



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

— **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE **UPV** INGENIEROS
DE TELECOMUNICACIÓN

Programación de Robots para mejorar la atención terapéutica de niños con Trastornos Generalizados del Desarrollo (TGD).

Autor: Diego Paracuellos de los Santos

Tutora: Valery Naranjo Ornedo

Cotutora: Soledad Gómez García

Cotutora: Rocío Fernández Piqueras

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación.

Curso 2016-17

Valencia, 13 de septiembre de 2017

Resumen:

El Autismo es una condición psicológica que ha sufrido un auge en relevancia en los últimos años, en especial, en relación al número de casos detectados. Su diagnóstico y terapia requieren de un componente humano del que no siempre es posible disponer en el grado necesario, ya sea por condiciones específicas o por la indisposición del paciente a actuar con personas ajenas. Por otro lado, la Robótica ha experimentado un auge en sus posibilidades de uso, tanto desde un punto de vista técnico como económico. En este Trabajo de Fin de Grado (TFG), se pretende analizar las condiciones de los conocidos como Trastornos de Espectro Autista (TEA), y la posibilidad de introducir la Robótica como herramienta de carácter general en la Terapia de Tratamiento para niños con TEA.

Resum:

L'Autisme és una condició psicològica que ha contemplat un auge en rellevància en els últims anys, en especial en relació al nombre de casos detectats. El seu diagnòstic i teràpia requereixen d'un component humà del que no sempre és possible disposar en el grau necessari, ja siga per condicions específiques o per la indisposició del pacient a actuar amb persones alienes. D'altra banda, la Robòtica ha experimentat un auge en les seues possibilitats d'ús, tant des d'un punt de vista tècnic com econòmic. En esta memòria es pretén analitzar les condicions dels coneguts com a Trastorns d'Espectre Autista (TEA), i la possibilitat d'introduir la Robòtica com a ferrament de caràcter general en la Teràpia de Tractament per a xiquets amb TEA.

Abstract:

Autism is a psychological condition that has undergone a surge in relevance in recent years, especially in relation to the number of cases detected. Their diagnosis and therapy require a human component that is not always possible to the extent necessary, either by specific conditions or by the patient's unwillingness to act with other people. On the other hand, Robotics has experienced a boom in its possibilities of use, both from a technical and economic point of view. In this report we intend to analyze the conditions known as Autism Spectrum Disorders (ASD), and the possibility of introducing Robotics as a general tool in the Treatment Therapy for children with ASD.

Agradecimientos

Antes de comenzar el actual documento, me gustaría agradecer a la tutora de este TFG, Valery Naranjo Ornedo, la oportunidad brindada con la realización del mismo, así como su apoyo durante las etapas llevadas con el proyecto en sí.

Además, me gustaría mencionar la labor de Soledad Gómez García y Rocío Fernández Piqueras, profesoras en la Universidad Católica de Valencia, por su colaboración en el co-tutelaje del TFG, que han permitido un desarrollo tanto a nivel profesional como personal, del trabajo realizado.

Para finalizar, me gustaría agradecer a Doña Marga Cañadas y al centro de educación infantil L'Alqueria por su colaboración en los aspectos más humanos de este trabajo, permitiéndonos obtener un mejor enfoque y entendimiento de los Trastornos de Espectro Autista.

Compendio de abreviaturas empleadas.

TEA	Trastorno de Espectro Autista
TFG	Trabajo de Fin de Grado
TIC	Tecnologías de la Información y la Comunicación
MIT	Massachusetts Institute of Technology
DSM	Diagnostic and Statistical Manual of Mental Disorders
SD	Secure Digital
API	Application Programming Interface
QR	Quick Response
PIM	Perfil de Inteligencia Múltiple
URL	Uniform Resource Locator
ASR	Automatic Speech Recognition
TTS	Text-to-Speech
DHCP	Dynamic Host Configuration Protocol
IP	Internet Protocol
SQL	Structured Query Language
OLED	Organic Light Emitting Diode
PC	Personal Computer
QR	Quick Response
RGB	Red-Green-Blue
USB	Universal Serial Bus
ASCII	American Standard Code for Information Interchange

1 Contenido

1	Introducción	1
1.1	Objetivos	2
1.2	Metodología	3
2	Marco Teórico	4
2.1	Robótica Educativa	4
2.1.1	Historia de la Robótica	4
2.1.2	Robótica y Educación	6
2.1.3	Aplicaciones sobre Robótica Educativa	10
2.2	Trastornos Generalizados del Desarrollo	11
2.2.1	Trastorno de Espectro Autista (TEA)	11
2.2.2	Experiencias de Robótica con TEA.	14
2.3	Robótica con Aisoy1.	15
2.3.1	Introducción a Aisoy1.	15
2.3.2	Aisoy1. Especificaciones generales.	16
2.3.3	Airos	18
2.3.4	Scratch	20
3	Trabajo empírico	21
3.1	Robot Aisoy	22
3.1.1	Iniciación y Conexión de Aisoy1	22
3.1.2	Entorno ScratchX	26
3.1.3	Funcionalidades Scratch.	28
3.1.4	Funciones Básicas de Aisoy1.	42
3.2	Propuesta de Actividades para TEA con Aisoy	59
3.2.1	Demo Práctica 1. Demo capacidades Aisoy1. (Demo)	59
3.2.2	Demo Práctica 2. Juego con estímulos. (General)	60
3.2.3	Demo Práctica 3. Acciones Rutinarias con Tarjetas y Horarios (Adaptable específico a TEA)	64
3.2.4	Demo Práctica 4. Expresiones y Estados de Ánimo Básicos. (Multi-nivel Especifico a TEA)	69
4	Conclusiones	73
5	Proyectos Futuros	74
6	Webgrafía	75

1 Introducción

En los últimos años, ha existido un gran impulso en las Tecnologías de la Información y la Comunicación (TIC). Este impulso, ligado a la cada vez más evidente inclusión de las mismas en el día a día, ha exigido un cambio en el paradigma general del sector.

Si bien estos precedentes han dirigido gran parte de los avances al desarrollo de dispositivos para el uso cotidiano y para un público general, como puede ser la Domótica o la Telefonía, hay otra serie de campos, cuyo objetivo ha sido más específico. Uno de estos campos es la robótica.

Este campo en particular ha encontrado una gran aceptación y desarrollo en sectores como la Industria, a modo de maquinaria automatizada, o el Entretenimiento, como por ejemplo con el desarrollo de Animatrónicos.

Por otro lado, desde principios del siglo XX, existe una corriente psicopedagógica dedicada a investigar e intentar comprender una determinada alteración mental que, en la actualidad, afecta a un determinado sector de la población, los denominados Trastornos de Espectro Autista o TEA.

Los mismos se conciben como una serie de Anomalías de Carácter Conductual y/o Intelectual que dificultan la conceptualización clásica de aspectos cotidianos del comportamiento clásico del individuo humano. Los mismos se engloban en lo que se conoce como Trastornos Generalizados del Desarrollo, junto con otras alternaciones como el síndrome de Rett o el síndrome de Heller.

Si bien las anomalías anteriormente nombradas pueden adoptar o abarcar diversos grados de afectación y gravedad, lo cierto es que, por lo general, sus efectos pueden ser minimizados con cierto grado de atención y terapia con el individuo en cuestión, especialmente en caso de diagnóstico temprano, en los primeros años de vida del mismo, durante la infancia o niñez.

No obstante, existe cada vez más casos de TEA documentados, y con un descubrimiento más temprano de la condición en sí, por lo que, en ocasiones, la atención requerida para estos casos no puede darse en el grado requerido. Por ello, gracias a la evolución de la electrónica y la robótica, existe la posibilidad y la propuesta de emplear estas áreas como métodos de terapia específicos para estos casos.

A lo largo de la memoria de este *Trabajo de Fin de Grado*, se definirán los preceptos base para el manejo y aplicación de un dispositivo concreto dentro de este ámbito, el Aisoy1, y su aplicación a un caso muy particular, la atención terapéutica para pacientes, en este caso niños, con Trastornos Generalizados del Desarrollo, focalizándonos en uno de los más relevantes: el autismo.

El trabajo en sí, no obstante, requerirá de una caracterización de los sectores de desarrollo en general y de los focos en particular, con el objetivo de informar sobre cada uno de los aspectos desarrollados.

1.1 Objetivos

A la hora de realizar el trabajo, se ha visualizado un Objetivo principal claro.

Objetivo General: Realizar una propuesta práctica para intervención con niños TEA usando el robot Aisoy1.

Con una propuesta práctica nos referimos a no solo la creación de una actividad para la intervención con el niño TEA, sino de la conceptualización de un entorno en el que un Terapeuta sea capaz de crear y/o adaptar una serie de actividades para su puesta en práctica con el niño.

Para poder dar respuesta al objetivo general planteado y con el foco de obtener un plan de acción a la hora de encarar el trabajo, se definen los siguientes objetivos específicos:

1. Adquirir una base sobre la robótica y su aplicación a un ámbito educativo.
2. Adquirir una base sobre los Trastornos de Espectro Autista (TEA).
3. Mezclar las bases de la robótica con las de TEA.
4. Aprender el manejo del robot Aisoy1.
5. Aprender el manejo del Entorno de programación Scratch.
6. Sintetizar una serie de ejercicios de carácter práctico para la Intervención con Niños TEA.

A la vez, los objetivos de este Trabajo de Fin de Grado podrían separarse en dos grupos, objetivos de carácter teóricos y de carácter prácticos.

Dentro de los teóricos encontraríamos aquellos relativos al aprendizaje base sobre los temas a tratar, lo que denominaríamos un marco teórico de las materias.

En lo relacionado con los prácticos, aquellos relacionados con la aplicación de los conocimientos adquiridos mediante el marco teórico, dentro de un espacio y/o contexto determinado.

1.2 Metodología

Para llevar a cabo los objetivos propuestos, es necesario seguir una serie de procedimientos, por ello, la siguiente memoria está dividida en dos partes básicas.

La primera, un estudio teórico tanto de la robótica como de los Trastornos de Espectro Autista, para contextualizar el trabajo en sí.

Para la realización de esta parte teórica se realizará en primer lugar un breve análisis a la historia de la robótica, comentando algunos de los hechos que marcaron un avance significativo dentro de la tecnología en sí. Seguidamente se analizará el contexto de la robótica como herramienta educativa dentro del marco educativo tomando como referencia el español.

Tras analizar el aspecto tecnológico base del trabajo en sí, se introducirá los conceptos de TEA, analizando la historia de los mismos y sus características.

Para finalizar esta parte se introducirá algunos detalles del hardware empleado, el robot Aisoy1, desde un punto de vista teórico del producto.

Después, se desarrollará un estudio más práctico con el Aisoy1, explicando el manejo del mismo con una de las opciones de programación disponibles, el lenguaje Scratch.

Para finalizar, se aplicarán los conceptos aprendidos sobre TEA y Scratch para desarrollar una serie de actividades que permitan una teórica intervención con niños TEA.

Además, con el objetivo de expandir y adecuar los conocimientos de TEA más allá de un contexto puramente teórico, se ha buscado la ayuda de un profesional externo a la dirección del TFG. En este caso se trata de la Directora del Centro de Educación Infantil y Atención Temprana L'Alqueria de la Universidad Católica de Valencia "San Vicente Mártir", Marga Cañadas, con la que se han concertado una serie de reuniones para pedir asesoramiento tanto desde el punto de vista teórico, permitiendo un acercamiento menos textual a lo que sería las características de los niños TEA, como del práctico, ayudando a un desarrollo de actividades más realista a las necesidades de estos gracias a las sesiones de demostración realizadas en su presencia.

2 Marco Teórico

2.1 Robótica Educativa

2.1.1 Historia de la Robótica

Si bien la robótica es un término relativamente reciente, su concepto tiene sus bases desde hace miles de años, bajo el nombre de autómatas.

Remontándonos en el tiempo, ya en el Antiguo Egipto, existían construcciones que evocaban al concepto en sí, con brazos mecánicos en las estatuas de los templos, propulsados por los sacerdotes en los mismos. También en Grecia, existían construcciones similares a las anteriores, basadas en esa ocasión por sistemas hidráulicos.

Más tarde, entre los siglos XVII y XVIII, aparecerían los primeros autómatas humanoides, como los músicos creados por Jacques de Vaucansons, cuyo objetivo era claro, el entretenimiento.

En esa misma época y principios del siglo XIX, durante la revolución industrial, aparecerían los primeros indicios de maquinaria automática, como la hiladora mecánica de Crompton (1779) o el telar de Jacquard (1801).

En 1805, Henri Miallardert fabricó una muñeca mecánica con la capacidad de dibujar mediante lo que, para aquella época, era un complejo sistema de levas.

No se consiguió un avance significativo en la robótica hasta el desarrollo de la electrónica moderna, siendo uno de los mayores avances en este sentido el desarrollo de un dispositivo controlador, de manos de George Charles Devol en 1946, capaz de registrar señales eléctricas por medios magnéticos y permitir su uso en elementos mecánicos. Desde esa fecha hasta hoy en día, han existido diversos hitos dentro del sector.

A continuación, se citan algunas de dichas fechas e hitos [1].

1954	G.C. Devol desarrolla diseños para Transferencia de artículos programada. Patente emitida en Estados Unidos para el diseño en 1961.
1959	Se introdujo el primer robot comercial por Planet Corporation controlado por interruptores de fin de carrera.
1960	Se introdujo el primer robot ‘Unimate’’, basada en la transferencia de artic. programada de Devol. Utilizan los principios de control numérico para el control de manipulador y era un robot de transmisión hidráulica.
1961	Un robot Unimate se instaló en la Ford Motors Company para atender una máquina de fundición de troquel.
1966	Trallfa, una firma noruega, construyó e instaló un robot de pintura por pulverización.
1968	Un robot móvil llamado ‘Shakey’ se desarrolló en SRI (Standford Research Institute), estaba provisto de una diversidad de sensores, así como una cámara de visión y sensores táctiles y podía desplazarse por el suelo.
1973	Se desarrolló en SRI el primer lenguaje de programación de robots del tipo de computadora para la investigación con la denominación WAVE

- 1981 Se desarrolló en la Universidad de Carnegie- Mellon un robot de impulsión directa. Utilizaba motores eléctricos situados en las articulaciones del manipulador sin las transmisiones mecánicas habituales empleadas en la mayoría de los robots.
- 1984 Robots 8. La operación típica de estos sistemas permitía que se desarrollaran programas de robots utilizando gráficos interactivos en una computadora personal y luego se cargaban en el robot.

Tal y como se apuntó al principio de este apartado, el concepto es antiguo, el termino Robótica es más reciente, el mismo se atribuye al escritor Isaac Asimov, y a sus “Tres Leyes de la Robótica”. Las mismas establecen los siguientes conceptos:

- Un robot no dañará a un ser humano o, por su no actuación, permitirá que un ser humano sufra daño.
- Un robot deberá realizar las órdenes dadas por los seres humanos, excepto en el caso que estas órdenes entrasen en conflicto con la 1ª Ley.
- Un robot debe proteger su propia existencia siempre y cuando esta protección no entre en conflicto con la 1ª o la 2ª Ley.

Por otro lado, el concepto robot es atribuido a Karel Kapec y su obra “Rossum’s Universal Robots” en 1917, bajo el término “Robota” que en checo vendría a significar trabajador forzado.

2.1.2 Robótica y Educación

Si bien la robótica como tal ha sido empleada en sectores de carácter generalmente industrial, con el auge de la tecnología sufrido en las últimas décadas, la misma ha sido un área que se ha planteado en diversas ocasiones como potencial contenido y herramienta para otras áreas más particulares, como la enseñanza.

Se podría decir que uno de los precursores de la unión de la robótica y la educación es Seymour Papert, investigador del Instituto Tecnológico de Massachusetts (MIT) y desarrollador de LOGO, un lenguaje de programación educativo pensado para niños.

Por otro lado, algunas corrientes lo atribuyen a Saldaño, Vega y Rojas, donde es creada en 1989 (Saldaño), esquematizada en 1991 (Vega), modelada en 1993 (Rojas y Saldaño) y desarrollada y aplicada en los años siguientes.

La unión de ambos mundos, conocida como robótica educativa, comúnmente se asocia a una relación cruzada entre el proceso educativo y tecnológico, en otras palabras, “la integración de lo que es la tecnología y el curriculum educativo”.

Uno de los objetivos que se pretenden conseguir con esta asociación es conseguir un cambio en la percepción y contextualización de los elementos y/o conceptos que se engloban bajo la realidad de las bases educativas y tecnológicas, provocando no un cambio solo en el individuo sino a nivel cultural.

