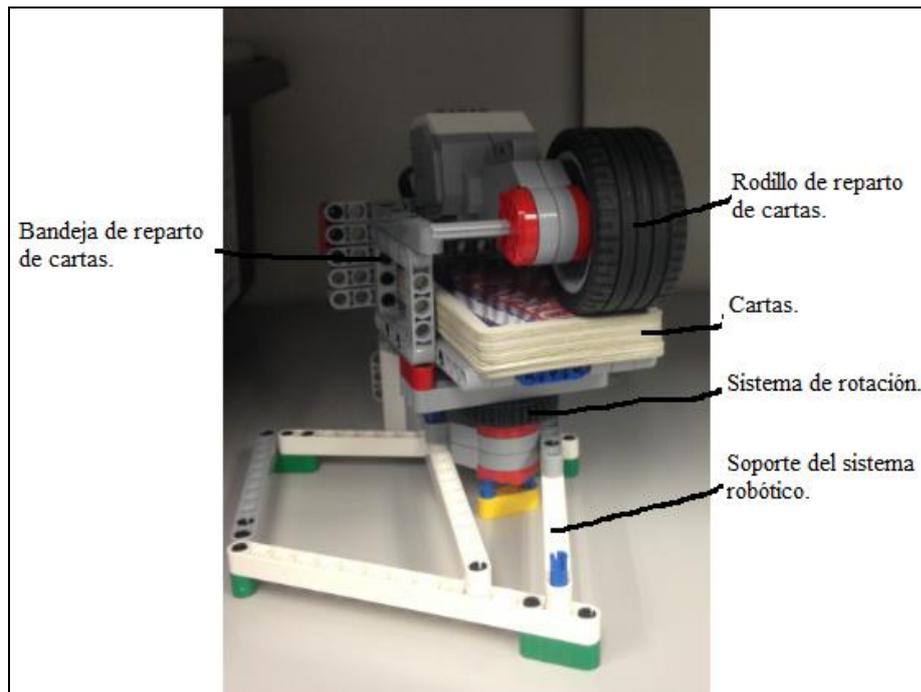


Figura 4-9: Modelo inicial con piezas del kit 4544 de LEGO.

Una vez construido, se realizaron unas pruebas iniciales para saber si su funcionamiento era el correcto, obteniéndose estas conclusiones:

- i. El sistema de rotación no fue estable cuando giró y repartió las cartas, haciendo que se caigan algunas cartas cuando gira.
- ii. El rodillo para repartir las cartas no permitía repartir la totalidad de las cartas, debido a que la rueda de éste sometía demasiada presión. Se juzgó necesario levantar el motor y hacer que la bandeja de reparto sea un poco más grande.
- iii. El soporte del sistema robótico no fue estable: a veces caían todas las cartas antes de empezar a repartir, al repartir la carta al jugador correspondiente o al girar al jugador correspondiente.
- iv. Las cartas al repartir a veces caían boca arriba, por lo que fue necesario añadir al final de la bandeja una rampa para que las cartas nunca cayeran boca arriba.

Con las conclusiones del modelo inicial, se realizaron una serie de mejoras:

- a. Para el soporte del sistema robótico, se ideó un sistema fijo compacto para sujetar la bandeja y el sistema de rotación, evitando así los problemas observados en el modelo inicial. Se utilizaron las siguientes construcciones:

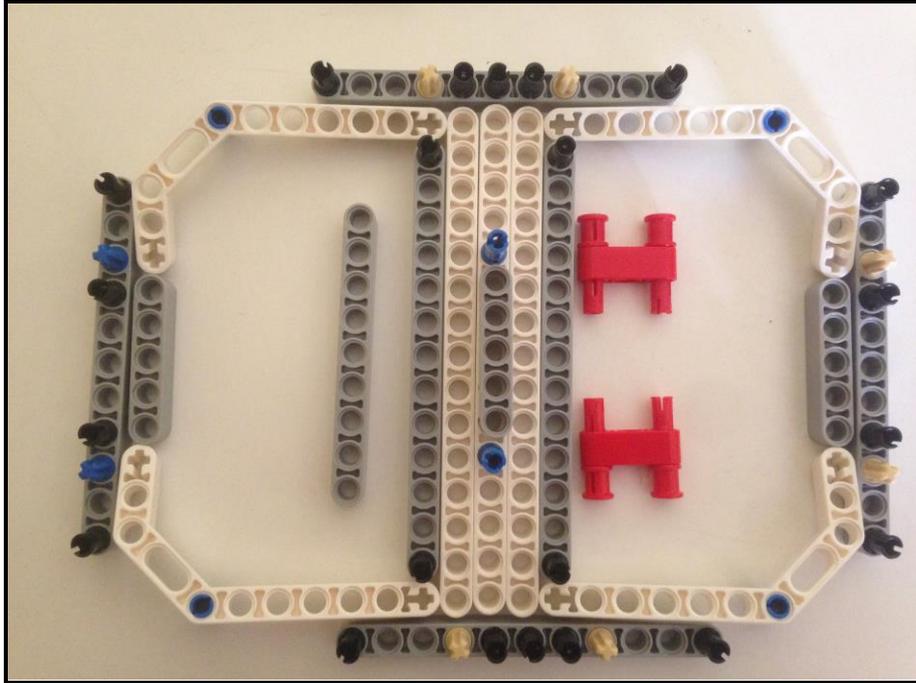


Figura 4-10: Piezas usadas para el ensamblar el soporte del sistema robótico del kit 4544 de LEGO; junto con cuatro ruedas del kit 31046 de LEGO.

Sobre las piezas de la Figura 4-8, se han colocado un conjunto de piezas ensambladas encima para construir el sistema de rotación. Esta parte del sistema robótico, es necesario que gire en conjunto con la bandeja de las cartas para conseguir que reparta las cartas a cada jugador.

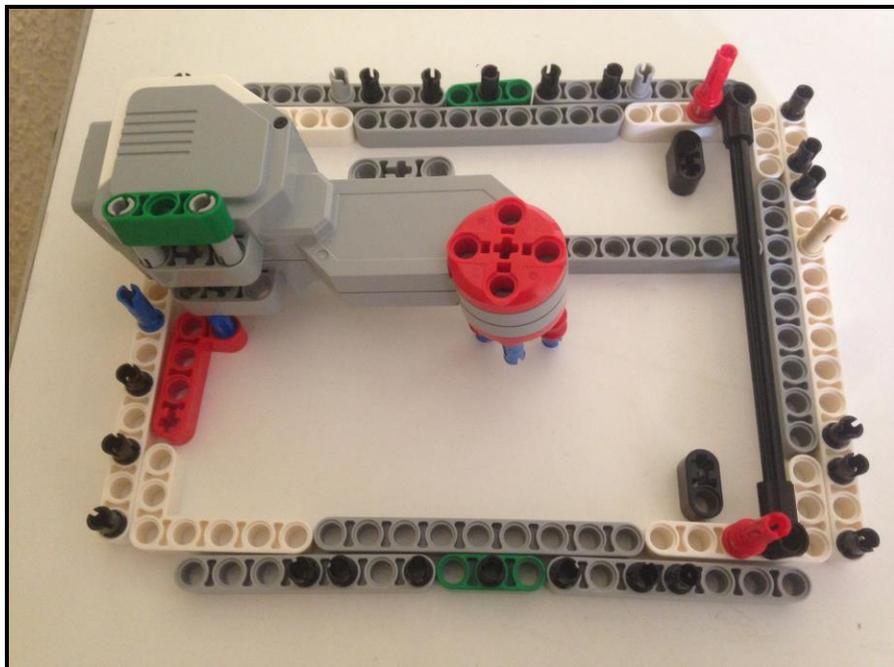


Figura 4-11: Piezas usadas para ensamblar el sistema de rotación junto con un servomotor largo, correspondientes al kit 4544 de LEGO.

- b. Se ha creado una rampa al final de la bandeja de reparto para evitar que las cartas caigan boca arriba; irá ensamblada a la bandeja de reparto. Para ello se han usado las piezas del kit 4544 de Lego.

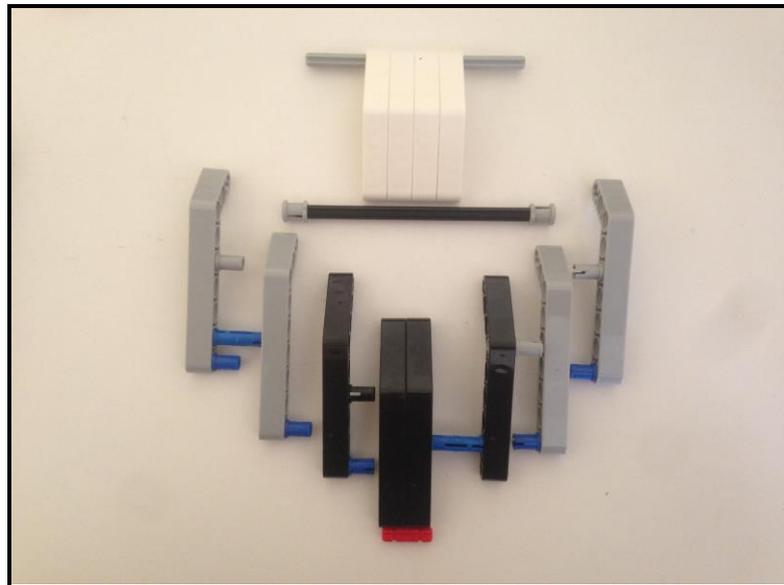


Figura 4-12: Piezas usadas para rampa con el kit 4544 de LEGO.