Por lo general, las distintas propuestas educativas concebidas en base a los modelos de Saldaño y/o Seymour se basan en aprendizajes o programas educativos multifase, en el que las salidas/resultados de una fase se emplean como entradas para la siguiente, respondiendo a un esquema habitualmente de cinco fases [2]:

- Primera: Planteamiento de Solución/Necesidad, en la que el profesor plantea una situación.
- Segunda: Planificación, en la que existe una interacción entre el profesor y el/los alumnos para buscar el método óptimo para implementar la solución.
- Tercera: Elaboración de piezas, en el que el alumno realiza la selección de las piezas a emplear.
- Cuarta: Armado del robot, en la que el alumno monta el robot.
- Quinta: Prueba de Ensayos, en la que se evalúa el grado de obtención de la solución.

Dentro del marco educativo español, han existido en los diferentes planes de estudios bloques que de manera específica o indirecta han podido ser relacionados con la robótica. Analizando algunas de las Leyes educativas podemos encontrar lo siguiente:

Ley	Contenido	Materia	1º, 2º y 3º ESO	4º ESO
LOGSE (1991)	Control Automático y Robótica	Tecnología	<p>Bloque 4. Tecnologías de la información. Lenguajes de programación y desarrollo de aplicaciones.</p> <p>Bloque 7. Control y robótica. Máquinas automáticas y robots: automatismos. Arquitectura de un robot. Elementos mecánicos y eléctricos para que un robot se mueva.</p>	<p>Bloque 3. Tecnologías de la información. El ordenador como dispositivo de control: señales analógicas y digitales. Adquisición de datos. Programas de control. Bloque 6. Control y robótica. Percepción del entorno: sensores empleados habitualmente. Lenguajes de control de robots: programación. Realimentación del sistema.</p>
LOE (2006)	Control y Robótica	Tecnología		<p>Bloque 4. Control y robótica: Experimentación con sistemas automáticos, sensores, actuadores y aplicación de la realimentación en dispositivos de control. Diseño y construcción de robots. Uso del ordenador como elemento de programación y control. Trabajo con simuladores informáticos para verificar y comprobar el funcionamiento de los sistemas diseñados.</p>

Tabla 1.1. Tecnología y Robótica en el marco educativo español.

Ley	Contenido	Materia	1º, 2º y 3º ESO	4º ESO
LOMCE (2013)	Tecnología y Tecnologías de la información y de la comunicación	Tecnología	Bloque 1. Proceso de resolución de problemas tecnológicos. Bloque 2. Expresión y comunicación técnica. Bloque 3. Materiales de uso técnico. Bloque 4. Estructuras y mecanismos: máquinas y sistemas. Bloque 5. Tecnologías de la Información y la Comunicación.	Bloque 1. Tecnologías de la información y de la comunicación. Bloque 3. Electrónica. Bloque 4. Control y robótica. Bloque 6. Tecnología y sociedad.
		Tecnologías de la Información y de la Comunicación		Bloque 1. Ética y estética en la interacción en red. Bloque 2. Ordenadores, sistemas operativos y redes. Bloque 3. Organización, diseño y producción de información digital. Bloque 4. Seguridad informática. Bloque 5. Publicación y difusión de contenidos. Bloque 6. Internet, redes sociales, hiperconexión.

Tabla 1.2. Tecnología y Robótica en el marco educativo español.

En la actualidad, el concepto de Robótica Educativa está tomando un mayor sentido y acogimiento dentro de los centros de enseñanza, siendo cada vez más común la inclusión de estos dispositivos y enseñanzas dentro de las aulas. Dentro de la metodología actual podemos encontrar dos grandes partes diferenciadas:

- La construcción del robot, como un medio lúdico de enseñanza, con el objetivo de enseñar solo los preceptos básicos de la electrónica.
- La dotación de “vida” al robot, mediante lenguajes básicos de programación y la proposición de rutinas de manejo.

Por un lado, esto presenta una serie de beneficios dentro del contexto de enseñanza en las aulas [3]:

- Incentiva la Creatividad, una de las capacidades más importantes de la Inteligencia, mediante el planteamiento de pequeños problemas de los que deban buscar una forma de desarrollarlos.
- Incentiva el Pensamiento Crítico y Lógico, planteando al alumno unos supuestos y relaciones entre las partes del robot.
- Afianza las bases físicas y matemáticas de cara a un posible futuro como ingenieros.
- Crea una base para comportamiento organizativo.
- Permite un modelo educativo más actual, que divierte a la vez que enseña.

No obstante, pueden existir una serie de aspectos determinantes, que ralentizan y/o perjudican las posibilidades de la robótica educativa:

- Cambios en el modo de desarrollo de las clases, pasando de un modelo tradicionalista a modelos con carácter eminentemente más prácticos, como el “flip teaching”.
- Requerimiento de Profesorado con conocimientos en la materia y voluntad para su implementación.
- Agudización de la brecha Analógico – Digital.
- Coste de Implementación actualmente Elevado.
- Falta de Visión Político-Educativa en la materia.

Aun con estas limitaciones descritas, es un hecho que existe una mayor aceptación e inclusión de la robótica como tecnología del día a día, tanto desde un punto de vista teórico, como un conocimiento más común de la materia; como práctico, con una mayor implementación en las funciones que pueden desarrollar.

2.1.3 Aplicaciones sobre Robótica Educativa

Dentro de las aplicaciones dadas a la Robótica Educativa podemos encontrar multitud de propuestas educativas, algunas de carácter más relacionado con la programación, otras con el montaje y un grupo intermedio que denominaríamos aplicación mixta.

Las primeras, se basan en, a partir de un hardware ya otorgado, un robot construido, y un lenguaje ya definido, generar una rutina específica, como puede ser un movimiento condicionado.

En el otro extremo, el segundo caso, encontramos que, a partir de una serie de elementos de construcción, no todos ellos necesariamente requeridos para la elaboración, diseñar un robot que sea capaz de realizar algunas acciones pedidas, como la capacidad de recoger un objeto, reconocer una cara, seguir unas líneas, etc.

Finalmente, encontramos el caso mixto, en el que se juntan conceptos de ambos extremos, planteando un problema de carácter general, se facilitan los materiales y medios, y se espera una solución al problema planteado.

Se puede encontrar una cantidad aceptable de oferta en relación a las actividades a realizar sin importar su naturaleza. Un ejemplo de ello se encuentra en los productos LEGO WeDo, consistentes en una serie de kits de construcción y una suite de programación propia para trabajar con sus productos. Para estos mismos productos podemos encontrar la “Guía del profesor oficial de LEGO WeDo” [4] en la que encontrar multitud de sugerencias para poner en práctica en el aula, tanto de montaje como de programación.

2.2 Trastornos Generalizados del Desarrollo

2.2.1 Trastorno de Espectro Autista (TEA)

Se conoce como TEA una serie de alteraciones neurológicas y de desarrollo de carácter complejo que comienza en la niñez y dura toda la vida. Afecta a los aspectos base de la personalidad, como al comportamiento, la interacción entre personas, la comunicación y el aprendizaje.

El término autismo, del griego autos (por cuenta propia), e -ismo (acción), fue empleado por primera vez por el psiquiatra suizo Eugen Bleuler en 1912, en referencia a un comportamiento atípico de pacientes esquizofrénicos. Este comportamiento, relacionado con el alejamiento de la realidad del paciente, intentó relacionarlo con la esquizofrenia, sin éxito aparente, al ser propio solamente de un subgrupo de los pacientes.

Años después de la muerte de Bleuler, en 1938, el que se podría considerar como padre del autismo, Leo Kanner, realizó un estudio en base a un grupo de niños reducido, todos menores de once años, con cuadros de esquizofrenia infantil y/o problemas emocionales. Anecdóticamente, estos síntomas parecían afligir de manera selectiva a hijos con padres con altas capacidades intelectuales.

En ese estudio, describió que los niños parecían más felices cuando estaban solos, en un estado pseudo-ansioso y obsesivo por la monotonía, y con una auto comunicación simbólica y repetitiva. Si bien los niños habían sido diagnosticados como “débiles mentalmente”, Kanner resaltó que no presentaban síntomas de deterioro cognitivo, siendo en la mayoría de niños todo lo contrario, poseyendo habilidades de memorización y uso del vocabulario por encima de la media. Esto le hizo concluir que si bien los niños sufrían de lo que calificó “autismo infantil temprano” y su comportamiento era similar a la esquizofrenia, las sintomatologías eran intrínsecamente diferentes [5].

De manera paralela, el pediatra vienés, Hans Asperger observó patrones de comportamiento similares en el Hospital Universitario Infantil de Viena. En su Tesis Doctoral, publicada en 1944, identificó una serie de deficiencias del comportamiento y facultades similares a las descritas por Kanner. Estas deficiencias se basaban en dificultades en la integración social, torpeza motora, mirada extraña, pobreza emocional, un lenguaje rebuscado, movimientos estereotipados y escasez de expresión facial.

La diferencia fundamental entre ambos autores radica en la condición de aparición de la sintomática. Mientras que Kanner mantenía que los síntomas eran evidentes desde los seis meses de edad, Asperger por su parte sostenía que su síndrome rara vez era diagnosticado antes de los tres años. Para el Autismo de Kanner, su aparición fue asociada a la aparición retrasada del habla o al mutismo, mientras para Asperger, si podían aprender el uso de gramática y vocabulario, aunque en un contenido pobre e inapropiado en relación al contexto social.

En tiempos recientes, el psicólogo y miembro de la British Academy, Simon Baron-Cohen ha relacionado las dificultades cognitivas del autismo con un retraso en el desarrollo de la Teoría de la Mente, la cual vendría a afirmar que la capacidad cognitiva de un organismo inteligente se basa en una manera de organizar los pensamientos del mismo, así como de la capacidad de relacionar esos mismos pensamientos. Por otra parte, ha realizado diversos estudios neurobiológicos y psicológicos con relación al autismo, así como de la hipermasculinización del autismo y su relación con la testosterona fetal.

En un marco más general, tanto el autismo como el síndrome de asperger, han sido englobados en la última edición (DSM-5) del Manual Diagnóstico y Estadístico de los trastornos mentales, considerada la biblia de la Psiquiatría, bajo el término “Trastornos de Espectro Autista”.

Si bien no está demostrado que los TEA sean una neuropatía genética estricta, al no saber exactamente la genética exacta que afecta a los mismos, sí que se cree que tiene un importante componente genético, aunque por otro lado se estima que este componente genético es solo para la predisposición a la sintomática, no siendo el desencadenante de la condición.

Se les denomina Trastorno de Espectro, debido a que tanto el grado de afectación como la sintomática puede variar entre sujetos, desde problemas en la dialéctica, hasta alteraciones en el comportamiento tanto de carácter intelectual en un sentido clásico como motriz.

Se estipula que los primeros indicios son apreciados aproximadamente a los tres años de edad.

Estas “deficiencias” se pueden clasificar en tres grandes grupos [6]:

- Interacción Social. Son aquellas afecciones donde afectan al sujeto en la relación con otras personas. Son de caracterización más intelectual, llegando a afectar a la integridad del afectado. En muchos casos, las relaciones con personas de fuera del núcleo familiar pueden verse comprometidas, pudiendo afectar de manera negativa el desarrollo del trastorno y la aceptación social del mismo. Algunos de los comportamientos propios de este grupo son la focalización de intereses, los déficits en la expresión emocional, un cierto grado de carencia de empatía o problemas con la comunicación no verbal.

Como ejemplo, típicamente las personas con TEA, sienten cierto grado de reticencia hacia la interacción con personas que no son de su entorno más cercano, como en el caso de sujetos con Asperger debido a la falta de Empatía.

- Comunicación. En este grupo encontraríamos aquellas relacionadas con las capacidades de interacción del afectado ya sea hacia otros seres o con el entorno. Por lo general encontramos dos tipos:

- Intelectuales: En forma de dificultades en la Comprensión y/o expresión Oral, ya sea por trastornos derivados como mutismo y/o por problemas en la entonación. Si bien estas no son una degeneración cognitiva que afecte a la totalidad de los casos de TEA, sí que es una de las que presentan una mayor afectación tanto indicativa como funcional.
- Motriz: En forma de dificultad para realizar acciones como la Gesticulación o la Mímica. Dentro de este tipo, podríamos diferenciar entre las forzosas, como pueden ser tics obsesivos, y las adquiridas o propias, como en el caso de los sujetos de Asperger, que pueden crear sus propios signos de comunicación no-verbal.

El grado de afectación en estas condiciones, puede derivar en la aparición de otras pertenecientes al de Interacción Social. Un ejemplo típico dentro de este grupo es la dificultad que presentan en precisamente en el ámbito de la mímica, lo que en muchos casos deriva en dificultades en el resto de grupos.

- Imaginación. Por último, en este grupo, se conglomeran una serie de comportamientos y/o limitaciones derivados de la limitación en el pensamiento simbólico, es decir, de la dificultad en relacionar acciones con formas, imágenes o situaciones. Algunas de estas limitaciones pueden aparecer con comportamientos obsesivos o rutinarios, dificultades con la expresión de sentimientos o sensaciones, derivado de una carencia en la capacidad mímica o afectaciones en el lenguaje corporal y/o del movimiento.

Un ejemplo curioso dentro de este grupo, es el gusto u obsesión de las personas con autismo por los objetos rotatorios como, por ejemplo, “los spinners”, con tanto auge “actual”.

En el siguiente grafico se recogen estas mismas ideas:

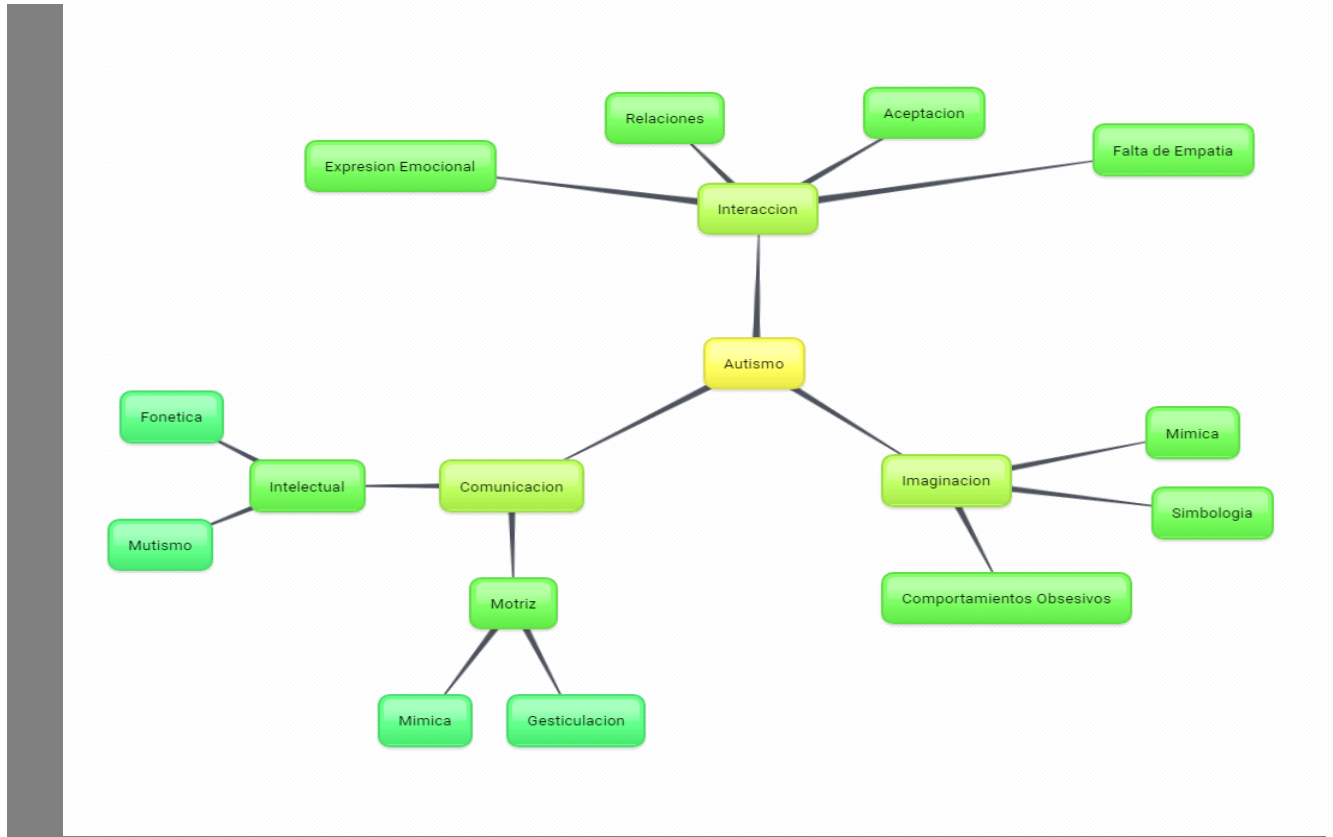


Figura 1. Pilares Conductuales de TEA

En los últimos años se han visto aumentados el número de casos de personas con algún tipo de Trastorno de Espectro Autista. En ocasiones se pretende culpabilizar de este aumento al modo de vida general o a condicionantes relacionados con la sanidad. Si bien es evidente este aumento, la causa más probable del mismo no sea un aumento real de casos, sino una mayor precisión en la identificación y gestión de los mismos, mediante técnicas y terapias más eficientes y enfocadas a una detección temprana de los mismos.

Aunque la detección de los casos ha experimentado un salto razonable, en la actualidad no existen sino métodos paliativos para los afectados, basándose en la minimización de los efectos a largo plazo, siendo que por ahora no existe una solución generalizada.

2.2.2 Experiencias de Robótica con TEA.

Si bien las personas con TEA suelen tener ciertas complicaciones para relacionarse con otras personas, por lo general tienen una buena relación con temas electrónicos y/o digitales. Existen estudios que demuestran que su grado de aceptación de lo relacionado con las Tecnologías de la Información y la Comunicación (TIC) en general es elevado [6][7][8].

Por ello, una de las ideas fundamentales dentro de la terapia para estas personas, pasa por la inclusión de las mismas en su rutina personal. Uno de los estudios que se viene realizando desde hace tiempo es la inclusión de robots dentro de esta terapia. En estos estudios se intenta caracterizar y relacionar los distintos grados de aceptación de los sujetos con TEA con las características de los robots, desde su tamaño hasta su forma [9] [10] [11].

Si bien los resultados de los mismos son en ocasiones dispares, en parte debido a que los grupos de prueba, aun siendo todo TEA, son diferentes, por lo general, existe una predilección por los robots pequeños, de forma pseudo-humanoide y con un alto grado de estimulación sensorial [12].

2.3 Robótica con Aisoy1.

2.3.1 Introducción a Aisoy1.

Aisoy1 es un robot educacional de carácter general. Su fabricante, Aisoy, lo define como “*un robot social para una educación moderna de los niños*”. De manera más técnica, podríamos definirlo como *una herramienta-interfaz de aspecto humanoide con un cierto nivel de inteligencia artificial*.

Una de las intenciones principales de este proyecto, es determinar la viabilidad de la inclusión de éste u otros robots similares en terapias relacionadas con el trastorno de espectro autista (TEA), analizando por separado las capacidades del robot en relación a las posibilidades que requiere el niño TEA, y por otro la complejidad para poner en práctica esas mismas capacidades, tomando como punto base la forma más simple que se tenga de operar con el robot.

En los siguientes sub-apartados, se describirá el robot y su contorno desde un punto de vista meramente teórico, dejando el aspecto practica para los apartados correspondientes a ese fin.

2.3.2 Aisoy1. Especificaciones generales.

Aisoy1, como ya se ha indicado, es un robot de rasgos humanoides cuya finalidad es la de servir de herramienta para el desarrollo educativo en un ámbito general.

Internamente, el modelo empleado (V5), está compuesto por una serie de actuadores y sensores, conectados a una Placa Raspberry Pi 3 Modelo B, y a un teórico Microcontrolador que denominan “Jetpack”.

En la actualidad, existen seis versiones de Aisoy, siguiendo una evolución clara en una versión con respecto a las anteriores.

Algunos de los aspectos más importantes son:

Los Modelos V1 y V2 no empleaban una Raspberry Pi. Su pantalla frontal era diferente.

En el Modelo V3 incorporó la Raspberry Pi, aunque de manera Externa.

En el Modelo V4 implementó la Raspberry Pi en el interior del robot.

En el Modelo V5 posee capacidades de movimiento (opcional) y batería interna.

El Modelo más actual, KiK, incluye mejoras tanto a nivel hardware (inclusión de un pulsador en la parte trasera de la cabeza), como Software (Airos 6).

El Aisoy1 posee un tamaño de alrededor de 22x16x15 cm con un peso aproximado de 1 kg. Actualmente, de los modelos disponibles, se encuentran configuraciones con autonomía de dos horas y cuatro horas [13].



Figura 2. Aisoy1 versión 5.

En relación a los sensores montados en el robot, se puede encontrar una Cámara de 3 Megapíxeles situada en el ojo izquierdo. También dispone de cinco sensores de presión, uno localizado en la parte trasera de la cabeza, estando los otros cuatro situados en los laterales, dos a cada lado. Además, Aisoy1 está provisto de micrófono y un Acelerómetro realizando la función de sensor de posición.

En cuanto a los Actuadores, encontramos un Altavoz Estándar con un rango de 20Hz a 20KHz y una pequeña pantalla OLED (Organic Light-Emitting Diode) de 128x64. También posee un conjunto de cuatro servos, para controlar el movimiento de la cabeza, tanto en horizontal como en vertical, las cejas y los ojos. Además, dispone de una serie de leds de estado y una Luz pectoral, con la capacidad de poder iluminarse de cualquier color. En los modelos con funciones de movimiento, se encuentran un par de Servomotores en la parte inferior del robot.

Además, el Aisoy1 dispone de conectividad, tanto por Ethernet como por 802.11n (wi-fi) y funcionalidades de Bluetooth 4.1 implementados en la Raspberry Pi.

En relación a las capacidades del robot, el mismo es capaz de realizar Reconocimiento de Voz, aunque existen ciertos fallos reconocidos en determinados lenguajes y en especial para voces graves.

También es capaz de realizar acciones de Texto a Voz con 8 modos de configuración distintos, funciones de Movimiento para los modelos de la V5 en adelante, acciones de Gesticulación con ayuda de los servos y un supuesto motor emocional, lectura de códigos QR y reconocimiento de Color mediante la cámara y Reconocimiento de Posición gracias al Acelerómetro.

Respecto a los lenguajes con los que trabajar, podemos encontrar por un lado Scratch y BYOB (Snap!), que son lenguajes de carácter gráfico.

Por el otro, es posible realizar la programación del robot mediante lenguajes de carácter textual como C++ y Python mediante APIs específicas.

2.3.3 Airos

Airos es el Sistema Operativo que da “vida” al robot Aisoy1. Es el encargado de dotar de determinadas capacidades a Aisoy1. Estas capacidades van desde la interpretación de instrucciones, ya sea por *Scratch* o *Python* por lo general, pasando por los motores lingüísticos (Text-To-Speech y Reconocimiento de Voz) terminando en un hasta cierto grado complejo motor emocional.

Por otro lado, desde la V5 de Airos, existe un software de gestión específico para realizar la conexión de los robots denominado Airos Manager, empleado generalmente para la conexión inalámbrica de los robots.

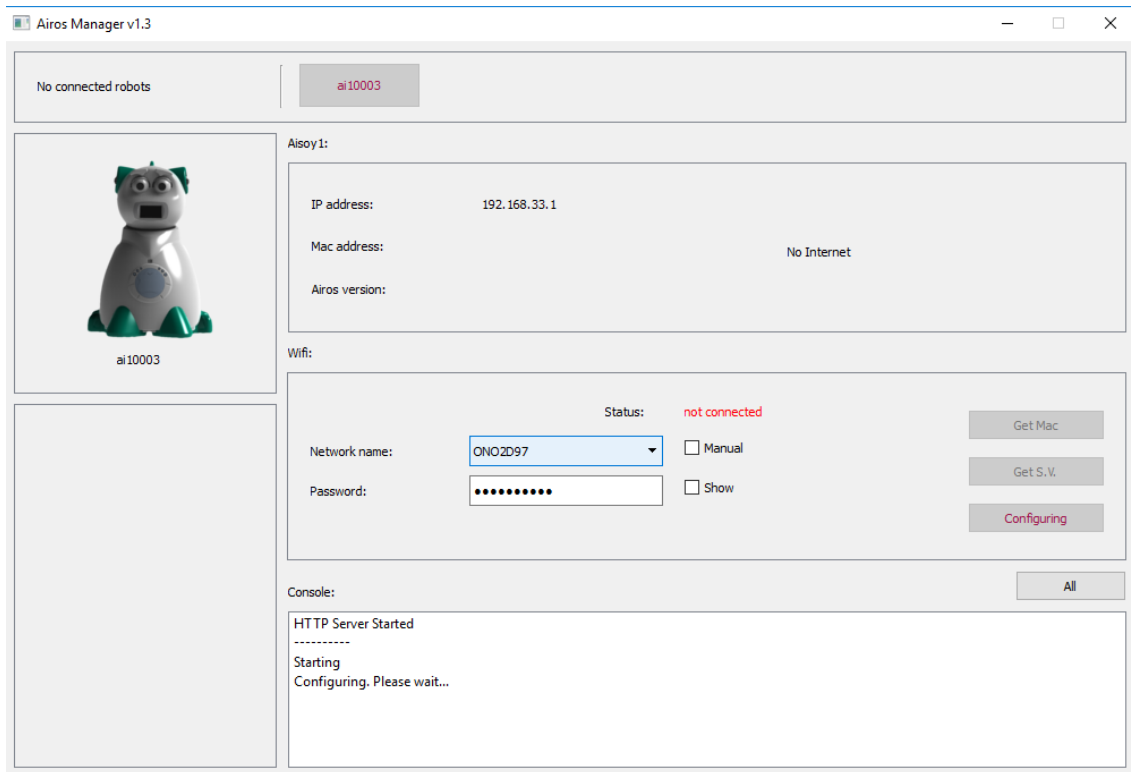


Figura 3. Airos Manager con Robot no vinculado.

En lo que respecta a *Airos Manager*, su interfaz es de carácter simple, permitiéndonos acciones como introducir la configuración *wi-fi* del robot y obtener sus datos de conexión, cambiar el idioma y volumen, Consultar su N° de Serie y la Versión de Airos, Registrarlo o Actualizar su versión de Airos. También es posible lanzar *Blockly*, una de las posibilidades para construir los programas que se ejecutarán con el Aisoy1 y *Scratch Online*, que será con el que trabajaremos.

Recientemente, ha sido lanzada la versión 6 de Airos. Aunque el desarrollo realizado durante este *TFG* y el contenido de la memoria ha sido desarrollado íntegramente *con y para la V5*, se ha considerado oportuno remarcar las mejoras encontradas en esta nueva versión.

Junto con las posibilidades que ya de por sí ofrecía el Aisoy1, la V6 ofrece una visión paralela al trabajo realizado en este *TFG*. Uno de los añadidos más relevantes encontrados junto a la versión *KiK*, es la incorporación de serie de una pseudo-IA que permite al robot aprender y desarrollarse como si de un ser humano se tratase. Este aprendizaje y desarrollo se generan mediante lo que se denominan *Perfiles de Inteligencia Múltiple (PIM)*, en un lapso de entre tres y siete horas desde la iniciación, y separado en tres etapas: Nacimiento, Juventud y Madurez.

Un PIM es un modelo de las capacidades desarrollado por Howard Gardner, en el cual la inteligencia no se considera una capacidad unitaria, sino que es posible separarla en diversas áreas con un determinado nivel de generalidad entre sus funciones. En este caso las áreas son: Musical, Corporal-Cenestésica, Interpersonal, Lingüístico-Verbal, Lógico-Matemática, Naturalista, Intrapersonal y Visual-Espacial.

En lo relacionado con las etapas, la primera, el *Nacimiento*, consiste en la primera hora hasta las dos primeras horas del robot. En la misma sus funcionalidades están limitadas, siendo capaz de entender solamente un pequeño conjunto de palabras y repetirlas.

Durante su *Juventud*, sus capacidades se expanden, permitiéndole entre otras entender frases simples y empezar a expresarse por sí mismo de manera limitada. Esta etapa durará entre dos y cinco horas, en función del nivel de implicación con el robot.

Por último, durante su *Madurez*, con el PIM ya definido, sus capacidades se desbloquean completamente, siendo capaz de entender y emitir frases complejas.

En la actualidad, el sistema sólo es capaz de trabajar con mono PIM, por lo que toda interacción realizada será utilizada para construir un único PIM.

Otro de los cambios más relevantes, es la forma de interacción con los sensores, con la posibilidad de diferentes comportamientos en función del modo en que se interactúe con ellos, como puede ser pulsaciones repetidas en un sensor para denotar una molestia, el mantenerlo para una caricia, etc.

La mayoría de los cambios, por otro lado, están relacionados hasta cierto punto con una aplicación denominada *Aisoy Lab*, que permite la visualización de los parámetros y un cierto nivel de control de manera inmediata, como puede ser la visualización del PIM o la ejecución de determinados juegos incluidos de serie en la V6 [14].

2.3.4 Scratch

Tal y como se ha denotado en anteriores apartados, existe la posibilidad de “programar” el Aisoy1 con diversas opciones diferentes. Como introducción a la operatividad del mismo, se ha elegido Scratch como base de la programación en esta ocasión.

Scratch es un lenguaje de programación de carácter gráfico por bloques gratuito desarrollado por el MIT. [15]

Como muchos otros lenguajes de desarrollo “moderno” su funcionamiento está ampliamente ligado a la interacción con un accionador, ya sea físico o virtual, lo que denominaríamos Eventos. Cabe destacar, que Scratch es un lenguaje orientado a objetos, característica legada de JavaScript, su lenguaje base.

Podemos definir un objeto en este contexto como una entidad que tiene un “estado”, en forma de atributos, un “comportamiento”, en forma de rutinas, y una “identidad”, siendo esta última una propiedad que diferencie un objeto del resto.

En lo que respecta a sus Capacidades de Datos, es capaz de operar con Variables, en forma numérica, y Listas en forma de estructura alfanumérica.

Por otro lado, a diferencia de otros lenguajes, cuyo objetivo o forma de trabajar es la integración del programa dentro de la memoria del dispositivo, con Scratch el programa se realiza y ejecuta dentro de un entorno dado en un Ordenador Personal (PC), lo que permite un cierto grado de interacción con el sistema huésped (PC).

Dentro del mundo de Scratch, existe una versión denominada ScratchX, que es una plataforma creada para funcionar con funcionalidades experimentales para Scratch, y por lo general programadas en otros lenguajes.

Aunque nos refiramos a Scratch como un lenguaje de programación, en el sentido estricto no lo es, al ser más cercano a un lenguaje intérprete.

3 Trabajo empírico

En este apartado abordaremos los aspectos técnicos derivados desde la iniciación y configuración del robot hasta el desarrollo de actividades con el mismo. Además, se realizará una explicación exhaustiva de las funcionalidades nativas de Scratch, así como de las añadidas por el Módulo Experimental de Aisoy.

3.1 Robot Aisoy

3.1.1 Iniciación y Conexión de Aisoy1

Antes de empezar a usar el Aisoy1, es necesario crear una tarjeta *microSD* con el software Airos en ella. Esta tarjeta, deberá ser insertada en la bahía de *microSD* de la Raspberry Pi del robot, situada en la parte trasera lateral del mismo. Este software es el que permitirá al robot contar con un *Sistema Operativo* para su uso, permitiendo la iniciación del mismo.

El robot Aisoy1 es capaz de funcionar tanto con conexión *wi-fi* como por conexión cableada con *Ethernet*, por medio de la placa Raspberry Pi. En ambos casos, existen una serie de limitaciones y ventajas, que dependerán del uso que queramos otorgarle al robot.

El siguiente listado comparativo, representa algunas de las ventajas e inconvenientes de ambos métodos de conexión.

Wi-Fi	Ethernet
Sin Cables. Movimiento Libre.	Cableado. Movimiento Restringido.
Requiere Configuración Adicional.	No requiere configuración Adicional.
Existe Posibilidad de Mala Conectividad (Baja Señal).	Conectividad Ligada al Estado de la Red.
Puede Trabajar de Manera Autónoma con un equipo gestor.	Requiere de Infraestructura de Red.