- c. Se ha ampliado la bandeja de reparto, para evitar que las cartas se caigan cuando se reparten a un jugador; y así elevar el rodillo de reparto de las cartas para solucionar el problema del modelo inicial que no podía repartir las cartas finales.

A esta bandeja de reparto, se le conecta el Ladrillo inteligente EV3 para contrarrestar el peso de las cartas y la bandeja de reparto; evitando que gire el sistema de rotación de forma inestable.



Figura 4-13: Piezas para ensamblar la bandeja de reparto y el Ladrillo inteligente EV3 con el kit 4544 LEGO

Una vez ensambladas la bandeja de reparto y el Ladrillo inteligente EV3, estas se conectan a la rampa; y todo ello al soporte del sistema robótico.

- d. Se ha elevado el rodillo para repartir las cartas, usando ruedas más pequeñas que las empleadas del modelo inicial. El motivo de estas modificaciones buscan una mayor precisión al dirigir cada carta al jugador correspondiente.

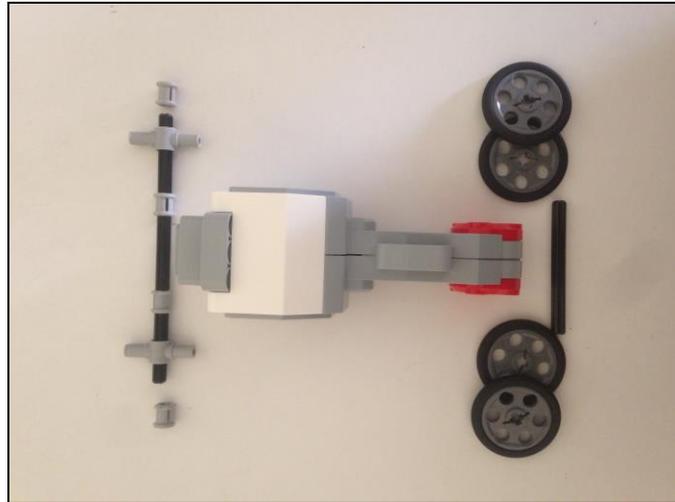


Figura 4-14: Rodillo de reparto de cartas usando un servomotor largo con el kit 4544 de LEGO.

El rodillo de reparto de cartas se coloca encima de la bandeja de reparto, dejando espacio para la conexión al Ladrillo inteligente. Este rodillo debe estar centrado para que pueda deslizarse y repartir correctamente las cartas.

Una vez construido, se realizaron una serie de pruebas de forma equivalente a las realizadas con el modelo inicial, obteniéndose las siguientes conclusiones:

- i. El sistema de rotación funcionó mejor que el modelo inicial, debido a que la estructura es más robusta, evitando que las cartas se caigan al rotar.
- ii. El rodillo de reparto de cartas funcionó mejor que el modelo inicial, pero al realizar las pruebas muchas veces repartía más de una carta a un jugador, cuando el objetivo es que reparta una carta a cada jugador cada vez que gira el sistema de rotación.
- iii. La bandeja de reparto de cartas, funcionó mejor que el modelo inicial. Pero al realizar las pruebas, había un problema con las últimas cartas del mazo.

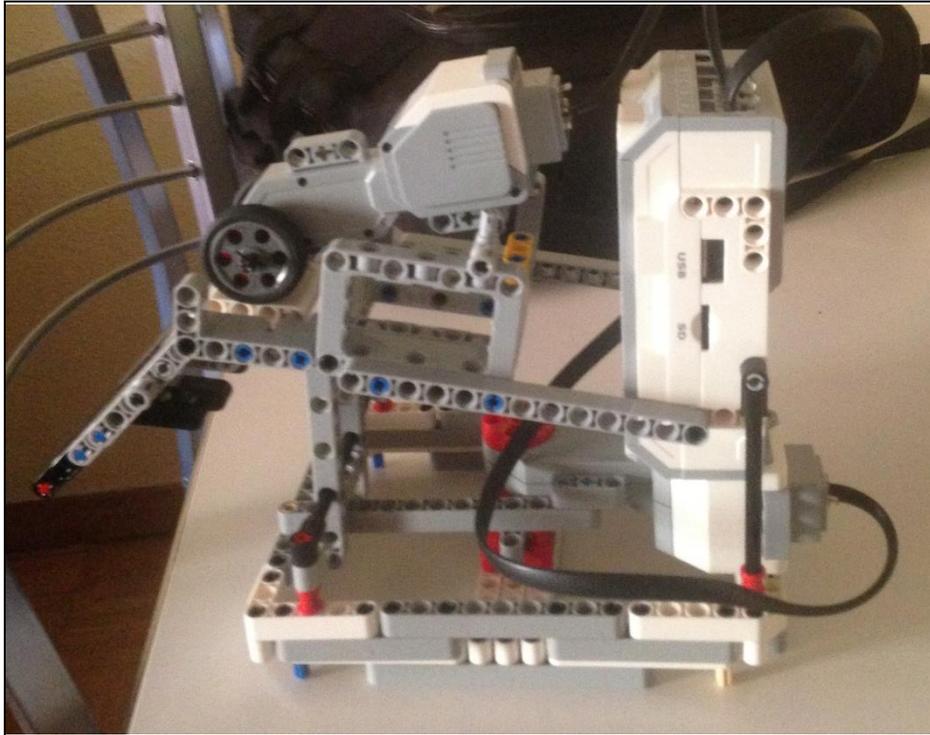


Figura 4-15: Modelo inicial 2 correspondiente al modelo 1 con piezas de los kits 4544 y 31046 de LEGO.

Con las conclusiones del modelo inicial 2, se realizaron una serie de mejoras partiendo de este modelo:

- a. Se elevó la bandeja de reparto de cartas, para evitar que repartiera más de una carta a cada jugador. Esta mejora, también ha permitido repartir las últimas cartas colocadas en la bandeja de reparto a cada jugador.

Entre la conexión de la bandeja de reparto de cartas y la rampa, se colocó un obstáculo para forzar a que se repartan la cartas una por una, evitando que repartieran dos cartas de una vez, como sucedía en el modelo 2.

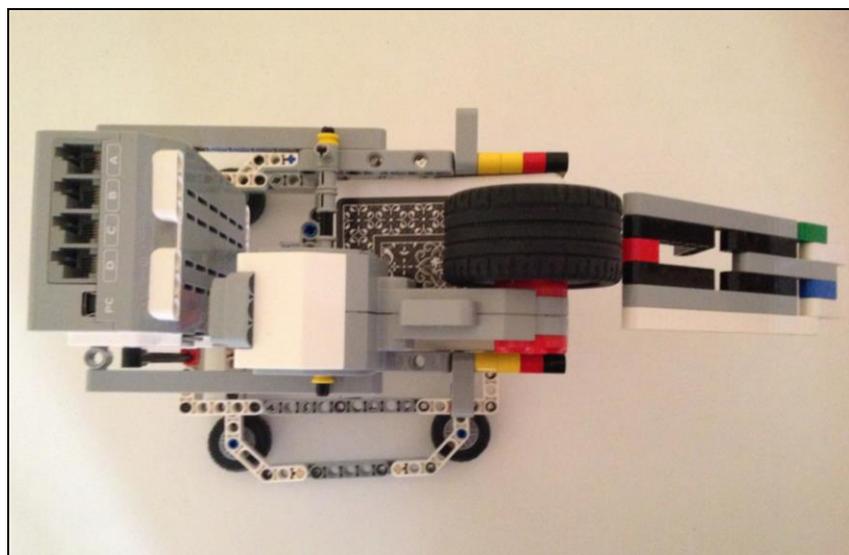


Figura 4-16: Bandeja de reparto de cartas mejorada con piezas de los kits 4544 de LEGO.

- b. Se ha mejorado la rampa de reparto de cartas, para que la carta no salga boca arriba cuando se reparte a cada jugador.

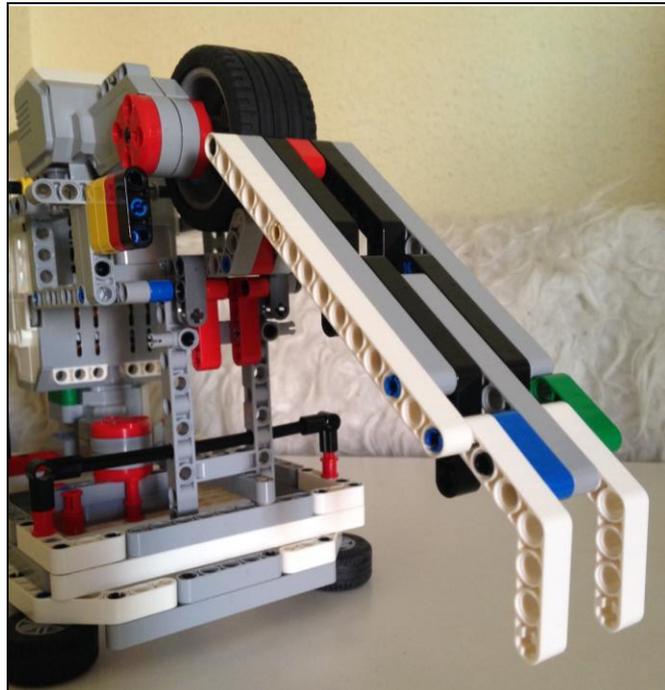


Figura 4-17: Rampa mejorada con piezas de los kits 4544 de LEGO.