Tabla 2. Diferencias Wi-Fi vs Ethernet

A la hora de realizar la Iniciación, en general sigue de manera aproximada el flujograma mostrado en la siguiente página:

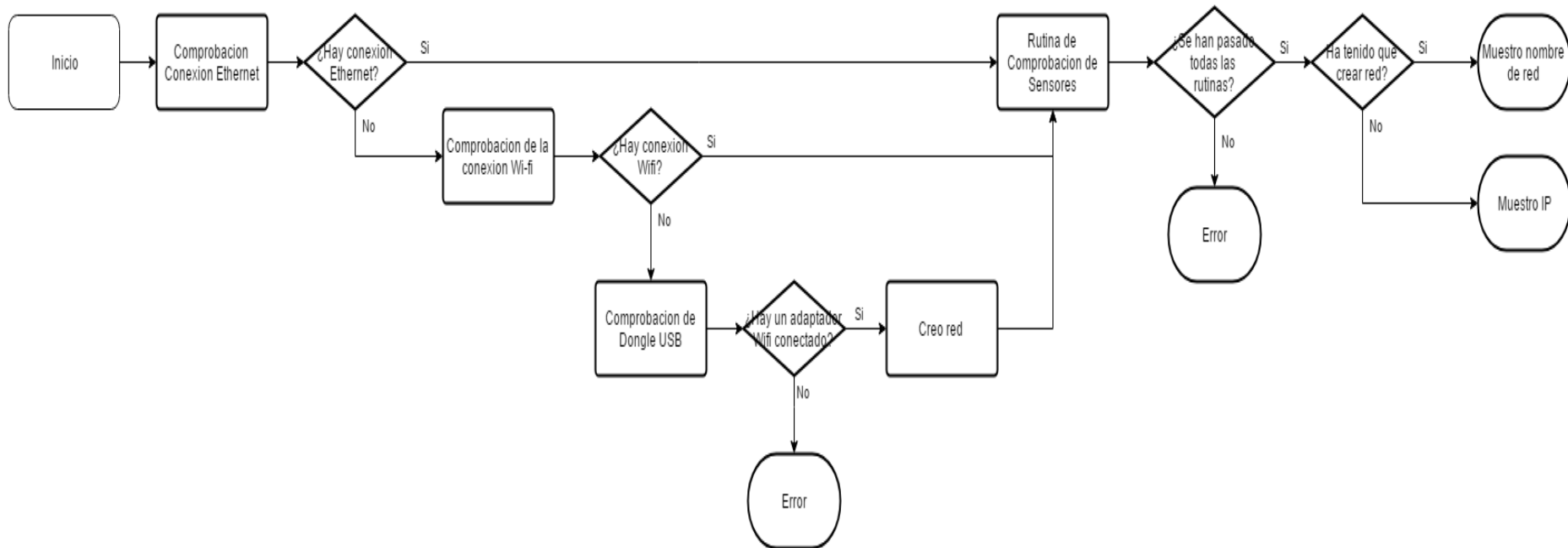


Figura 4. Flujograma de Iniciación del Aisoy1 versión 5

En el caso de querer la conexión por cable (*Ethernet*), la iniciación es directa, sin requerir de ninguna configuración adicional, siempre y cuando la red en cuestión tenga DHCP activo (lo más normal).

Por otro lado, tenemos la posibilidad de conexión inalámbrica (*wi-fi*). Dentro de este conexionado podemos encontrar dos opciones, conectarlo a una red o que el mismo cree una red.

En el primer caso, deberemos de apoyarnos en el *software de gestión Airos*, o en el caso de versiones anteriores en *Botapps*, mediante el mismo podremos descubrir y gestionar los robots Aisoy1 disponibles a nuestro alcance. Con este software, podremos indicarle al robot que se conecte a la red de nuestra elección, indicándole las credenciales oportunas. Una vez que el robot ha recibido las credenciales, y ha contestado al software gestor, procederá a realizar una inicialización de nuevo.

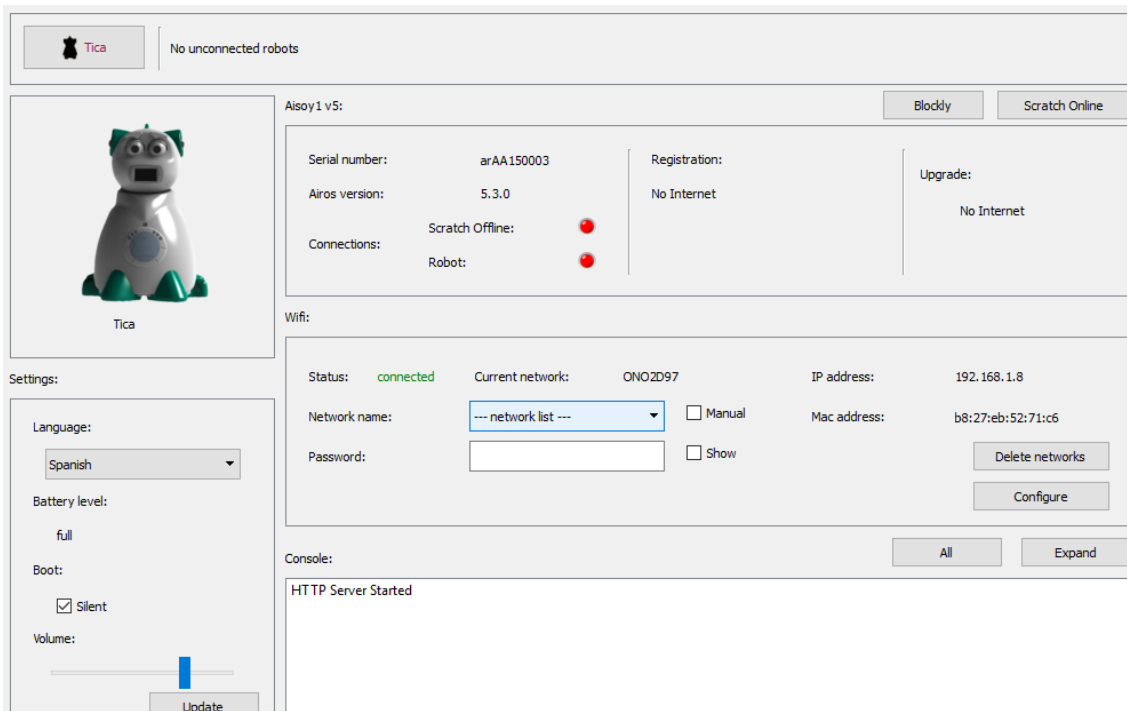


Figura 5. Airos con Robot configurado.

Por defecto, si no dispone de red configurada, el Aisoy1 crea una red autónoma con seguridad mediante un Dongle USB de estar conectado. Esta red permitirá un conexionado simple y directo, con el inconveniente de que, sin configuración adicional externa, no tener acceso a internet. Las credenciales de esta red son:

SSID: ai1xxxx / AisoyAP

Contraseña: "emotions"

Independientemente del método de conexión elegido, para poder operar con el entorno de trabajo, deberemos de habilitar la conexión del dispositivo gestor con el robot, realizando una "llamada" a la dirección IP asignada al robot y al *puerto 9090*.

Por ejemplo, en el caso de conexión mediante la red creada por el propio robot, esta dirección sería:

https:// 192.168.33.1:9090

O en el caso de la imagen anterior:

https://192.168.1.8:9090

Tras introducir esta IP, es posible que en algunos navegadores aparezca un mensaje de conexión no segura. En el caso de que aparezca, se deberá obviar el mismo y continuar, tras lo cual aparecerá la siguiente pantalla, indicando que la habilitación ha sido realizada correctamente.

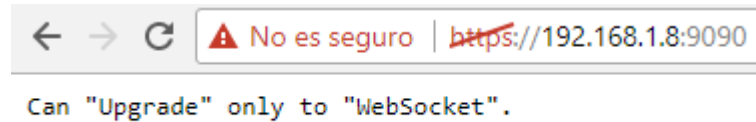


Figura 6. Pantalla WebSocket.

Por otro lado, existe la posibilidad de conectarse con la Raspberry del robot mediante SSH, lo que permitiría a un usuario avanzado desarrollar acciones no posibles de otro modo.

Mediante *Linux* o *MacOS* esta conexión se realizaría con una sintaxis similar a la siguiente en la línea de comandos:

ssh pi@<IP>, con la contraseña Raspberry

Para *Windows* sería posible realizar la conexión mediante programas del estilo *PuTTY*.

3.1.2 Entorno ScratchX

Aunque Aisoy1 ofrece la posibilidad de trabajar en diferentes lenguajes de programación, pudiendo elegir entre C++, Python y Scratch, hemos optado por empezar a trabajar con este último, debido a la simpleza que conllevaría el aprender su manejo para una persona con conocimientos informáticos de nivel medio.

Además, indicar que, aunque en cuanto a funcionalidades son similares todos ellos, existen rutinas que se simplifican en unos, y se vuelven por el momento imposibles de realizar con otros. Por ejemplo:

- Aplicaciones con redes neuronales, “fácil” de implementar mediante Python, pero prácticamente imposible mediante Scratch.
- Aplicaciones con Eye Tracking, posible de implementar en C++ y Python, pero difícil en Scratch.

Sin embargo, existe una funcionalidad clave en uno de ellos, Scratch es capaz de funcionar junto a *Python, SQL, Java, PHP* y *Flash*.

Aunque con Scratch existan una serie de limitaciones, debido a la simpleza de implementar subrutinas básicas, y gracias al proyecto experimental facilitado por la empresa, es el que consideramos óptimo para que un o una terapeuta sea capaz de desarrollar actividades para sus pacientes.

Por otro lado, una vez explicadas nuestras motivaciones para optar por un estilo de programación, gráfico en este caso con Scratch, queda definir el entorno de programación.

En este caso, para acceder al mismo disponemos de dos opciones: *offline* mediante la aplicación pertinente u *online* mediante una página web. En cuanto a funcionalidad, ambos entornos son iguales; sin embargo, no existe compatibilidad entre los desarrollos efectuados en una versión con la análoga. Debido a las posibilidades de conexión que existen hoy en día, hemos optado por centrarnos en la versión online de Scratch.

Aunque se ha indicado que se empleará Scratch como base de programación, puesto que debemos emplear las funcionalidades experimentales facilitadas por los desarrolladores del robot, se va a hacer uso de ScratchX, debido a la ya mencionada posibilidad de emplear extensiones con funcionalidades no nativas de Scratch.

Para usar Scratch con el proyecto con estas funcionalidades debemos acceder a la siguiente URL:

http://scratchx.org/?url=http://aisoy.github.io/ScratchX/aisoy_extension_airos5.js#scratch

Cabe destacar, que, para poder emplear la versión online, es requisito que el sistema gestor admita el empleo de JavaScript.

Tras acceder, el usuario se encontrará una interfaz similar a la siguiente.

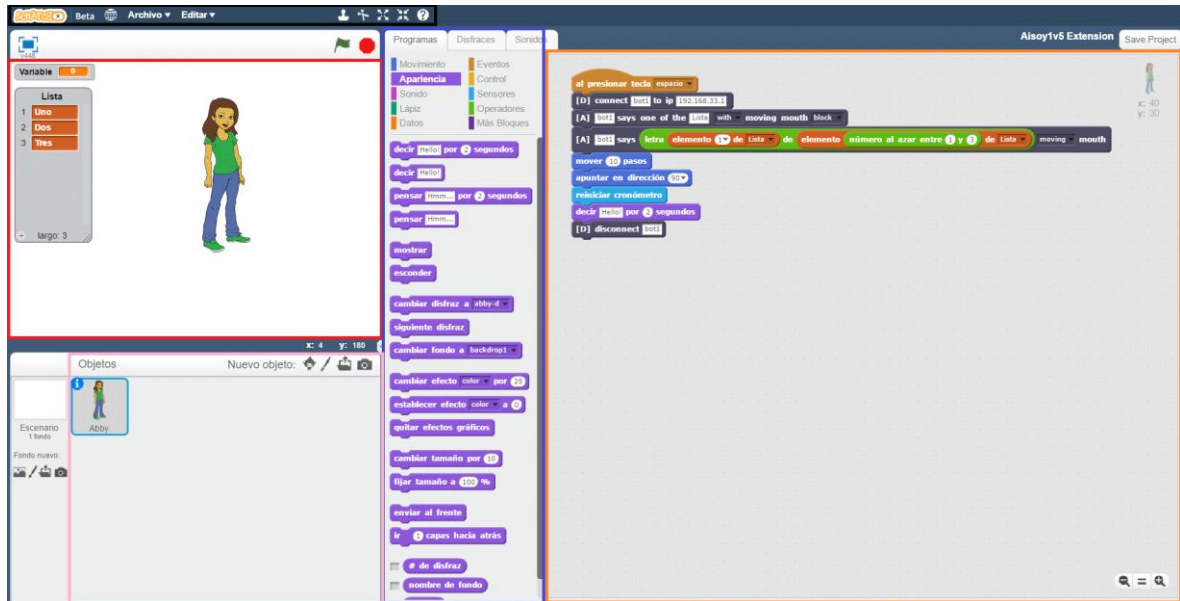


Figura 7. Espacio de Trabajo Scratch Web.

En la imagen se han demarcado una serie de áreas.

Primero, en color negro, en la zona superior izquierda, encontramos la barra de opciones típica de cualquier programa. Mediante la misma, es posible cambiar el idioma, guardar, cargar o incluso grabar la escena actual. También podríamos incluir en esta zona los accionadores de Comienzo, la bandera verde y de Parada, el octógono rojo.

Justamente debajo, marcado en rojo, encontramos la zona de escenarios, donde podemos ubicar las variables, listas y objetos a implementar.

Debajo de esta área, marcada en rosa, encontramos el espacio de Objetos, mediante el cual cabe la posibilidad de gestionar los objetos que queremos que aparezcan en nuestras escenas.

En la parte central de la interfaz, en color azul, encontramos la lista de bloques/instrucciones, mediante las cuales se podrá implementar las rutinas deseadas. Estas mismas instrucciones están separadas por grupos, en función de su funcionalidad. Además, esta funcionalidad vendrá indicada por la forma del bloque.

Finalmente, en la parte derecha de la interfaz, señalizada con color naranja, se encuentra el área de programa o espacio de trabajo. En el mismo, mediante bloques se podrá implementar las rutinas base para accionar al robot. El código por bloques, se puede ligar a un objeto, o al escenario.

3.1.3 Funcionalidades Scratch.

En el apartado anterior se han citado una serie de conceptos que deben ser definidos dentro del contexto de *Scratch*. A continuación, se describirán los que se consideran críticos para seguir.

- Escenario: Área gráfica de implementación de datos y figuras.
- Bloque: Representación gráfica de una instrucción.
- Objeto: Figura gráfica.
- Variable: Valor numérico cambiante en la memoria/escenario de Scratch.
- Lista: Estructura que permite almacenar diferentes fragmentos de información. Es lo que en otros lenguajes se denomina *Array*.

Una vez definidos estos elementos, es conveniente indicar que existen una serie de diferencias en algunas estructuras al resto de lenguajes. Para una comprensión completa y veraz de la manera de trabajo con Scratch, se procederá a explicar en rasgos generales los comandos disponibles en el mismo, indicando en los casos que sea oportuno las diferencias ya citadas.

A la hora de definir la interfaz, se ha indicado que los comandos están ordenados por grupos. Estos grupos están formados alrededor de una funcionalidad específica. La imagen siguiente muestra esos mismos grupos.



Figura 8. Pestañas de Bloques.

Tal y como se aprecia, existen diez grupos diferentes. Como ya se había adelantado, estos grupos tienen un foco particular en cuanto a su funcionalidad. En la siguiente tabla se define de manera simple el concepto base de cada grupo.

Grupo	Concepto
Movimiento	Posicionamiento de objetos.
Apariencia	Modificar objetos y escenario.
Sonido	Implementar y modificar efectos sonoros.
Lápiz	Herramienta de dibujo ligado a objetos.
Datos	Gestión de variables y listas.
Eventos	Activadores de código.
Control	Estructuras condicionales.
Sensores	Relacionado con sensores del equipo gestor.
Operadores	Operaciones aritméticas y lógicas.
Mas Bloques	Bloques creados "propios".

Tabla 3. Grupos de bloques Scratch

Tras presentar el concepto base de los bloques, a continuación, se procederá a explicar los grupos con un poco más de detalle.

Movimiento: Este grupo recoge los bloques relacionados con el movimiento y posicionamiento de Objetos entorno a la escena.

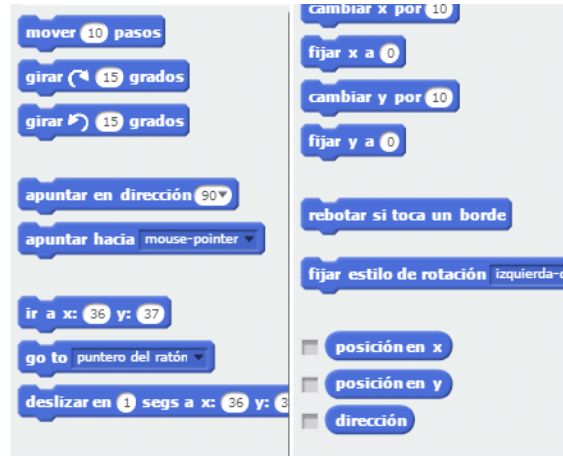


Figura 9. Pestaña Movimiento.

Los objetos son figuras graficas que se pueden mover y hasta cierto punto interactuar entorno a un escenario u otros objetos.

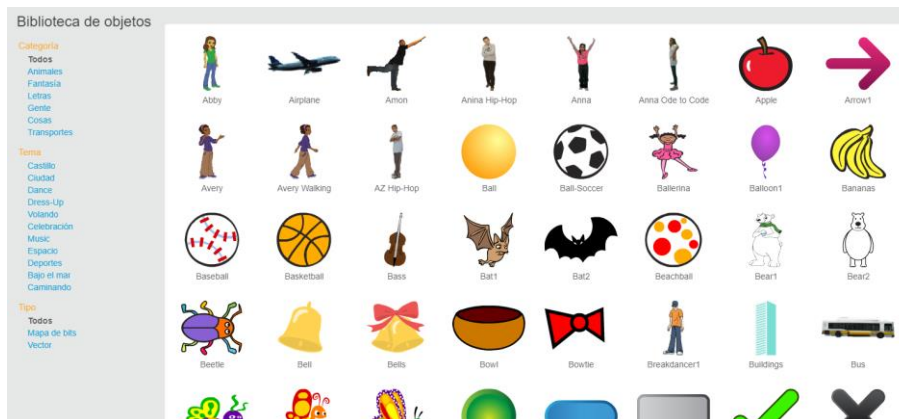


Figura 10. Biblioteca de Objetos.

Por defecto, Scratch ofrece una gran cantidad de objetos ya incluidos. Sin embargo, es posible crear objetos nuevos, o importarlos desde una imagen ya creada.

Respecto a las posibilidades de movimiento, las acciones disponibles se componen de:

- Mover un cierto número de pasos en línea recta.
- Rotar la figura un ángulo concreto tanto en sentido horario como anti-horario.
- Asignar una rotación particular al objeto.
- Relocalizar el objeto mediante bloques.
- Activar los bordes rebotantes, es decir, que cuando llegue al borde del escenario visible cambie su dirección.

En los bloques mostrados anteriormente, existen tres con una forma diferente, con bordes redondeados. Estos bloques son lo que podríamos denominar como *bloques de datos*. En este caso, esos datos indican la posición actual y rotación absoluta de la figura en cuestión, y solo podrán ser empleados en el código vinculado directamente a esa figura.

Apariencia: En este grupo se encuentran una serie de acciones o efectos realizables a los Objetos o el Escenario.

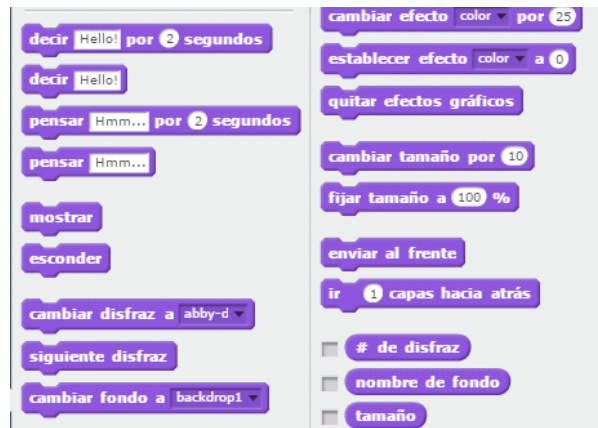


Figura 11. Bloques de Apariencia.

Acciones relacionadas con los objetos:

- Mostrar bocadillos de dialogo y pensamiento.
- Activar o desactivar la visibilidad del objeto.
- Modificar (Sustituir) la apariencia (disfraz) de un objeto.
- Aplicar Efectos (Filtros) determinados.
- Cambiar o Asignar un tamaño a un objeto.
- Modificar la capa de acción del objeto en el escenario.

En relación a las acciones relacionados con el escenario son más limitadas, permitiendo el cambio del fondo visual del mismo, así como la aplicación de Efectos Gráficos sobre el mismo.

Al igual que con los bloques de Movimiento, existe una serie de bloques de datos en relación a este grupo. Estos bloques son el número de disfraz actual de un objeto, el nombre y número del fondo actual y el tamaño actual porcentual del objeto.

La siguiente imagen muestra un ejemplo de acciones realizables mediante este Apartado.

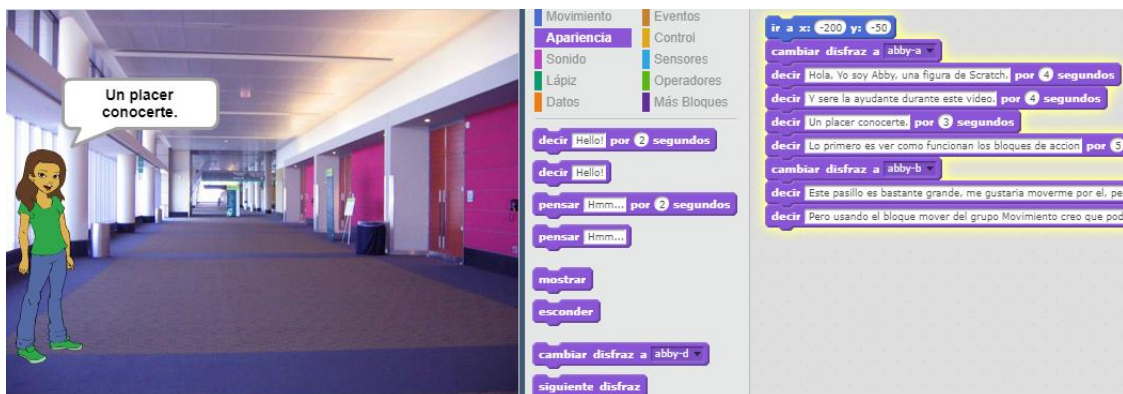


Figura 12. Ejemplo de Bloques de Apariencia.

Sonido:

En la pestaña de Sonido, podemos encontrar bloques que permiten controlar aspectos relacionados con la sonorización en el equipo gestor.

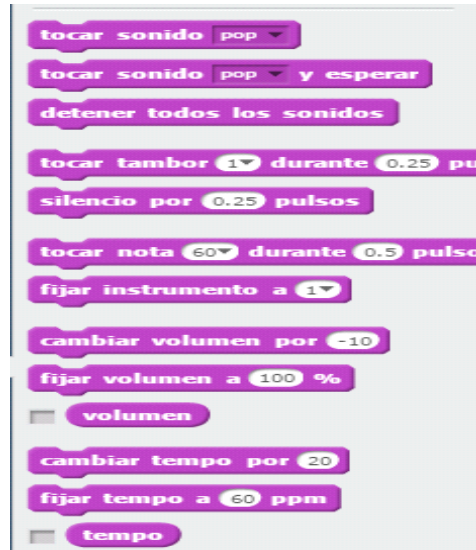


Figura 13. Bloques de Sonido.

Entre las acciones disponibles con estos bloques se incluye:

- Reproducir un Sonido.
- Grabar un Sonido.
- Cambiar el volumen
- Cambiar la temporización de reproducción (pulsos por minuto)

Con Scratch se tiene la posibilidad de emplear un número notable de sonidos diferentes de base, incluyendo diversos instrumentos, permitiendo componer bases sonoras de carácter simple.

Si bien el empleo de los mismos puede ser interesantes en tareas relacionadas con la interacción en el equipo gestor, sus posibilidades de uso con el robot son relativamente limitadas, sirviendo por ejemplo como avisos de la ocurrencia de un evento determinado.

Lápiz:

La pestaña de Lápiz, permiten controlar las características de Lápiz de *Scratch*. Según la Wiki de *Scratch*, esta es una de las categorías menos empleadas.

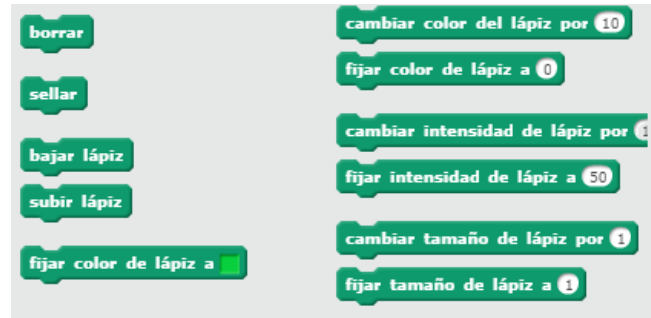


Figura 14. Bloques de Lápiz.

El Lápiz, es una funcionalidad que permite el empleo de una utilidad de dibujo, como si dibujásemos sobre un papel, salvo que está ligado al movimiento programado de un objeto.



Figura 15. Ejemplo de Funcionalidad Lápiz.

En el ejemplo anterior, el lápiz va dibujando la trayectoria de la pelota a tramos alternos.

En lo relacionado con sus funcionalidades, podemos:

- Borrar el dibujo realizado por el objeto.
- Empezar a dibujar (bajar el lápiz).
- Dejar de dibujar (subir el lápiz).
- Cambiar el color/espesor/intensidad del lápiz.

Su utilidad con respecto a la programación para el Aisoy1 se ha considerado bastante limitada, aunque posible si implementásemos rutinas que requiriesen la interacción con el terminal gestor.

Datos:

En esta pestaña, se encuentran los bloques relacionados con las funciones de almacenamiento de Información. Anteriormente se han citado dos estructuras, la variable y las listas. Ambas se comportan igual manera que los bloques de datos descritos anteriormente, con la excepción que los creados en esta pestaña, pueden ser nombrados y modificados a conveniencia del usuario.

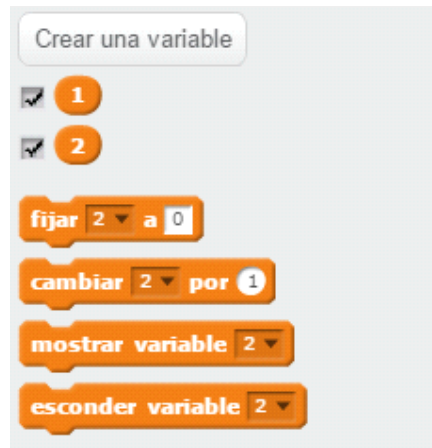


Figura 16. Bloques de Variable.

Respecto a las variables, que recordamos que son bloques de datos de carácter numérico, las operaciones que se pueden realizar son:

- Crear Variables.
- Fijar el Valor de la Variable.
- Incrementar o Decrementar una cantidad fija.
- Mostrar/Ocultar la variable.

Las *Variables* son útiles para almacenar información que podamos permitirnos guardar en formato numérico, ya sea de manera directa, por ejemplo, el número de veces que el código ha pasado por un punto en particular; o como codificación, por ejemplo, para un estado particular del código.

Tras crear una variable, esta será mostrada en el escenario. Si pulsamos dos veces sobre la misma, podremos cambiar el modo de visualización entre el por defecto, visualización sin nombre o visualización con control deslizante, siendo este último, el único modo que permite la interacción con la variable desde el escenario.

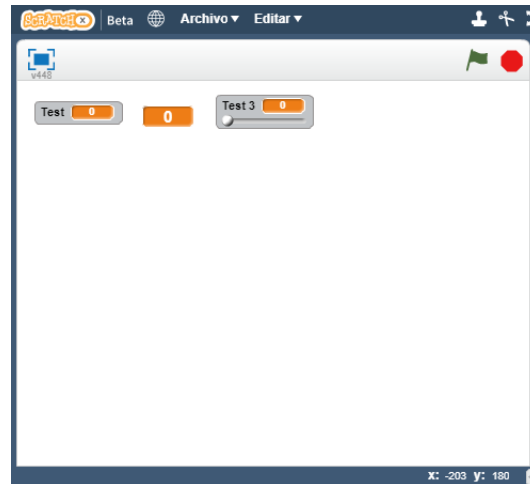


Figura 17. Variables en el Escenario.

Por otro lado, tenemos las listas, estructuras de datos de carácter alfanumérico.

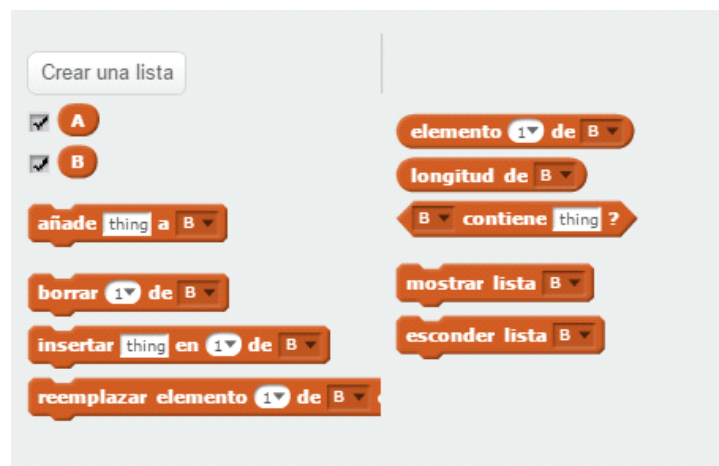


Figura 18. Bloques de Listas.

Podemos realizar acciones similares a las descritas con las variables, siendo las mismas:

- Creación de Listas.
- Añadir/Eliminar Elementos.
- Modificar Elementos.
- Consultar Elementos.
- Consultar el número de elementos y si contiene un elemento específico.
- Mostrar/Ocultar Listas.

Aunque el propósito general de ambas estructuras es similar, almacenar información, su diferencia radica en la forma y tipo de información que pueden guardar. Mientras que la variable solo puede guardar información en formato numérico, con las listas se puede guardar información

alfanumérica, y por extensión palabras, permitiendo agruparlas en las referidas listas para procesarlas como resulte conveniente.

De igual manera que las variables, estas podrán ser visualizadas en el escenario. Además, a diferencia de las variables, podrán ser modificadas directamente desde el escenario.

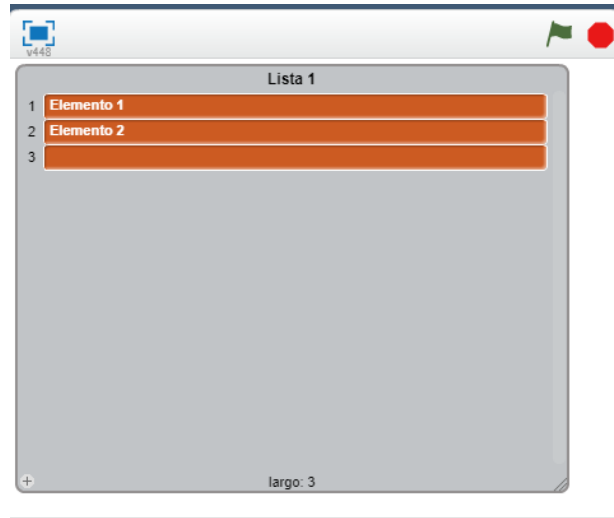


Figura 19. Listas en el Escenario.

Eventos:

En este grupo se encuentra los bloques relacionados con el accionamiento de eventos.



Figura 20. Bloques de Evento.

Un evento es aquella acción que permita activar de manera condicional un segmento aislado del código, por ejemplo, una subrutina. Las acciones que pueden activar estos eventos son:

- Al accionar el botón de comienzo (botón de la bandera verde).
- Al presionar una tecla.
- Al presionar en el escenario.
- Al cambiar el fondo.
- Condicionado al Volumen, Tiempo de Ejecución o Movimiento.
- Condicionado a Flags (mensajes).

Los Mensajes podrían definirse como pseudo-variables booleanas cuya función podría ser, por poner un ejemplo, activar un sub-código determinado en función de una decisión condicionante.

Estos se crean en esta misma pestaña, cuyo envío se puede realizar de manera inmediata, en cuyo caso si hubiese más código a continuación se seguiría ejecutando directamente, o con interrupción, en cuyo caso la ejecución del código subsiguiente se pausaría hasta que se completase el código receptor del mensaje.

Por el modo de funcionamiento de Scratch, es posible realizar código con funcionamiento pseudo-paralelo empleando dos eventos iguales.

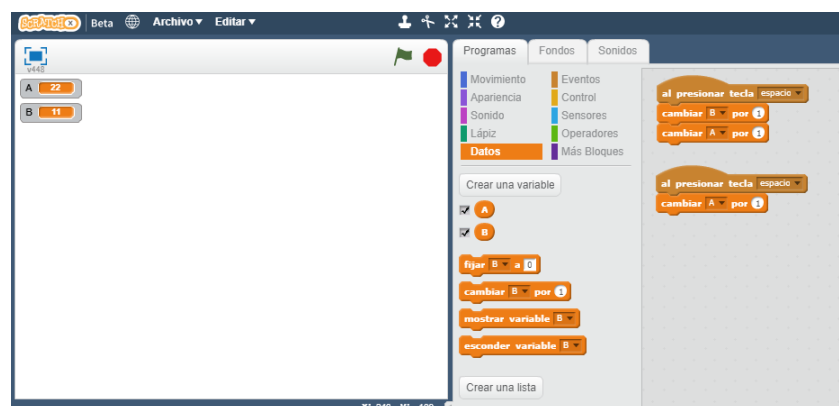


Figura 21. Código Pseudo-Paralelo.

Control:

Los Bloques de Control son lo que, en muchos lenguajes, se conoce como estructuras condicionales.



Figura 22. Bloques de Control.

Las estructuras condicionales son métodos comparativos para determinar la acción a realizar a continuación, en nuestro caso en función del código. Estos métodos pueden ser simples o múltiples.

Como métodos simples podemos encontrar:

- If, en cuyo caso ejecutará SI se cumple la condición.
- Repeat <X times>, que repetirá la acción (código) un número fijo de veces.
- Wait Until, que detendrá el código hasta que se cumpla la condición.
- Repeat Until, que repetirá el código hasta que se cumpla la condición, tradicionalmente se la conoce como Do/While.

Aunque como métodos múltiples se conoce una serie extensa de estructuras, en Scratch de base solo se encuentra la más simple, el If/Else, que ejecutará un código u otro en función de si el condicionante es verdadero o falso.