La construcción de este modelo es exactamente que el modelo inicial 2, salvo por estas dos mejoras. Tras realizar una serie de pruebas iniciales, se obtuvieron las siguientes conclusiones:

- i. Todos los elementos que conforman este modelo, funcionan exactamente igual que el modelo inicial 2, salvo el rodillo que reparte la carta, que se ha sustituido por una rueda grande que debe estar centrada en la carta; debido a que esta rueda tiene que girar menos que una rueda pequeña, y funciona mejor con el obstáculo que se conecta la rampa con la bandeja de reparto.
- ii. Este modelo, si se usa con los parámetros adecuados, está libre de fallos dentro de las especificaciones que se han fijado: que en 20 repartos completos de manos de Bridge no hubiera ningún error.

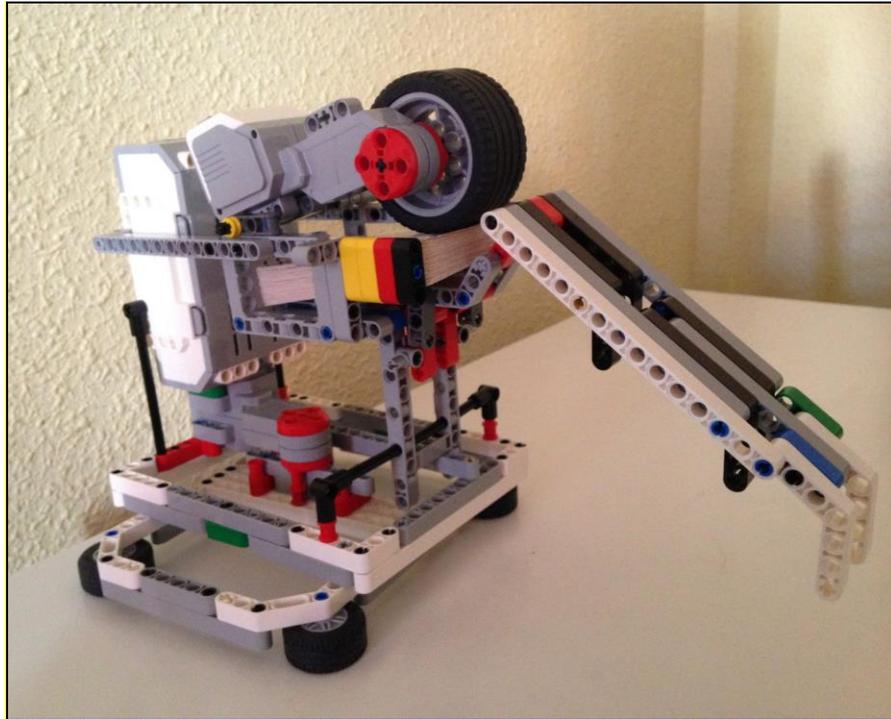


Figura 4-18: Modelo óptimo correspondiente al modelo 1 con piezas de los kits 4544 y 31046 de LEGO.

4.2.2 Modelo 2.

Así como el modelo 1 reparte las cartas a los jugadores de forma “posicional”, es decir, como haría un repartidor humano que lanza las cartas en las cuatro direcciones de los jugadores, el modelo 2 en cambio reparte las cuatro manos en una única bandeja con cuatro ranuras para acoger cada una de las manos de los cuatro jugadores. De esta forma, tras repartir una carta la bandeja sólo tiene que moverse a otro espacio para repartir la carta a otro jugador, por lo que se podría esperar mayores velocidades de reparto.

Para este modelos se han empleado los siguientes elementos:

- i. Bandeja para colocar las cartas, usando las piezas del kit de 4544 para conseguir que entren las cartas con su tamaño correspondiente.
- ii. Bandeja dónde se reparten las cartas de cada jugador, las piezas del kit de 4544 para conseguir que entren las cartas con su tamaño correspondiente.
- iii. Rodillo parecido a las máquinas de imprenta, colocado encima de las cartas para su reparto a cada jugador, necesitando otro servomotor largo del kit 4544 para que realice esta función junto con piezas de este kit.
- iv. Soporte sobre el que va montado todo el sistema robótico: bandeja de reparto, bandeja dónde se reparten las cartas y rodillo para el reparto de cartas, utilizando piezas del kit 4544 para su construcción.
- v. Rampa al final de la bandeja dónde se colocan las cartas, conectado al soporte del sistema robótico.

Para el rodillo de reparto de cartas, la bandeja de reparto y la rampa final se han empleado las mejoras ensayadas en el Modelo 1, anteriormente explicado. Solamente se ha modificado la forma de reparto (bandeja) y el soporte robótico.

Una vez construido, se realizaron una serie de pruebas iniciales, obteniéndose estas conclusiones:

- a. La bandeja dónde se va a repartir las cartas debe tener una pared a cada lado, para evitar que la carta caiga de lado, que debe estar unida a la bandeja dónde se colocan las cartas.

Esta bandeja debe tener una separación lo suficiente amplia para diferenciar las cartas de un jugador de otro. Esta separación, debe ser fija para evitar que las cartas de un jugador se repartan en otro.

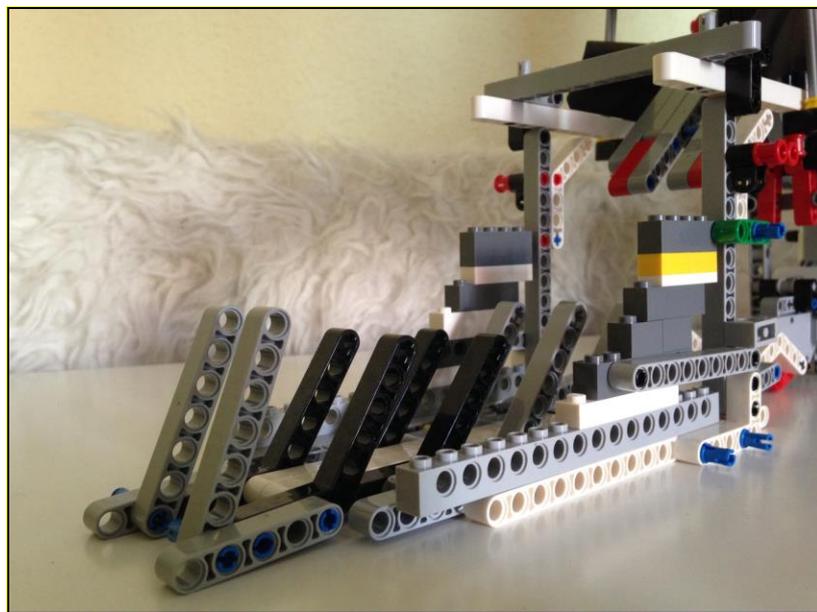


Figura 4-19: Bandeja dónde se va a repartir las cartas correspondientes al modelo 2 con piezas del kit 4544 de LEGO.

- b. El soporte se ha conseguido colocando el Ladrillo inteligente EV3 detrás de la bandeja de reparto de las cartas, para contrarrestar el peso de la bandeja donde se colocan las cartas, y poder elevarlo para que reparta correctamente a la bandeja de reparto.

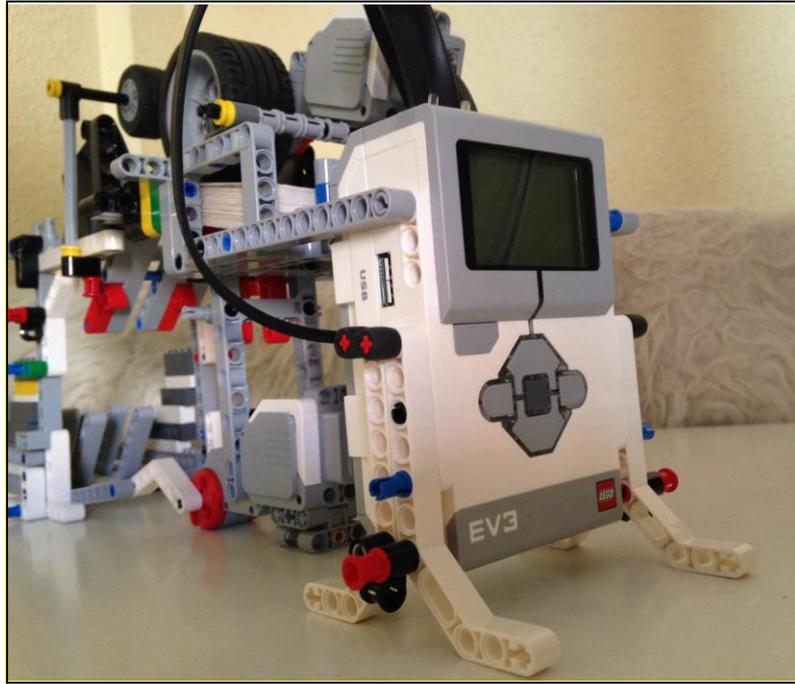


Figura 4-20: Soporte del sistema robótico del modelo 2 con piezas del kit 4544 de LEGO.

Tras realizar una serie de pruebas para su correcto funcionamiento, fue necesario colocar una rueda en la rampa que une el final de la bandeja de reparto de cartas, para evitar que la carta sea repartida en la ranura de otro jugador distinto al que le corresponde, y para que no caiga boca arriba.