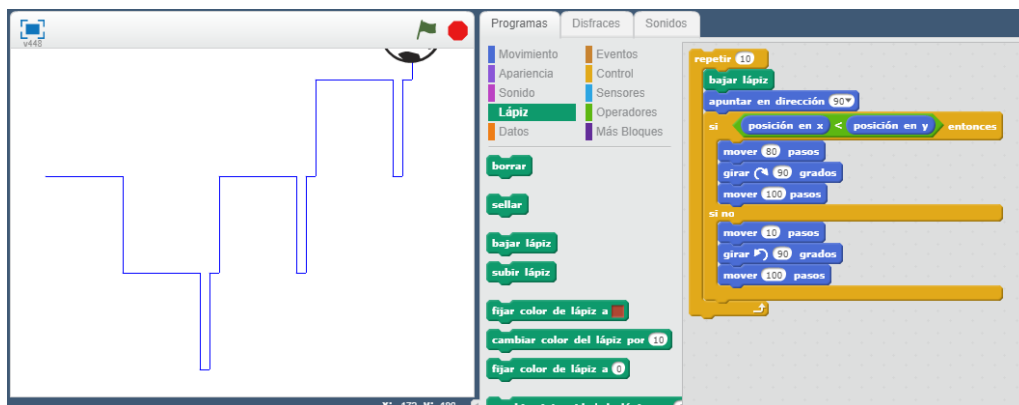


Figura 23. Ejemplo de Código Condicional.

Cabe destacar que la condición condicionante por lo general se consideran valores booleanos, es decir verdadero o falso. Además, en esta pestaña se abarcan los bloques de clonación de objetos y parada de código, ya sea parcial o total.

Sensores:

En la pestaña de Sensores encontramos bloques relacionados con la Interacción con el Usuario y el equipo gestor.

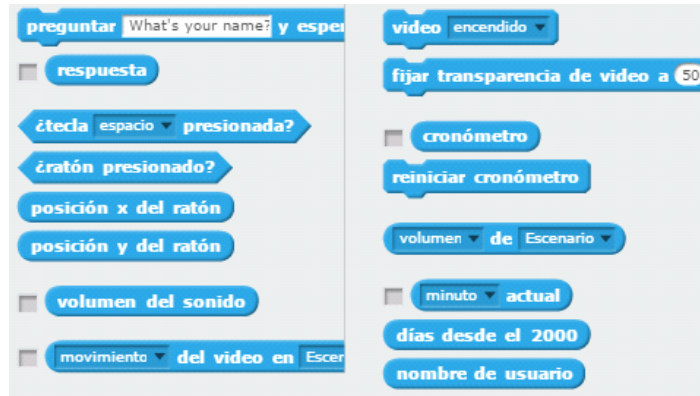


Figura 24. Bloques Sensores.

Dentro de esta pestaña, se puede destacar una serie de bloques en particular, aquellos relacionados con el tiempo. Entre estos bloques, encontramos el bloque/variable “<minuto/hora/día> actual”, el cual nos facilita la información sobre el tiempo en relación al sistema gestor.

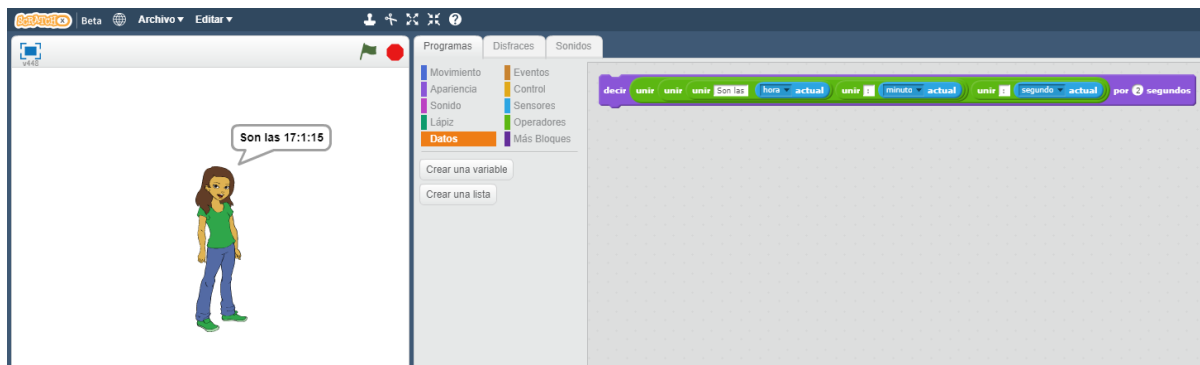


Figura 25. Funciones temporales.

Además, podemos encontrar el cronómetro, que nos facilita el tiempo transcurrido desde la iniciación del mismo.

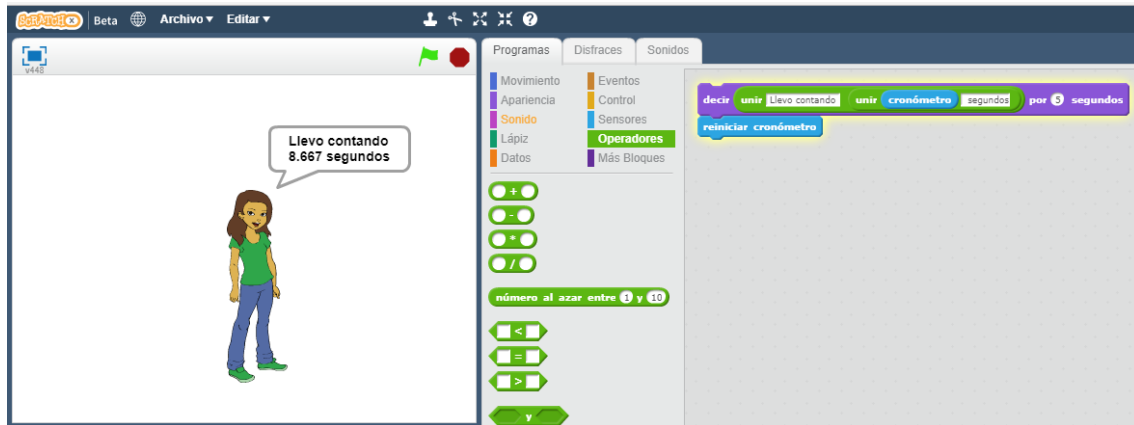


Figura 26. Funciones de Cronómetro.

Operadores:

En el Bloque de Operadores, se pueden encontrar operaciones de carácter lógico-matemático.



Figura 27. Bloques Operador.

Entre los mismos, podemos encontrar multitud de operaciones Aritméticas y Lógicas.

Operadores Aritméticos:

- Suma/Resta/Multiplicación/División.
- Random.
- Join/Split.
- Función modulo.
- Longitud.
- Round.
- Funciones varias (sqrt cos sin log exp).

En lo que respecta a las funciones aritméticas, por lo general, Scratch trata a los bloques como variables, como denotan sus bordes redondeados.

Operadores Lógicos:

- Mayor/igual/menor que.
- AND/OR/NOT-

Al contrario que los operadores Aritméticos, las funciones Lógicas, las trata como condiciones y por lo tanto como valores booleanos.

Más Bloques: En este grupo se encontraría aquellos bloques desarrollados de manera experimental, siendo en nuestro caso aquellos propios del robot. La naturaleza de los mismos se explicará en el siguiente apartado.

3.1.4 Funciones Básicas de Aisoy1.

A la hora de interactuar con el Aisoy1, es evidente que Scratch por sí mismo no nos proporciona los mecanismos necesarios para trabajar con el robot, sino que nos proporciona una serie de herramientas de carácter general para sintetizar código base. Para esos mecanismos, el fabricante facilita la extensión experimental ya citada, que permite la comunicación e interacción con el robot. Esta extensión consta de una serie de bloques adaptados de su extensión de BYOB. Los bloques componentes de dicha extensión, encontrados en el grupo de “*Más Bloques*”, se podrían clasificar en *Configuración*, *Movimiento*, *Evento*, *Comunicación* y *Emocional*.

Bloques de Configuración:

Los Bloques de configuración sirven para inicializar o modificar la configuración de un robot.

Bajo esta clasificación podemos encontrar los bloques *Connect*, *Disconnect*, *Set/language*, *Grammar* y *ASR Recognition*. A continuación, se definirán estos mismos.

Bloque Connect

El bloque connect permite iniciar la comunicación del equipo gestor con un robot.



Figura 28. Bloque Connect.

En el mismo encontramos dos campos. El primero permite asignarle una etiqueta a el robot, mientras que el segundo sirve para indicar la dirección única del robot asignada. Antes de poder interactuar con un robot, es obligatorio realizar una acción de conexión entre el mismo y el equipo gestor.

Una vez asignada una etiqueta a una IP, todas las acciones/bloques con ese mismo identificador serán transmitidas al robot en cuestión.

Esto además posibilita la interacción con más de un robot al mismo tiempo, asignándoles dos identificadores diferentes.

Bloque Disconnect

Análogamente al bloque connect, se encuentra el bloque disconnect. Este permite finalizar la conexión de un robot con el equipo gestor.

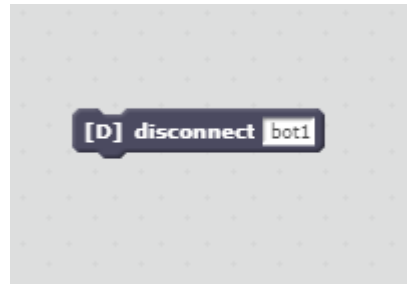


Figura 29. Bloque Disconnect.

La llamada al robot para desconectar se realiza mediante la etiqueta previamente asignada con el bloque connect.

Bloque Set/Language

Por otra parte, está el Bloque Set/Language, mediante el cual podemos configurar el idioma y género de la Voz Empleada por el robot para las funcionalidades de TTS y ASR.



Figura 30. Bloque Set/Language.

En este bloque encontramos cuatro campos:

- El primero, define el robot para dirigirle la acción mediante su etiqueta. Este mismo campo lo encontraremos en la mayoría de bloques de los facilitados por Aisoy.
- El segundo campo, es un menú desplegable que sirve para definir si queremos modificar solo la configuración ASR, TTS o Ambas. En ocasiones, es probable que queramos que el robot entienda en un idioma, pero se exprese en otro distinto, como puede ser en ejercicios de traducción.
- El tercer campo permite elegir el idioma entre inglés, francés, catalán y español para la característica seleccionada en el segundo campo. En relación a los idiomas, cabe destacar que el idioma ingles es el que mejor compatibilidad presenta con respecto a la utilización

de ASR, mientras que existen ciertos fallos de captación en el resto de idiomas, especialmente en francés y catalán.

- Mediante el cuarto campo es posible modificar el género en el que se expresa mediante TTS o debe esperar recibir las secuencias ASR el Aisoy1.

En relación a las características ASR del robot encontramos dos comandos base de configuración adicionales. El *ASR Recognition* y el *Grammar*.

Bloque ASR Recognition

En primer lugar, el bloque *ASR Recognition* permite activar o desactivar el Reconocimiento de Voz.



Figura 31. Bloque ASR Recognition.

En este bloque podemos encontrar dos campos además del identificador. Mediante el primero, podremos elegir si *activar o desactivar la funcionalidad en sí*. Más adelante, se explicará la importancia que puede llegar a tener esta característica.

El ultimo campo de este bloque permite que la rutina se detenga hasta la adquisición de un reconocimiento correcto, o por el contrario se permita continuar con la misma tras ejecutar el bloque.

Bloque grammar

Finalmente, en relación a los bloques de configuración, está el bloque Grammar. Mediante el mismo, podemos asignar un conjunto de palabras o frases, mediante listas, que queramos que el robot en cuestión reconozca.



Figura 32. Bloque Grammar.

Consta de un campo identificador y un campo para asignar la lista en cuestión.

Es importante crear, rellenar y asignar la lista en cuestión, puesto que de otro modo el robot no sería capaz de reconocer las palabras que queramos, lo que imposibilitaría la utilización de eventos por reconocimiento de voz.

Bloques de Movimiento:

Otra serie de bloques encontrados son los bloques de movimiento. Estos bloques, inicialmente no encontrados en la extensión experimental, aparecen tras conectar un robot Aisoy1 con capacidades motoras. Los mismos se componen de cinco bloques, moves forward, moves backward, rotate left, rotate right y stops. El funcionamiento y estructuración de los cuatro bloques moves es similar entre los mismos, por lo que para la explicación se juntaran en un solo apartado, definiendo las diferencias de funcionamiento entre unos u otros.

Bloques moves y rotate

Los bloques moves permiten controlar los servos relacionados con el movimiento físico del robot.



Figura 33. Comandos de movimiento

A continuación, se ofrece una tabla con las diferencias de funcionamiento entre los diferentes bloques.

Bloque	Acción
Moves Forward/backward	Movimiento Adelante/Atrás.
Rotate Left/right	Giro Izquierda/Derecha sobre sí mismo.

Tabla 4. Funciones de Movimiento

En relación a su estructura, encontramos dos campos característicos en estos bloques, además del identificador de robot. Mediante el primer campo adicional, podemos indicar el tiempo que se desea que se mueva/gire en una dirección.

El otro campo es un menú desplegable mediante el cual podemos elegir la velocidad de movimiento o giro deseado entre lento, medio o rápido. Existen diferencias notables en las velocidades con y sin el cargador conectado, por lo que se ha de tener un cierto grado de precaución a la hora de asignar o ajustar el movimiento del robot.

Bloque stops

Por otro lado, está el bloque stops. El mismo permite detener de manera súbita el movimiento del robot.



Figura 34. Bloque stops

En el mismo únicamente encontramos el campo identificativo del robot que recibe la orden de parar.

Bloques de Evento:

Estos bloques, al igual que los bloques de evento de Scratch permite gestionar la inicialización de acciones, con la salvedad de que estos están orientados a los sensores del robot, no al equipo anfitrión.

Bloque When/touched

En primer lugar, el bloque When/touched permite la activación de código ligada a uno de los pulsadores del Aisoy1.

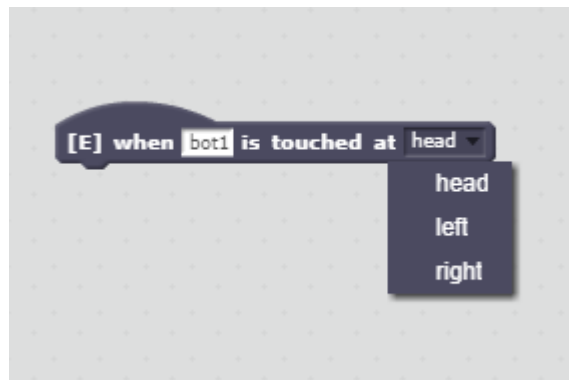


Figura 35. Bloque When/touched.

Su campo particular permite seleccionar el pulsador ligado al accionamiento de código, pudiendo elegir entre el encontrado en la parte trasera de la cabeza o los situados en los laterales del torso del robot.

Bloque When/position

Por otro lado, está el bloque When/position que permite ejecutar código en relación a la posición del robot, es decir ligado al acelerómetro del mismo.



Figura 36. Bloque When/position.

En el menú desplegable encontrado en el bloque, se puede seleccionar la posición que activará ese código, a elegir entre:

<u>Opción</u>	<u>Posición</u>
Left	Tumbado a la Izquierda.
Right	Tumbado a la Derecha.
Forward	Tumbado frontal.
Backward	Tumbado trasero.
Standup	De pie.
Facedown	Sostenido del revés.

Tabla 5. Opciones de Posición

Cabe destacar, que la posición en sí no activa el *Evento*, siendo el cambio a esa posición lo que lo activa.

Por otro lado, mediante el uso de *Variables* y *Condicionales*, sería posible ligar y restringir la ejecución de código entre posiciones, como podría ser que ejecute un determinado código si su posición previa era de pie u otro si era Tumbado en una determinada posición.

Bloque When/hears

El bloque *When/hears* permite activar código ligado a las funcionalidades ASR del Aisoy1.



Figura 37. Bloque When/hears.

Su campo particular, “sentence”, permite determinar la palabra o frase que activará el código subsiguiente.

Aunque se incluyan *Eventos When/hears*, es necesario configurar y activar la funcionalidad ASR para que sean correctamente detectados.

Bloque When/detects.

El bloque When/detects permite ejecutar código tras un evento básico de reconocimiento facial.



Figura 38. Bloque When/detects.

En su campo específico, permite designar el número de rostros exacto para activar el evento.

Aunque posee cierto grado de utilidad, en especial para ejercicios relacionados con la interacción social y/o actividades grupales, el potencial de esta funcionalidad se puede amplificar mediante la utilización de *Python* y Redes Neuronales, con mejoras como seguimiento facial o reconocimiento facial específico.

Bloque When/gets QR

Este bloque permite la implementación de la funcionalidad de Códigos QR del robot en el programa.