La construcción del lanzador de cartas de este modelo es igual que la del modelo inicial, salvo por estas dos mejoras. Tras realizar una serie de pruebas iniciales, se obtuvieron las siguientes conclusiones:

- i. Todos los elementos que conforman el lanzador de cartas modelo funcionan exactamente igual que el modelo inicial.
- ii. Con la mejora introducida del modelo anterior, funciona correctamente sin ningún fallo, según se explicará en capítulo 5, obteniéndose un modelo óptimo para esta forma de repartir.

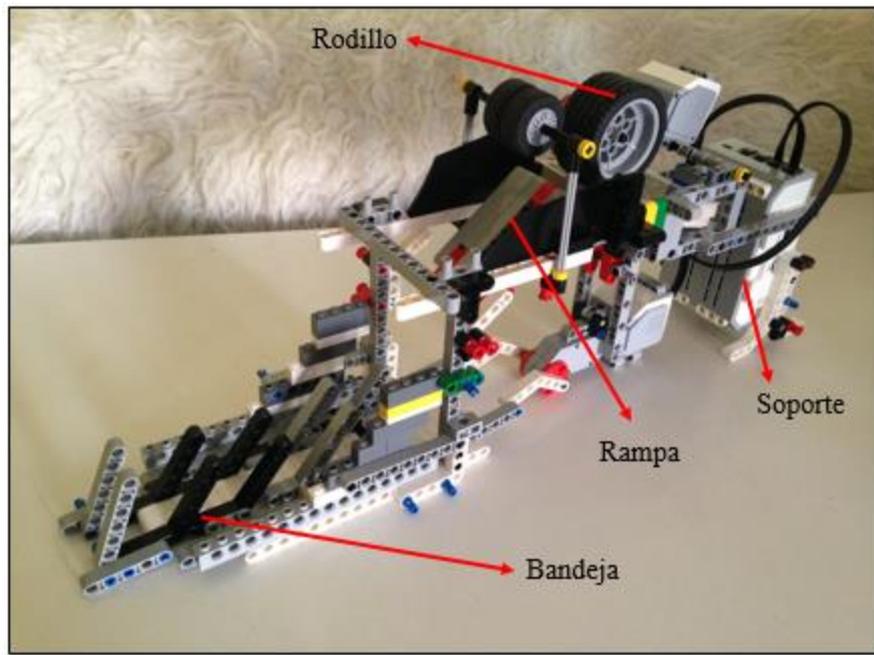


Figura 4-21: Modelo óptimo del modelo 2 con piezas del kit 4544 de LEGO.

5 Pruebas de campo.

5.1 Procedimiento.

En este capítulo se mostrarán las pruebas realizadas para cada algoritmo y cada modelo, justificando en qué situaciones y con qué tipo de cartas es conveniente usar un algoritmo u otro para cada uno de los modelos explicados en el capítulo 4.

También se justificará la evolución de cada modelo respecto, según se fue haciendo necesario por los resultados de las pruebas.

En todas las pruebas para ambos modelos, se tendrán en cuenta los siguientes aspectos:

- i. Se realizarán entre 15 a 20 pruebas con distintos niveles de velocidad del motor y distintos tipos de cartas (nuevas, seminuevas, usadas) para el reparto de cartas, mostrando la media de fallos y desviación típica para cada algoritmo y cada modelo correspondiente. Se quiere estimar la velocidad óptima para cada modelo y observar la eficacia de cada algoritmo (resultados en el Anexo C: Tabla de Resultados).
- ii. Tasa de fallos, se define como el número de cartas erróneas frente al número de cartas totales. Una carta errónea se refiere a: un jugador que tiene más de una carta que los demás o que no se ha repartido la carta a su jugador en la posición que le corresponde. La expresión para calcularlo:

$$\text{tasa de fallos} = \frac{\text{número de cartas erróneas}}{\text{número de cartas totales}}$$

Figura 5-1: Expresión de tasa de fallos.

- iii. Tiempo medio de ejecución; es decir, el tiempo medio que tarda el robot en repartir todas las cartas (52 cartas) para cada conjunto de pruebas para las distintas cartas y distintos algoritmos, de cada modelo correspondiente.
- iv. No se considera que ha habido un error cuando los motores dejan de funcionar por unos segundos, ya que este problema se debe a la API en la que se ha basado para construir la plataforma robótica, y no al robot o al software desarrollado en este proyecto. Sin embargo, para el correcto funcionamiento del mismo sería necesario que dicho API fuera corregida, o desarrollar nuestra propia API, lo que está fuera del alcance de este Proyecto.

5.2 Pruebas del modelo 1.

5.2.1 Algoritmo A.

Para realizar las pruebas se partió del modelo 2, explicado en el capítulo 4.

Primero fue necesario comprobar que repartía las cartas una por una; sin tener que repartir a cada jugador. Para ello, realizaron una serie de pruebas cambiando una serie de parámetros correspondientes al motor que actúa sobre el rodillo de reparto de las cartas:

- a. Grados de reparto, es decir; grados necesarios para conseguir que salga la carta correctamente de la bandeja de reparto.
- b. Velocidad de reparto; es decir; la velocidad necesaria para que consiga sacar una carta correctamente de la bandeja de reparto; salvo para la última carta. Este parámetro se ha ido ajustando en función de cuántos grados de reparto han sido necesarios.
- c. Grados de retroceso, es decir, grados necesarios para evitar que salga más de una carta, cuando ya ha echado una carta de la bandeja de reparto; salvo para la última carta. Este parámetro se ha ido ajustando en función de los anteriores.

En este algoritmo se ha introducido dos grados de retroceso, para asegurarse que nunca reparta más de una carta cuando ya ha echado una carta anteriormente (diagrama de flujo: *Figura 4-2: Algoritmo A.*).

- d. Velocidades de retroceso, es decir; las velocidades necesarias para cada grado de retroceso; explicadas en el anterior punto. Este parámetro se ha ido ajustando en función de los grados de retroceso necesarios.

En este algoritmo a cada velocidad le corresponde su grado de retroceso, cuyos valores se han ido ajustando tras unas pruebas iniciales (diagrama de flujo *Figura 4-2: Algoritmo A.*).

Se han ido probando distintos conjuntos de parámetros, buscando el conjunto óptimo para este modelo (*C Tabla de resultados.*).

Tras realizar una serie de pruebas iniciales, los grados y velocidades de retroceso para este algoritmo:

Grados de retroceso (1° retroceso)	-80
Velocidad de retroceso (1° retroceso)	180
Grados de retroceso (2° retroceso)	-80
Velocidad de retroceso (2° retroceso)	80
Grados de reparto	280

Tabla 5-1: Parámetros del Algoritmo A para el modelo 1 (*).

(*) Las velocidades son negativas debido a la programación; que se explicará en el Anexo B: Manual del programador.

(*) Los grados son absolutos debido a la programación; que se explicará en el Anexo B: Manual del programador.

Con estos mismos parámetros de la tabla anterior, se procedió a realizar una serie de pruebas iniciales. Se encontró una tasa de fallos estaba por encima de 5, es decir; había más de 5 cartas que no se repartían correctamente.

Para evitar que la tasa de fallos estuviese por encima de 5, se procedió a realizar una serie de modificaciones; explicadas en el capítulo 4 de forma detallada:

- a. Mejorar la rampa de reparto de cartas.
- b. Elevar la bandeja de reparto de las cartas.
- c. Colocar un obstáculo entre la rampa y la bandeja de reparto de las cartas, para que reparta de una en una; y no reparta más de una carta a cada jugador.

Con las mejoras realizadas anteriormente explicadas, se procedió a realizar las pruebas variando la velocidad de reparto; cuyos resultados se muestran en la *Tabla 0-2* con las siguientes conclusiones:

- i. Ninguna de las velocidades probadas para realizar es óptima, debido a que presenta tasa de fallos.

De todas las velocidades probadas, la menos conveniente es la máxima (-100), debido a que es la presenta más tasa de fallos, y la más conveniente, es la velocidad mínima (-60), debido a que es la que presenta menos tasa de fallos.

Esta tasa de fallos se debe a que las cartas posicionadas de la 30 a la 52 dejaban de ser correctamente repartidas.

- ii. El tiempo medio de ejecución es aceptable, debido a que no tarda más de 5 minutos en repartir las cartas a todos los jugadores.

- iii. No se consigue con este algoritmo una tasa de fallos nula.

La velocidad máxima (-100) no es conveniente debido a que de las quince pruebas, siete tienen una tasa de fallos entre 2 y 3 sobre el número total de cartas. Y si observa la desviación típica, está muy desviada de la media.

Con la velocidad mínima (-60) tampoco es conveniente, debido a que de las quince pruebas, cinco tienen una tasa de fallos de 1 sobre el número total de cartas. Funciona mejor que la velocidad máxima debido a que su media y su desviación se aproximan a ser nulas.

Para las demás velocidades (-75 y -80), sus medias y desviaciones están en torno a 1 o 2. De manera, que si va disminuyendo las velocidades, sus medias y desviaciones se van reduciendo, de forma que va obteniendo una tasa de fallos casi nulo; pero nunca nulos.

- iv. Se han realizado las pruebas, con todo tipo de cartas (nuevas, seminuevas y viejas), y todas dan los mismos resultados con los parámetros presentados. Por lo tanto, este algoritmo se puede usar para todo tipo de cartas.