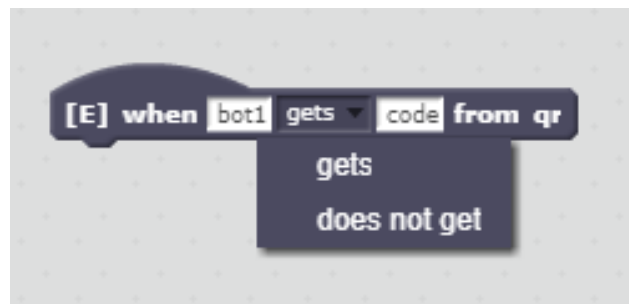


Figura 39. Bloque Evento Lectura QR.

Los Códigos QR, del Ingles *Quick Response*, se podrían definir como la evolución de los Códigos de barras convencionales, en el que se emplea una matriz bidimensional para almacenar la información.

Esta Matriz/Código se puede emplear con un dispositivo con capacidad QR, para redirigir de manera rápida a una aplicación específica, por ejemplo, a una página web. Hoy en día su funcionalidad ha evolucionado, permitiendo acciones como el cifrado/obtención de credenciales Wi-Fi automáticamente para un dispositivo o la redirección dinámica de las direcciones objetivo.

La siguiente imagen representa un código QR con la dirección <https://www.etsit.upv.es/> codificada en él.



Figura 40. Ejemplo de Código QR.

Sin embargo, no necesariamente se tiene que codificar una dirección, pudiendo realizar la acción con un texto dado, como puede ser una palabra o conjunto de las mismas o incluso con imágenes.

Volviendo a la explicación del bloque, el mismo posee dos campos particulares. El primero es una lista en la que se puede implementar la acción como activada al recibir el código, o mientras no se reciba. Por otra parte, en el segundo, podremos indicar el código QR que active el Evento.

Bloque When/covered

El bloque When/covered permite controlar rutinas ligadas a la obstaculización en la visión de la cámara del robot.

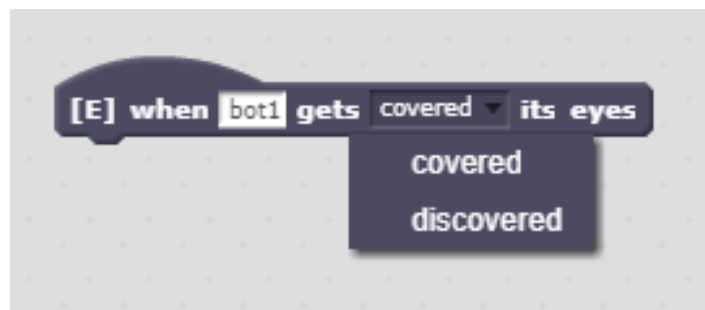


Figura 41. Bloque When/covered.

La activación del evento vendrá condicionada al modo de configuración del bloque, pudiendo ser al negar la visión (*covered*) o al recuperarla (*discovered*).

Bloque When/detects color

Por último, en relación a los bloques de evento, está el bloque When/detects color. El mismo permite la vinculación de rutinas a la detección de un color específico.

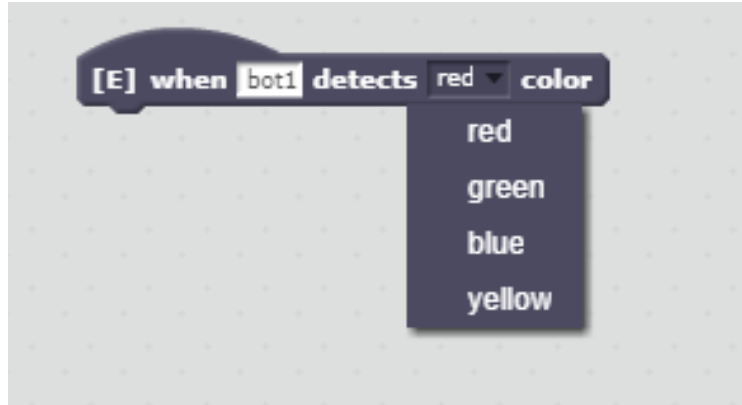


Figura 42. Bloque When/detects color.

La detección del color se hace mediante el rango visual del mismo, no mediante la caracterización RGB del mismo, siendo rojo, verde, azul y amarillo los colores disponibles para detectar.

Bloques de Comunicación:

Los bloques de comunicación son aquellos relacionados con las formas de expresarse que tiene el robot. Con esta catalogación podemos identificar tres bloques diferentes, los bloques Says y Says one of the list, relacionados con la funcionalidad TTS del Aisoy1 y el bloque Writes in mouth relacionado el funcionamiento de la pantalla en la zona labial.

Bloque Says

El bloque “Says” es uno de los elementos principales para ejecutar las funciones de TTS del Aisoy1.

El Text-to-Speech es una funcionalidad que permite reconocer y convertir una cadena de texto en sonidos. El reconocimiento y conversión están, por lo general, ligados a las reglas fonéticas de la lengua configurada, por lo que los barbarismos podrían ser pronunciados de manera errónea.

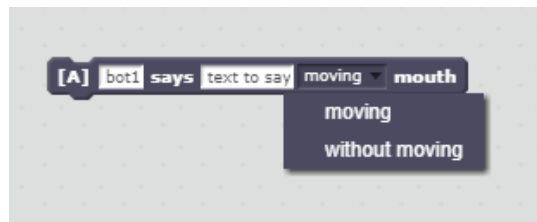


Figura 43. Bloque Says.

Este bloque está compuesto por tres campos.

- El primero, al igual que en la mayoría de bloques, indica el robot destino del comando.
- El segundo campo, es donde se introduce el texto que queremos que pronuncie el robot.
- Y el tercero y último, se refiere un menú desplegable para asignar si queremos que el robot haga una animación como si estuviese hablando, o simplemente reproduzca el texto.

Bloque Says one of the list

El bloque “Says one of the list” se comporta de manera similar al bloque “Says”, con la excepción de que, en lugar de repetir un texto simple, repite un elemento encontrado en una lista. Es especialmente útil en aquellas ocasiones que queremos dotar al robot de una personalización mayor a la hora de contestar a un impulso.

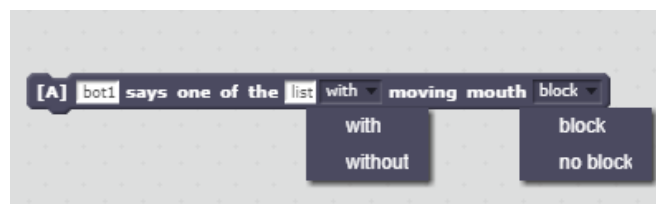


Figura 44. Bloque Says one of the list.

El bloque en sí está compuesto de cuatro campos.

- El primero para asignar un robot objetivo.
- El segundo para denotar la lista en la que se encuentra las palabras o frases que se desee reproducir.
- El tercero se refiere a la posibilidad de realizar una animación de gesticulación con el mensaje.
- El cuarto permite bloquear/pausar la ejecución de código hasta terminar el mensaje.

Bloque Writes / in mouth

El bloque “Writes / in mouth” permite controlar directamente lo mostrado en la pantalla bucal del robot. Si bien de manera básica este bloque sólo puede mostrar frases, y de una longitud limitada, es posible controlando el posicionamiento de los caracteres crear expresiones (dibujos) en formato ASCII.

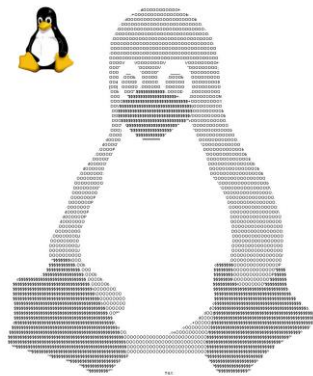


Figura 45. Ejemplo de Dibujo ASCII.

Un dibujo ASCII sería la representación de un objeto mediante los caracteres disponibles en un ordenador.



Figura 46. Bloque Writes/ in mouth.

La composición del bloque está formada por la asignación del robot objetivo, y el texto que queramos que se muestre en la pantalla.

Bloques Emocionales:

Los bloques Emocionales son aquellos que pueden relacionarse con la expresión anímica del robot.

Dentro de esta categoría, podemos encontrar seis bloques diferentes.

Por un lado, los bloques de tipo *Moves*, para controlar la posición de la cabeza y expresión facial del robot.

Por el otro, los bloques *heart light* y *State*, que servirán para definir el estado anímico del robot.

Bloques *Moves head horizontal/vertical*

El primero de los bloques de movimiento relacionados con la expresión emocional es el bloque *Moves head horizontal*. Como su nombre indica, permite controlar la posición de la cabeza en el plano horizontal.



Figura 47. Bloque *Moves head horizontal*.

La posición de la cabeza se define mediante un parámetro entre los valores 0, que indica la posición más a la izquierda y 1 que indica la posición más a la derecha. El valor intermedio 0.5 indicara la posición central en el plano horizontal. Los valores de giro máximo son de aproximadamente 45°, hacia ambos lados.

Además de la posición, es posible controlar la velocidad de movimiento a la nueva posición, a elegir entre despacio, media y rápida.

De forma análoga al plano horizontal, existe el bloque *moves head vertical*, que permite controlar la inclinación vertical de la cabeza.

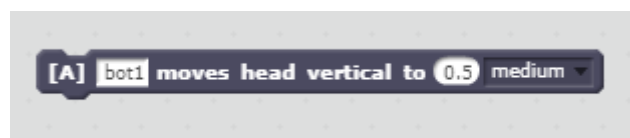


Figura 48. Bloque *Moves head vertical*.

La composición de este bloque es similar a la vista en el bloque anterior, definiendo el valor 0 para la posición a aproximadamente 45° por debajo del horizonte y 1 para 45° por encima del horizonte.

Bloque moves eyes

Mediante el bloque *moves eyes* es posible controlar la posición de los párpados del robot.

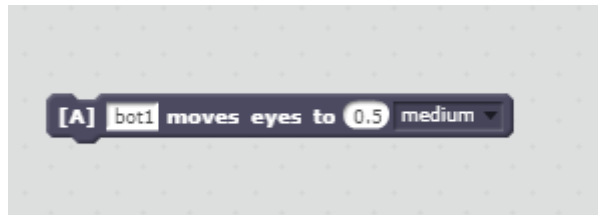


Figura 49. Bloque Moves eyes.

Su comportamiento y composición es similar al visto en los planos horizontal y vertical del movimiento de la cabeza, definiendo la posición del párpado con un valor entre 0, para el ojo cerrado y 1, para la posición con el ojo totalmente abierto.

Bloque moves eyebrows

Finalmente, con relación a los bloques de movilidad facial, está el *moves eyebrows*, que permite controlar la posición de las cejas del robot.

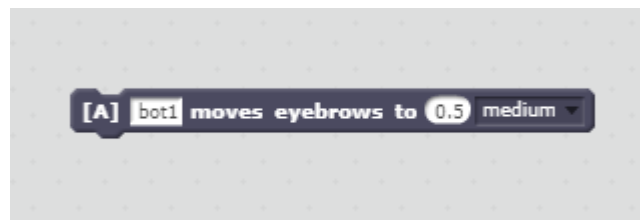


Figura 50. Bloque Moves eyebrows.

Al igual que el resto de bloques de movimiento facial, se compone de los campos de posición de la parte y la velocidad de movimiento de la misma.

No obstante, cabe destacar que esta parte es ligeramente menos robusta que el resto, y existe la posibilidad de daños en el robot si se realizan movimientos bruscos de la misma, especialmente con los valores límite de los servos. Para evitarlo, se recomienda dejar un margen de 0.1 puntos respecto a los valores límite.

Bloque heart light

Con el bloque *heart light*, es posible controlar el color de la luz pectoral del Aisoy1.



Figura 51. Bloque Heart light.

El color de dicha luz se controla mediante tres campos relativos a la intensidad de los tres colores primarios: rojo, verde y azul. Además, es posible controlar el tiempo mínimo de iluminación con dicho color.

Dicha intensidad se define mediante una codificación tridimensional conocida como RGB, del inglés: Red, Green & Blue. El modelo de color RGB define la intensidad de cualquier color como una composición de estos tres colores primarios. Así pues, se definiría cualquier color como una composición de tres puntos [Red, Green, Blue] contenidos entre dos valores. El caso más típico es entre 0 (hex: 00) y 255 (hex: FF).

Esto nos permitiría construir un espacio denominado “espacio del color RGB” que formaría un cubo con los diferentes colores asignados a unas coordenadas dentro de dicho espacio.

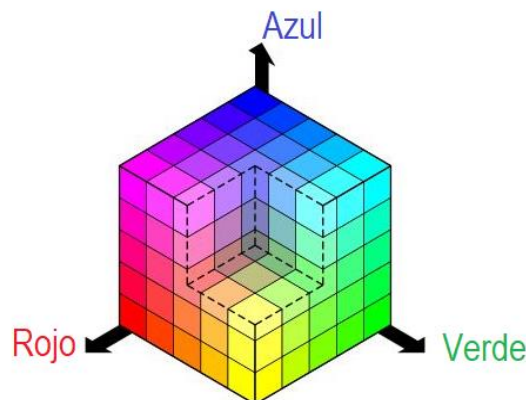


Figura 52. Representación del espacio RGB.

Con ello se definirían los tres colores primarios como $[255,0,0]$ en el caso del Rojo más intenso, $[0,255,0]$ para el verde y $[0, 0, 255]$ para el azul. El Amarillo en este caso sería $[255, 255, 0]$ y el Cian vendría a ser $[0, 255, 255]$. Los colores Negro y Blanco se definirían como la ausencia o saturación total de todas las intensidades de color, con los valores $[255, 255, 255]$ y $[0, 0, 0]$ respectivamente.

Bloque state

El bloque *state*, permite asignar un estado emocional por defecto al robot.

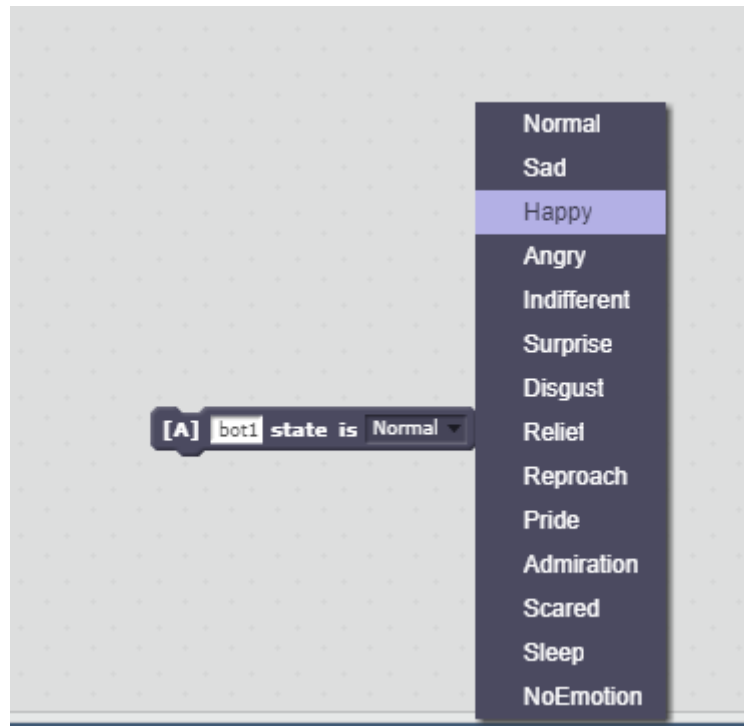


Figura 53. Bloque State.

La asignación de dicho estado se realiza mediante un menú desplegable con una serie de emociones pre-fijadas.

La escenificación de dicha emoción se realiza mediante un control general de los componentes vistos anteriormente, siendo dichos componentes, las cejas, los párpados, la luz pectoral, la posición de la cabeza y la pantalla labial.

Cada estado tiene definido una posición para cada uno de estos componentes. A continuación, se mencionará la posición aproximada de los mismos.

<u>Emoción</u>	<u>Cabeza H</u>	<u>Cabeza V</u>	<u>Cejas</u>	<u>Párpados</u>	<u>Pantalla</u>	<u>Luz</u>
Feliz	Centrada	Centrada	Centrada	Abiertos	Sonrisa	Amarilla
Triste	Centrada	Centrada	Decaídos	Centrados	Decaída	Azul
Enfadado	Centrada	Centrada	Alzadas	Alzados	Dentada	Roja
Dormido	Centrada	Centrada	Normal	Cerrado	Z z Z	Apagada

Tabla 6. Caracterización de emociones

Bloque plays/sound

Por último, en relación a los bloques emocionales, encontramos el bloque plays/sound, que permite reproducir un sonido de los almacenados en el robot.