Con estas observaciones, se deduce que no es conveniente usar este algoritmo bajo ninguna circunstancia debido a que su tasa de fallos no es nula. Por tanto no tiene sentido realizar pruebas incluyendo el sistema de rotación, ya que ni siquiera el lanzamiento de cartas sin rotación funciona.

Se proponen una serie de mejoras para este algoritmo, para que su funcionamiento sea óptimo:

- i. Hacer que no sean fijos los grados de reparto empleados para el motor del rodillo para repartir la carta, sino que vayan aumentando conforme van repartiendo las cartas, para evitar que haya problemas al repartir las últimas cartas.
- ii. Que la rueda de lanzamiento de cartas retroceda una sola vez tras echar una carta y reparta la siguiente, para reducir el tiempo de ejecución para repartir todas las cartas.

5.2.2 Algoritmo B.

Este algoritmo, es el Algoritmo A antes explicado; pero con las mejoras planteadas en diseño y construcción.

Primero fue necesario comprobar que repartía las cartas una por una; sin cambiar la posición para cada jugador.

Los grados empleados van en aumento, salvo las dos últimas cartas cuyos grados son fijos. Mediante la siguiente tabla, tras realizar una serie de pruebas iniciales:

Número de carta	Grados
1-10	280
11-20	290
21-30	300
31-40	310
41-50	320
51-52	360

Tabla 5-2: Grados según la carta seleccionada.

Los grados de reparto desde la primera carta hasta la antepenúltima, van aumentando de diez en diez, partiendo de un valor inicial que es el empleado en el Algoritmo A: 280 grados.

Con estos parámetros, se observó que ya no hacía falta un segundo retroceso del motor tras echar la carta, obteniendo estos parámetros:

Grados de retroceso (1° retroceso)	-80
Velocidad de retroceso (1° retroceso)	180

Tabla 5-3: Parámetros del Algoritmo B ().**

(*) Las velocidades son negativas debido a la programación; que se explicará en el Anexo B: Manual del programador.

(*) Los grados son absolutos debido a la programación; que se explicará en el Anexo B: Manual del programador.

Con estas mejoras, se procedió a realizar las pruebas; variando solamente la velocidad de reparto; cuyos resultados se muestran en la *Tabla 0-3*, con las siguientes conclusiones:

- i. Todas las pruebas realizadas tienen una tasa de fallos nula, salvo para la velocidad máxima (-100). Esto se debe, a que se han aumentado los grados de reparto para el motor que forma parte del rodillo de reparto; de manera que no hay problemas en repartir las últimas cartas.

Observando la velocidad máxima, su tasa de fallos es menor produciéndose solamente tres fallos de las quince pruebas realizadas. Y los fallos oscilan entre 1 y 2 cartas erróneas de las totales que se reparten.

- ii. La media y la desviación son nulas en todas las velocidades, salvo para la máxima explicado en el punto anterior.
- iii. El tiempo de ejecución es aceptable, tardando aproximadamente casi dos minutos y medio en repartir todas las cartas.

Se observa que este algoritmo reparte las cartas correctamente una por una, consiguiendo una tasa de fallos nula y un tiempo de ejecución aceptable.

A continuación se procede a realizar una serie de pruebas para el sistema de rotación cuyo objetivo es encontrar la velocidad y grados de rotación óptimos; correspondientes al motor que actúa sobre el sistema de rotación. El objetivo de estas pruebas es poder diferenciar bien las cartas de un jugador de los adyacentes.

Después de realizar una serie de pruebas iniciales, se utilizó como parámetro fijo el número de grados del motor correspondiente al sistema de rotación. Partiendo de un valor inicial de 60 grados se va aumentando de forma lineal, por ejemplo: del jugador uno al jugador cuatro es necesario aumentar tres veces el valor inicial, o del jugador uno al jugador dos es necesario aumentar dos veces el valor inicial.

Se procedió a realizar las pruebas variando la velocidad de reparto; cuyos resultados se muestran en la *Tabla 0-4*, con las siguientes conclusiones:

- i. Todas las pruebas realizadas tienen una tasa de fallos nula, tanto para la velocidad máxima como para la mínima.
- ii. El tiempo de ejecución es aceptable, tardando aproximadamente casi tres minutos en repartir las cartas.

Con todas estas mejoras, se ha conseguido un algoritmo óptimo para este modelo que consiga repartir correctamente y con un tiempo de ejecución aceptable.

5.2.3 Conclusiones y posibles mejoras del modelo 1.

- a. El algoritmo B funciona mejor que el Algoritmo A, debido a que no tiene tasa de fallos y su tiempo de ejecución es menor.
- b. Con estos dos algoritmos, siendo uno la evolución de otro; se podrían implementar más mejoras:
 1. Reducir más el tiempo de ejecución: mientras gira el sistema de rotación sea capaz de retroceder la carta antes de repartir la siguiente.

2. Crear una bandeja de reparto, en la que cada carta caiga a un espacio reservado para cada jugador, de manera que se evite que las cartas de distintos jugadores se mezclen.

5.3 Pruebas del modelo 2.

El algoritmo A no tiene sentido seguir probándolo, ya que como se explicó en durante las pruebas del modelo 1 este algoritmo no funcionó.

5.3.1 Algoritmo B.

Este algoritmo, es el Algoritmo A del modelo 2 antes explicado; pero con las mejoras planteadas en diseño y construcción.

El procedimiento para realizar las pruebas de este algoritmo, es el mismo para el algoritmo B del modelo 1, con una diferencia por su forma de repartir las cartas.

En este modelo, ha sido necesario realizar pruebas con velocidades más bajas que en el modelo 1 para el motor que se encarga de echar las cartas. Esto se debe a que la carta debe caer de manera síncrona en el espacio reservado para cada jugador. En las pruebas se obtuvieron los valores de *Tabla 0-5*:

- i. Todas las pruebas realizadas tienen una tasa de fallos nula, salvo a partir de una velocidad de -60.

Con la velocidad máxima (-100), la media y la desviación típica es la más elevada, teniendo fallos en todas las pruebas para esta velocidad.

En este modelo, es conveniente tener una velocidad de echar las cartas no tan elevada, porque puede ocasionar que la carta no caiga en el hueco correspondiente de cada jugador.

Las velocidades óptimas se encuentran en un intervalo entre -40 y -50; siendo la mejor velocidad -45.

- ii. El tiempo de ejecución es aceptable, tardando aproximadamente 2 minutos y medio.

Una vez realizadas las pruebas referentes al motor de echar cartas, se realizaron las pruebas incluyendo el sistema de movimiento de la bandeja. En este modelo, a diferencia del modelo 1, para poder diferenciar los espacios reservados para cada jugador se han tenido que encontrar los grados óptimos para cada uno de ellos (Anexo de resultados), es decir; los grados óptimos para desplazarse de un jugador a otro. Por ejemplo: desplazar la bandeja del jugador 1 al jugador 4 (dado el diseño, el movimiento de la bandeja no es lineal respecto a la rotación del motor).

Con los grados óptimos encontrados se han realizado distintas pruebas para encontrar las velocidades óptimas de la bandeja de reparto para desplazarse de un jugador a otro, obteniéndose los datos de las tablas restantes a partir de la *Tabla 0-6*:

- i. Para cada velocidad de cada tabla correspondiente se han obtenido diferentes velocidades óptimas.

Para aquellas velocidades con una tasa de fallos elevada su media y desviación típica es alta.

- ii. El tiempo de ejecución es aceptable, tardando aproximadamente casi dos minutos y medio en repartir las cartas.

Con todas estas mejoras, se ha conseguido un algoritmo óptimo para este modelo que consiga repartir correctamente y con un tiempo de ejecución aceptable.

6 Líneas de futuro y conclusiones

6.1 Conclusiones

- i. Se ha realizado un estudio exhaustivo para elegir la plataforma robótica adecuada, teniendo en cuenta que cualquier usuario sin conocimientos técnicos pueda realizar el montaje y el uso de la plataforma robótica.
- ii. Se han desarrollado dos modelos distintos de reparto, uno en forma rotacional (como reparte un humano), y otro en una bandeja.
- iii. Para llevar a cabo la construcción óptima de cada modelo, se han tenido que realizar varias evoluciones de un modelo de partida, en función de las pruebas realizadas explicadas en los capítulos 4 y 5.
- iv. Una de las grandes dificultades del desarrollo ha sido encontrar los parámetros adecuados para el motor que se encarga de repartir las cartas, junto con el diseño óptimo para dicho sistema de reparto. La manera de encontrarlo ha sido mediante una serie de pruebas explicadas en el capítulo 5.
- v. Las versiones finales de los modelos y los algoritmos de reparto tiene una fiabilidad adecuada (no se observan fallos en 20 repartos de manos), y una velocidad suficiente (entre 2 y 3 minutos por reparto).
- vi. Cualquier usuario puede construir el sistema robótico construido, con sólo tener los kits propuestos, un PC y el software asociado (que se proporcionará en forma de ejecutable).
- vii. El sistema desarrollado y elegido es considerablemente más barato que cualquiera de los disponibles en el mercado.

Sin embargo hay dos problemas conocidos en la versión actual del sistema: a) la duración de la batería de la plataforma elegida es limitada; y b) la API que empleamos para la conexión del PC con el robot (que es software libre de terceros) tiene algunos bugs que afectan a la fiabilidad del sistema.

6.2 Trabajo futuro

A continuación se muestran aquellos puntos a mejorar o que se deben revisar:

- i. Hay que resolver los problemas detectados con la API de comunicación entre el PC y el robot. Si no sale una versión mejorada que haya resuelto esos bugs, habría que analizar la posibilidad de desarrollar nuestra propia API, o de buscar otra.
- ii. Añadir un módulo de reconocimiento de cartas, de manera que no sea necesario partir de un mazo de cartas ordenado previamente.
- iii. Que la interfaz de usuario se pueda comunicar con el robot mediante Wi-Fi o Bluetooth (actualmente es mediante USB).
- iv. Implementar una app para Smartphone o Tablet; válida para cualquier sistema operativo, de manera que un usuario sin conocimientos técnicos lo pueda manejar desde cualquier dispositivo.
- v. Conectar el sistema de reparto de naipes con un sistema de confección de manos.

Referencias

- [1] Arduino características, ventajas y desventajas. 2.2.1

<https://es.wikipedia.org/wiki/Arduino>
<https://sites.google.com/site/temasdedisenoymanufactura/arduino#TOC-Ventajas-y-desventajas-del-Arduino>
<http://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino>
<http://www.prometec.net/aref/>
<http://www.cortoc.com/p/arduino.html>
<http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>
<http://spainlabs.com/wiki/index.php?title=Arduino>
<http://tienda.bricogeek.com/53-shields-arduino>
<https://en.wikipedia.org/wiki/Arduino>

- [2] Raspberry Pi: características, ventajas y desventajas. 2.2.2

https://en.wikipedia.org/wiki/Raspberry_Pi
<http://www.raspberrypi-spy.co.uk/2014/07/raspberry-pi-model-b-3-5mm-audiovideo-jack/>
<http://www.xataka.com/componentes/nueva-rasberrypi-vs-sus-competidores-buscando-el-mejor-mini-ordenador-barato>
<http://www.makeuseof.com/tag/how-noobs-for-raspberry-pi-can-help-first-time-users/>

- [3] Phidgets: características, ventajas y desventajas.2.2.3

<http://ro-botica.com/pdf/Catalogo-RO-BOTICA-Fischertechnik-2014.pdf>
http://www.phidgets.com/products.php?product_id=1018
http://www.phidgets.com/docs/Digital_Input_Primer
<https://phidgets.wordpress.com/2014/07/07/where-we-found-our-old-frenchman/>
http://www.phidgets.com/products.php?category=0&product_id=1002_0
<http://www.phidgets.com/phorum/>
http://www.phidgets.com/docs/1002_User_Guide
http://www.phidgets.com/docs/Software_Overview#Operating_System_Support
http://www.phidgets.com/docs/What_is_a_Phidget%3F
http://www.phidgets.com/docs/Category:Application_Guides
http://www.phidgets.com/docs/1018_User_Guide#Technical_Details

- [4] Meccano: características, ventajas y desventajas. 2.2.4

<http://www.meccano.com/faqs>
http://www.meccano.com/?locale=es_ES
<http://www.meccano.com/meccanoid-build-anything>
<http://www.hwlibre.com/meccanoid-g15-ks-el-robot-open-source-de-meccano/>
<http://www.meccano.com/meccanoid-opensource>
http://cdn.meccano.com/open-source/Meccano_SmartModuleProtocols_2015.pdf
<https://en.wikipedia.org/wiki/Meccano>

- [5] Lego Mindstorm EV3: características, ventajas y desventajas. 2.2.5

https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3
<http://www.lego.com/es-es/mindstorms/?domainredir=mindstorms.lego.com>
https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3

[6] Fischertechnik: características, ventajas y desventajas.2.2.6

<http://www.fischertechnik.de/en/Home.aspx>
<http://www.ro-botica.com/Producto/Software-ROBO-PRO-Fischertechnik/>
<http://www.fischertechnik.com.mx/acerca-de>
<http://www.vsi.cs.uni-frankfurt.de/research/fischer/apidoc-1-1/index.html>
<https://www.jrl.cs.uni-frankfurt.de/web/projects/fischertechnik-meets-linux-macosx/>
<http://www.ni.com/example/30731/en/>
http://www.fischertechnik.de/PortalData/1/Resources/didactic/documents/activity-booklet/LTBeginnerLab/Beginner_E.pdf
<http://ro-botica.com/pdf/Catalogo-RO-BOTICA-Fischertechnik-2014.pdf>

[7] Demanda y oferta. 3

<http://www.duplimate.com/dupuk/Home.html>
http://web.mit.edu/mitdlbc/www/manager/Duplimate_vs_Dealer4.pdf
<https://www.ebu.co.uk/newsletters/?id=11&page=14>
<http://www.dealer4.com/index.htm>
<http://worldbridge.org/zonelist.aspx?qzone=1>
<https://www.youtube.com/watch?v=w9CMtBYJvQE>
<https://www.youtube.com/watch?v=IpiVRIXzXq8>
<https://www.youtube.com/watch?v=3I2ZO65Nrsc>
<https://www.youtube.com/watch?v=2h1DkAUdmvs>

[8] Videos de guía para el diseño del robot.

<https://www.youtube.com/watch?v=RivnyFhDn-c>
<https://www.youtube.com/watch?v=QdyzqpKL6u0>
<https://www.youtube.com/watch?v=YUO2zbYEvLg>

[9] API del proyecto.

<https://lego3v3.codeplex.com/documentation>

Glosario

API	Application Programming Interface
Algoritmo	Conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.
C#	Lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework.
Diagramas de flujo	Representación gráfica de un algoritmo o proceso.
Gestor de arranque	Programa que se encarga de preparar todo lo que necesita el sistema operativo para funcionar.
IDE	Arduino software.
LCD	Pantalla de cristal líquido.
LabVIEW	Plataforma y entorno de desarrollo para diseñar sistemas, con un lenguaje de programación visual gráfico.
Microcontrolador	Circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.
Nivel de caché	Memoria de acceso rápido de una computadora, que guarda temporalmente los datos recientemente procesados.
Processing	Lenguaje de programación y entorno de desarrollo integrado en código abierto basado en el lenguaje de programación Java.
RAM	Memoria principal de la computadora, donde residen programas y datos, sobre la que se pueden efectuar operaciones de lectura y escritura.
Robot	Sistema electromecánico que, por su apariencia o sus movimientos, ofrece la sensación de tener un propósito propio.
Soc	System on chip, un ordenador con un solo chip.
SPI	Especificación de interfaz de comunicación serie síncrona usada para comunicación a corta distancia.

Wiring	Plataforma electrónica para prototipos de código abierto desarrollado por Hernando Barragán a partir del año 2003. Compuesto por: un lenguaje de programación, entorno de desarrollo integrado (IDE) y un microcontrolador de una sola tarjeta.
Windows CE	Sistema operativo desarrollado por Microsoft para sistemas embebidos.
XBMC	Conocido como Kodi, centro de multimedia de entretenimiento plataforma bajo la licencia GNU/GPL (Licencia pública del sistema operativo tipo Unix).

Anexos

A Manual de instalación

Para la instalación del sistema completo, es necesario un pc con sistema operativo Windows 8.1 o superior.

Para poder instalar el programa que lanza las ejecuciones del sistema robótico es necesario tener instalado el programa Microsoft Visual Studio 2013 o superior.



Figura 0-1. Logo Microsoft Visual Studio 2013.

Una vez instalado, abre el programa: Microsoft Visual Studio 2013; y crea un nuevo proyecto.

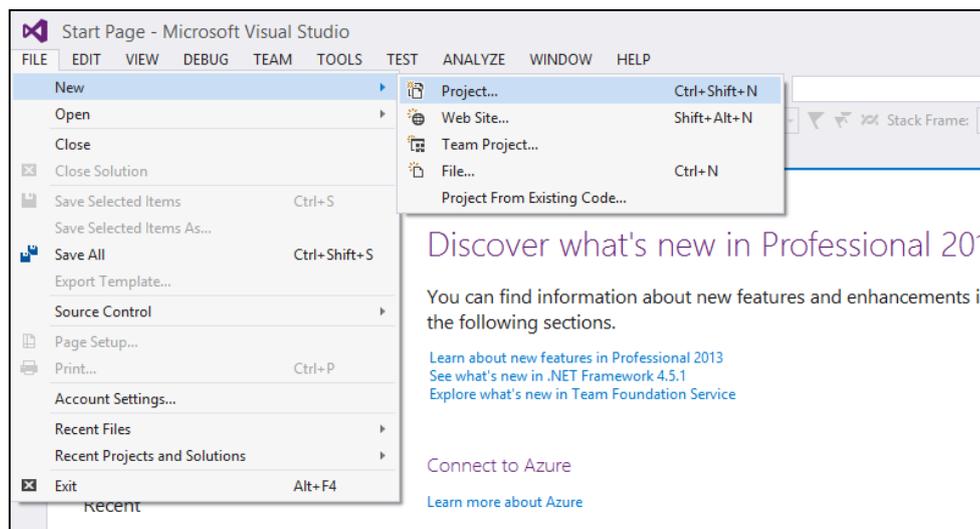


Figura 0-2: Creación de un proyecto.

Una vez abierta, pinche sobre “Proyect” con las opciones del programa mostradas y guarde en la carpeta que desee o la dada por defecto, con el nombre que usted desee.

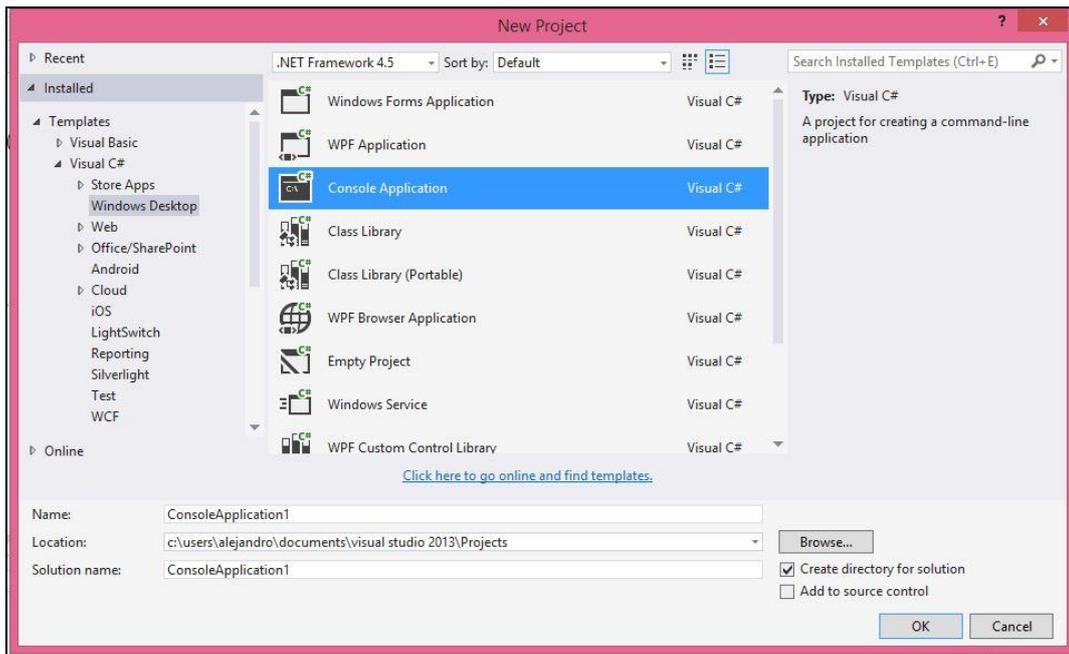


Figura 0-3: Opciones del proyecto.

Una vez creado el proyecto, procede a instalar la API, librería adicional dónde se encuentran las acciones del robot.

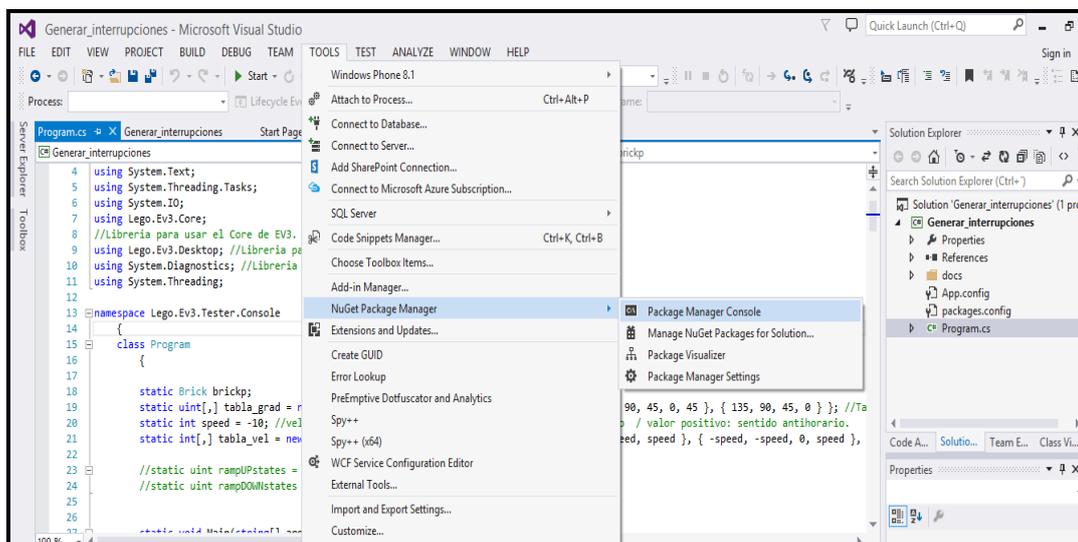


Figura 0-4: Instalación de la API para la plataforma LEGO Mindstorms EV3.

Se le abrirá una pantalla de consola en la pestaña inferior, en la que escribirá: Install-Package Lego.EV3.

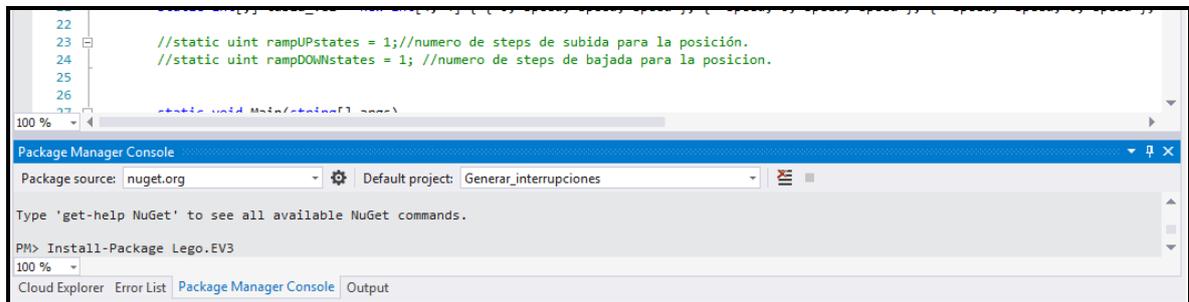


Figura 0-5: Instalación de la API de la plataforma LEGO Mindstorms EV3.

B Manual del programador.

B.1 Formato de fichero de texto para el reparto de manos.

El formato del archivo de texto para repartir las manos es el siguiente:

- i. Las cartas correspondientes a cada jugador, se separan por “:”.
- ii. Los palos de cada carta (Corazones, picas, diamantes y tréboles) separados por “.”
- iii. Cada jugador debe tener 13 cartas, en el que puede tener cartas del mismo o distinto palo.
- iv. Cada carta tiene una correspondencia con un carácter que permite este formato:

Formato del fichero	Carta que corresponde
A	AS
K	K
Q	Q
J	J
0	10
9	9
8	8
7	7
6	6
5	5
4	4
3	3
2	2

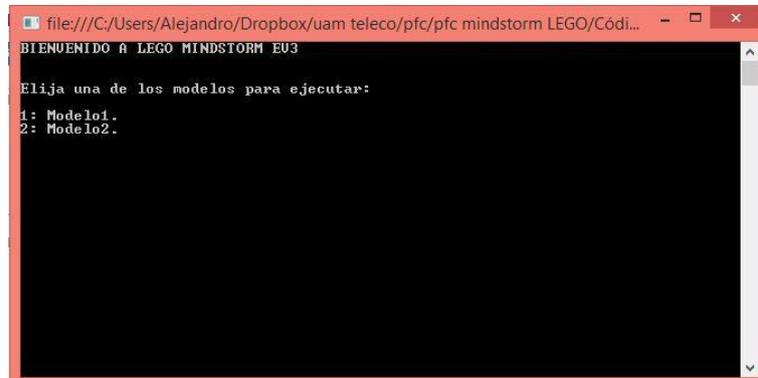
Tabla 0-1: Formato de correspondencia del fichero con las cartas.

- v. Devuelve la carta que corresponde a cada jugador, ordenando las cartas de cada palo tal como se indica en la tabla anterior. Una vez ordenadas las cartas de cada palo, se deben colocar de esta manera, empezando a repartir por el palo de picas, después el palo de corazones, después el palo de diamantes; y por último el palo de tréboles.
- vi. Para la ejecución de este proyecto, se ha utilizado el siguiente formato; pero podría utilizarse otro distinto :

QJ862.Q6.KJ764.K:K07.AK953.083.Q7:A9.0874.Q.J86432:543.J2.A952.A095

B.2 Interfaz de usuario.

Al ejecutar el programa se ha creado una interfaz, para que el usuario pueda elegir el modelo que desea ejecutar. Para luego, elegir el algoritmo que desea ejecutar.



```
file:///C:/Users/Alejandro/Dropbox/uam teleco/pfc/pfc mindstorm LEGO/Códi... - □ ×
BIENVENIDO A LEGO MINDSTORM EU3

Elija una de los modelos para ejecutar:
1: Modelo1.
2: Modelo2.
```

Figura 0-6: Interfaz de usuario.

C Tabla de resultados.

Se muestran los resultados para cada uno de los algoritmos correspondientes a cada modelo óptimo encontrado. En cada tabla se muestran los siguientes resultados:

- i. Tiempo medio de ejecución aproximada; en minutos, que tarda en repartir todas las cartas a todos los jugadores.
- ii. Para cada prueba, se mostrarán el número de cartas que no reparten correctamente para cada parámetro usado; que puede ser grados así como velocidad, de cada prueba. Se han realizado 15 pruebas.
- iii. Para cada parámetro correspondiente al modelo con su algoritmo correspondiente. Se muestra al final de la tabla, la media y la desviación típica encontrada.

C.1 Modelo 1.

Pruebas / Velocidades	-60	-75	-80	-100
1	0	0	0	3
2	0	0	0	0
3	1	5	4	3
4	0	0	0	0
5	1	0	3	3
6	0	3	0	0
7	1	0	2	1
8	0	0	0	3
9	1	3	4	0
10	0	0	0	0
11	0	0	0	2
12	1	4	3	0
13	0	0	0	3
14	0	0	0	0
15	0	0	0	0
Media	0,3333	1	1,0667	1,2
Desviación típica	0,488	1,7728	1,6242	1,4243
Tiempo - minutos	3' 25"	3' 24"	3' 27"	3' 28"

Tabla 0-2: Pruebas realizadas para el algoritmo A del modelo 1 cuando reparte en una única posición (*).

(*) Cada columna muestra el número de cartas fallidas sobre el total, con una velocidad correspondiente para cada prueba. Estos parámetros, se corresponden al motor que forma el rodillo que reparte las cartas.

Pruebas / Velocidades	-60	-63	-65	-70	-75	-80	-100
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	2
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	1
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
Media	0	0	0	0	0	0	0,2667
Desviación típica	0	0	0	0	0	0	0,5936
Tiempo - minutos	2'40"	2' 38"	2' 37"	2' 36"	2' 34"	2' 36"	2' 35"

Tabla 0-3: Pruebas realizadas para el algoritmo B del modelo 1(**).

(**)Cada columna muestra el número de cartas fallidas sobre el total, con una velocidad correspondiente para cada prueba. Estos parámetros, se corresponden al motor que forma el rodillo que reparte las cartas.

Las columnas pintadas en colores, son las velocidades : máxima y mínima; óptimas para este algoritmo. Siendo el color azul claro, la velocidad óptima para este algoritmo, y los demás colores: la velocidad máxima y mínima; para que funcione correctamente este algoritmo.

Pruebas / Velocidades	-60	-63	-65
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	0
10	0	0	0
11	0	0	0
12	0	0	0
13	0	0	0
14	0	0	0
15	0	0	0
Media	0	0	0
Desviación típica	0	0	0
Tiempo - minutos	2' 45"	2' 44"	2'45"

Tabla 0-4: Pruebas realizadas para el algoritmo B del modelo 1 cuando el sistema robótico reparte las cartas a cada jugador (*)**.

(***)Cada columna muestra el número de cartas fallidas sobre el total, con una velocidad correspondiente para cada prueba. Estos parámetros, se corresponden a la velocidad del motor que forma el sistema de rotación del sistema robótico.

La columna azul, se corresponde a la velocidad óptima encontrada para este algoritmo para el modelo 1.

C.2 Modelo 2.

Pruebas / Velocidades	-40	-45	-50	-60	-70	-100
1	0	0	0	0	3	3
2	0	0	0	2	0	4
3	0	0	0	0	0	3
4	0	0	0	0	0	4
5	0	0	0	0	3	5
6	0	0	0	2	0	2
7	0	0	0	0	3	3
8	0	0	0	2	0	4
9	0	0	0	0	0	3
10	0	0	0	2	0	4
11	0	0	0	0	0	3
12	0	0	0	0	3	2
13	0	0	0	2	2	2
14	0	0	0	0	0	2
15	0	0	0	0	0	3
Media	0	0	0	0,66667	0,93333	3,13333
Desviación típica	0	0	0	0,9759	1,38701	0,91548
Tiempo - minutos	2'51"	2'50"	2'50"	2'50"	2'49"	2'48"

Tabla 0-5: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas a cada jugador (*****).

(*****) Cada columna muestra el número de cartas fallidas sobre el total, con una velocidad correspondiente para cada prueba. Estos parámetros, se corresponden al motor que forma el rodillo que reparte las cartas.

Las columnas pintadas en colores, son las velocidades: máxima y mínima; óptimas para este algoritmo. Siendo el color azul claro, la velocidad óptima para este algoritmo, y los demás colores: la velocidad máxima y mínima; para que funcione correctamente este algoritmo.

Jugador actual	Jugador siguiente	Grados
0	0	0
0	1	34
0	2	64
0	3	108
1	0	34
1	1	0
1	2	36
1	3	62
2	0	62
2	1	36
2	2	0
2	3	40
3	0	108
3	1	62
3	2	40
3	3	0

Tabla 0-6: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para cada posición.

Las siguientes tablas, son las pruebas realizadas para distintas velocidades para cada una de las posiciones posibles del jugador actual y el siguiente:

Pruebas / Velocidades	-40	-60	-65
1	0	0	0
2	0	1	0
3	0	0	0
4	0	0	0
5	0	1	0
6	0	0	0
7	0	1	1
8	0	0	1
9	0	0	1
10	0	1	2
11	0	1	1
12	0	0	1
13	0	0	2
14	0	0	1
15	0	0	2
Media	0	0,33333333	0,8
Desviación típica	0	0,48795004	0,77459667
Tiempo - minutos	2'54"	2'51"	2'51"

Tabla 0-7: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para los jugadores 1 a 2.

Pruebas / Velocidades	-40	-50	-60
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0	0	1
8	0	0	1
9	0	0	1
10	0	0	2
11	0	0	1
12	0	0	1
13	0	0	2
14	0	0	1
15	0	0	2
Media	0	0	0,8
Desviación típica	0	0	0,77459667
Tiempo - minutos	2'54"	2'51"	2'51"

Tabla 0-8: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para los jugadores 1 a 3.

Pruebas / Velocidades	-40	-50	-60
1	0	0	0
2	0	0	1
3	0	0	0
4	0	0	1
5	0	0	0
6	0	0	0
7	0	0	0
8	0	0	0
9	0	0	1
10	0	0	2
11	0	0	1
12	0	0	0
13	0	0	2
14	0	0	1
15	0	0	2
Media	0	0	0,73333333
Desviación típica	0	0	0,79880864
Tiempo - minutos	2'54"	2'51"	2'51"

Tabla 0-9: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para los jugadores 1 a 4.

Pruebas / Velocidades	-40	-50	-65
1	0	0	0
2	0	0	2
3	0	0	1
4	0	0	0
5	0	0	1
6	0	0	0
7	0	0	0
8	0	0	1
9	0	0	0
10	0	0	0
11	0	0	1
12	0	0	1
13	0	0	0
14	0	0	1
15	0	0	2
Media	0	0	0,66666667
Desviación típica	0	0	0,72374686
Tiempo - minutos	2'54"	2'51"	2'51"

Tabla 0-10: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para los jugadores 2 a 3.

Pruebas / Velocidades	-40	-50	-65
1	0	0	1
2	0	0	0
3	0	0	1
4	0	0	0
5	0	0	2
6	0	0	0
7	0	0	1
8	0	0	0
9	0	0	1
10	0	0	0
11	0	0	1
12	0	0	1
13	0	0	2
14	0	0	0
15	0	0	2
Media	0	0	0,8
Desviación típica	0	0	0,77459667
Tiempo - minutos	2'54"	2'51"	2'51"

Tabla 0-11: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para los jugadores 2 a 4.

Pruebas / Velocidades	-40	-42	-50
1	0	0	2
2	0	0	2
3	0	0	0
4	0	0	1
5	0	0	0
6	0	0	1
7	0	0	0
8	0	0	0
9	0	0	1
10	0	0	0
11	0	0	0
12	0	0	1
13	0	0	0
14	0	0	1
15	0	0	2
Media	0	0	0,73333333
Desviación típica	0	0	0,79880864
Tiempo - minutos	2'54"	2'51"	2'51"

Tabla 0-12: Pruebas realizadas para el algoritmo B del modelo 2 cuando el sistema robótico reparte las cartas con grados para los jugadores 3 a 4.

(*****)Cada columna muestra el número de cartas fallidas sobre el total, con una velocidad correspondiente para cada prueba. Estos parámetros, se corresponden al motor que forma la bandeja dónde reparte las cartas a cada jugador. Las columnas azul las velocidades óptimas para este modelo con el algoritmo elegido.

Las pruebas realizadas de los jugadores anteriores son las mismas dando velocidades positivas; obteniéndose los mismos resultados.

D Material entregado.

- i. Código para programar la plataforma robótica.
- ii. Videos de construcción de la plataforma robótica.
- iii. Memoria en la que se ha explicado el desarrollo y diseño de la plataforma robótica.
- iv. Modelos construidos de la plataforma robótica junto con los kits necesarios para su construcción.

PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... 2.000 €
- Alquiler de impresora láser durante 6 meses.....50 €
- Material de oficina 150 €
- Compra de las plataformas robóticas.....800 €
- Compra de piezas adicionales para el robot.....30€
- Compra de baterías y pilas..... 50€
- Total de ejecución material 3.080 €

2) Gastos generales

- 16 % sobre Ejecución Material 352 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material 132 €

4) Honorarios Proyecto

- 640 horas a 15 € / hora 9600 €

5) Material fungible

- Gastos de impresión..... 60 €
- Encuadernación 200 €

6) Subtotal del presupuesto

- Subtotal Presupuesto..... 13424 €

7) I.V.A. aplicable

- 21% Subtotal Presupuesto 2819.04€

8) Total presupuesto

- Total Presupuesto 16243.04€

Madrid, julio de 2016

El Ingeniero Jefe de Proyecto

Fdo.: Alejandro De Agustín Urea
Ingeniero de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales, que guiarán la realización de este proyecto de un: **desarrollo de un sistema robótico para el reparto personalizado de manos de contract Bridge**. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es

obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.