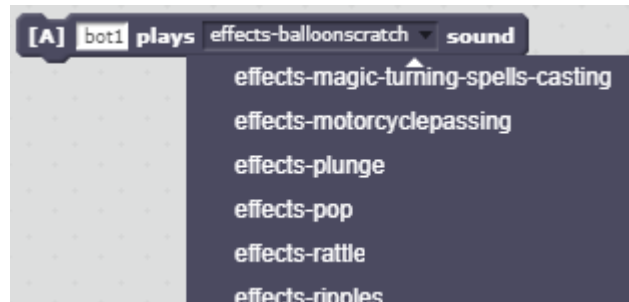


Figura 54. Bloque plays/sound

En el mismo encontramos una lista desplegable en la que seleccionar el sonido a reproducir.

Este bloque no aparece por defecto en la lista de bloques de la extensión experimental, sino que aparece una vez se ha conectado el robot con Scratch, en una sub-sección bot1, como una extensión adicional.

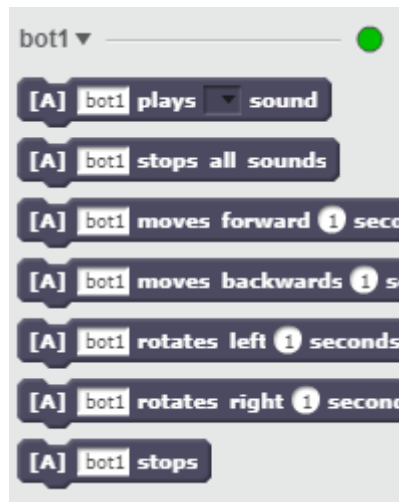


Figura 55. Comandos cargados con el robot.

3.2.2 Demo Práctica 2. Juego con estímulos. (General)

La segunda de las actividades propuestas es una adaptación del juego “Simón dice”.

El mismo se ha adaptado de la siguiente manera:

- Una sola parte por Ronda.
- Los Aciertos proporcionan una estimulación de manera positiva, en nuestro caso, una frase y una rotación.
- Los Fallos son indicados de manera neutral con un accionamiento emocional y la posibilidad de volver a intentarlo.
- Tras un número determinado de rondas, el juego finaliza con una señalización emocional y locutiva.

Esta actividad, ligada más bien a un estadio general del aprendizaje que al autismo, pretende basarse en los modelos de condicionamiento clásico para buscar una respuesta correcta.

Por lo general, un niño es capaz de diferenciar los conceptos de izquierda y/o derecha en cuerpos ajenos de manera completa entorno a los seis u ocho años de edad, aunque la conceptualización del fenómeno se cree que se empieza a desarrollar entorno a los cinco años, en los primeros estadios de aprendizaje de áreas como las matemáticas, la lectura o la escritura. En el caso de niños con TEA, el desarrollo del concepto puede verse retrasado hasta los nueve a doce años de edad, que es la franja en la que por lo general se alcanza la edad mental prevista, siendo en algunos casos relativamente difícil inculcar el concepto antes de esta edad y tras mucha terapia previa.

Junto a la explicación teórica de la Actividad que se desarrollará a continuación, se adjuntará un enlace con la construcción de la actividad y su funcionamiento.

Para la realización del código, se ha seguido una estructura de tres partes/rutinas. Una primera parte de Iniciación y configuración, una segunda de activación por evento, y una última de comprobación y notificación.

Para llevar a cabo el juego, se han fijado una serie de Variables y Listas en el Escenario.



Figura 57. Escenario Actividad 2.

Se pueden observar tres variables: “Parte”, “Iteraciones”, “Fin”, y una lista “Partes”.

En relación a las Variables, la primera, “Parte”, indica el número de la parte actual que va a pedir el robot. La generación de este número, como se verá más adelante, se realiza mediante un selector pseudo-aleatorio con distribución de probabilidad uniforme.

La Variable “Iteraciones”, indica el número de partes total que se han pedido hasta el momento. Aunque el uso aportado ha sido meramente cuantitativo, podría resultar útil en casos en el que se separasen el número de aciertos y fallos como un modo de contabilizarlos.

Por último, la variable “Fin”, señala si el juego se ha terminado o no. Esta última variable, aunque tal y como se ha planteado el código no sería necesaria, resultaría útil en el caso de desear seguir interactuando con el robot de algún otro modo.

Por otro lado, está la lista “Partes”. En la misma se encuentra la denominación de la parte y un número identificador asignado a la misma que coincide con su posición en la lista. Como se ha comentado anteriormente, se ha diseñado de manera que la distribución de partes sea uniforme, no obstante, sería posible enfatizar la probabilidad de elegir una posibilidad en particular, realizando entradas duplicadas de la misma en la lista.

Iniciación:

La rutina de iniciación está constituida por una conexión con el robot y la configuración inicial del entorno del juego.

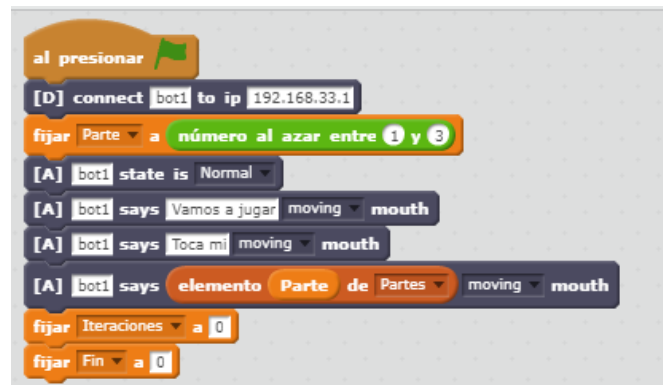


Figura 58. Bloque Inicial.

Para esta misma configuración, se ha realizado una asignación aleatoria de la Variable “Parte” entre las disponibles, y una inicialización del resto de variables a un valor cero.

Además, se ha dotado al robot de unas mínimas líneas de diálogo y un estado anímico neutro, como medio señalizador del inicio de la actividad.

Eventos:

Para realizar la comprobación de la actividad se ha realizado una serie de subrutinas de evento ligadas a la acción requerida, en este caso a la pulsación de la parte en cuestión.



Figura 59. Bloques de Evento.

La construcción de estos eventos, está formada por el accionador, una doble comprobación condicional de las variables Iteraciones y Parte, sirviendo la primera para bloquear la respuesta si el número de interacciones ha superado al deseado y la última para realizar una comprobación de si la respuesta dada es la correcta. En caso de serlo, se enviará un mensaje “Correcto”, siendo en caso contrario “Incorrecto” el mensaje enviado.

Respuestas:

Finalmente, están las subrutinas de “Respuesta”. Estas rutinas están ligadas a los mensajes “Correcto” e “Incorrecto” mostrados anteriormente.

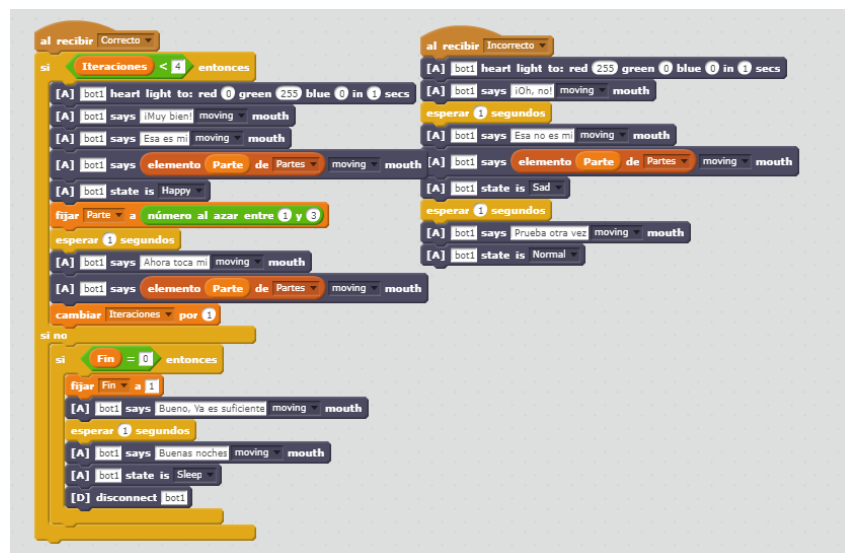


Figura 60. Bloques de Confirmación.

En lo relacionado a la subrutina de “Correcto”, esta se ve formada por una serie de condicionales.

El Primero, vinculado a “Iteraciones” se activará cuando el número de Interacciones con el robot Correctas haya sido menor que el número máximo deseado menos uno, en este caso cuatro. En caso de que se active, el robot denotará el acierto con una señal lumínica, iluminando la luz pectoral de color verde y con una señal acústica y emocional, con un ligero diálogo y gesticulación. Tras el mismo, se elegirá una nueva parte, se avanzará el contador de Interacciones Correctas y se instará al niño a que continúe con esa parte.

En caso de no activarse el condicional de “Iteraciones”, se realizará otra comprobación condicional simple con la variable “Fin”. En caso de activarse esta comprobación, se fijará la variable “Fin” a uno y tras un breve diálogo el robot, se desconectará. Como ya se ha indicado, la existencia y empleo de esta variable viene condicionado a habilitar la posibilidad de seguir interactuando con el robot una vez terminada la actividad en cuestión.

Finalmente, está la subrutina de “Incorrecto”. Su estructuración sigue el patrón encontrado dentro del bloque activo del condicional de “Iteraciones”, con la salvedad de que al no ser correcta la respuesta, se denota con un tono menos jovial. Además, tampoco se actualiza el estado actual de juego, alentando y permitiendo al niño volver a probar las veces necesarias.

Para finalizar el apartado, se procederá a resumir las especificaciones de la actividad, además de adjuntar la URL al contenido relacionado con el mismo.

- Nombre de la actividad: Juego con estímulos: “Simon Says”.
- Rango de edades recomendado: 5+ (general), 9-15 (TEA).
- Objetivo principal: Realzar la concepción del eje corporal.
- Habilidades desarrolladas:
 - o Interacción social.
 - o Cognitiva.
- URL contenido: <https://www.youtube.com/watch?v=Dt2F19E8xOQ>

3.2.3 Demo Práctica 3. Acciones Rutinarias con Tarjetas y Horarios (Adaptable específico a TEA)

Como tercera actividad, se ha realizado una recopilación rutinaria típica de acciones para un caso de TEA. Una de las formas que tienen algunos casos de TEA para comunicarse es mediante unos dibujos denominados pictogramas. Los mismos consisten en la representación gráfica de conceptos como Papa/Mama, Yo, Casa o Gato y/o actividades como Ir al Lavabo, Déjame Solo, Abrazame, etc.

En nuestro caso, con ayuda de códigos QR, se han desarrollado una serie de tarjetas y se ha producido el código necesario para interpretarlas y actuar en cada caso de una manera determinada.

Si bien los *pictogramas* pueden ser una forma efectiva de comunicación para los niños TEA, su empleo está ligado a las etapas intermedias del desarrollo de las habilidades comunicativas del niño, por lo general a partir de los 8 años.

El código podría ser separado en dos partes diferenciables:

- Por un lado, las acciones que están ligadas a un espacio temporal determinado, como puede ser la hora de comer, la hora para la siesta, el tiempo en el colegio, etc. A este código lo llamaremos *Bloque Temporal*.
- Por el otro, aquellas acciones que pueda realizar sin importar el momento del día, como jugar, ir al Lavabo o Acciones Afectivas. Este bloque lo llamaremos *Bloque Atemporal*.

El objetivo es que, por un lado, el niño sea capaz de obtener una respuesta condicionante a la hora de mostrar uno de los pictogramas al robot; siendo en el caso de las temporales negativa si no se cumplen las condiciones temporales fijadas, como puede ser dormir fuera de los horarios de siesta o nocturno, o jugar dentro de los mismos.

Bloque Temporal.

Para el bloque temporal, con el fin de simplificar la rutina se han fijado cuatro segmentos temporales. La siguiente imagen recoge estos mismos segmentos y algunas de las acciones que puede y no puede realizar en esas horas.

	No Puede	Puede
10:00 a 13:00	Dormir Bañarse Ir al Parque	Ir al Colegio Leer Jugar
13:00 a 15:00	Bañarse Jugar	Comer Dormir (Siesta) Ir al Parque
15:00 a 19:00	Comer Dormir Ir al Parque	Leer Ir al Colegio Jugar
19:00 a 22:00	Ir al Colegio	Ir al Parque Bañarse Dormir

Figura 61. Tabla ejemplo de Bloque Temporal.

Uno de los problemas que puede plantear este tipo de actividad, es que, ni los bloques propios de *Scratch*, ni los de *Aisoyl* permiten la activación de un evento por tiempo.

Una solución parcial encontrada para esto, es lo que podríamos denominar una estructura de mensaje cíclica. La siguiente Imagen ejemplifica el funcionamiento de esta estructura.

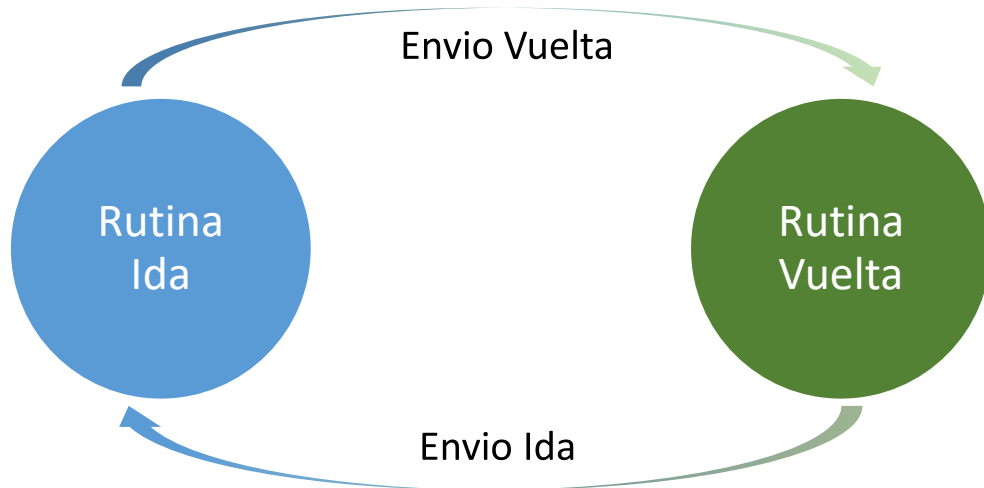


Figura 62. Estructura Cíclica.

En donde en el bloque “*Rutina Ida*” podríamos incluir las pautas de algunos de los cambios de Segmento Temporal (por ejemplo, codificadas como Variables), con una notificación acústica del cambio, y el bloque “*Rutina Vuelta*” otra serie de cambios de los segmentos temporales.

Otra posible solución propuesta sería la inclusión de condicionales (bloque If/Else) ligados al tiempo (bloque Operador “Mayor/Menor/Igual que” con bloque Sensor “minuto/hora actual” para restringir) en cada una de las posibilidades de activación QR.

Si bien, esta solución puede funcionar bien en códigos con Activaciones por Segmentos limitadas, conforme aumenta el número de segmentos temporales y actividades, su complejidad crece, pudiendo generarse código mayor del necesario.

Por último, cabría contemplar una solución basada en una rutina auto-repetitiva, no obstante, las pruebas realizadas con esta estructura en particular no han sido fructíferas, provocando la misma determinados fallos derivados del exceso de ejecuciones del mismo, puesto que Scratch puede ejecutar más de 64.000 ejecuciones del código por segundo en sistemas relativamente modernos.

The code block for 'Bloque Vuelta' starts with an 'al recibir' (when received) trigger set to 'Vuelta'. It contains a 'si' (if) block with a condition: 'hora actual = 08 y minuto actual = 00 y segundo actual = 00'. If true, it executes: '[D] connect bot1 to ip aisoy1.local', '[D] bot1 set todos language to Spanish with Female voice', 'fijar Horario a 0', and 'esperar 1 segundos'. A second 'si' block has a condition: 'hora actual = 13 y minuto actual = 00 y segundo actual = 00'. If true, it executes: 'cambiar Horario por 1' and 'esperar 1 segundos'. The block ends with 'enviar Ida'.

Figura 63. Bloque Vuelta

Se ha optado por la opción de Estructura Cíclica puesto que, aunque puede ser ligeramente más compleja la codificación del cambio de las pautas, nos permite implementar mensajes como pueden ser recordatorios en determinadas horas. Además, nos permite realizar una estructuración visualmente menos abrupta del código.

The code block for 'Bloque Ida' starts with an 'al recibir' (when received) trigger set to 'Ida'. It contains three 'si' (if) blocks. The first has a condition: 'hora actual = 15 y minuto actual = 00 y segundo actual = 00'. If true, it executes: 'cambiar Horario por 1' and 'esperar 1 segundos'. The second has a condition: 'hora actual = 19 y minuto actual = 00 y segundo actual = 00'. If true, it executes: 'cambiar Horario por 1' and 'esperar 1 segundos'. The third has a condition: 'hora actual = 22 y minuto actual = 00 y segundo actual = 00'. If true, it executes: '[D] disconnect bot1'. The block ends with 'enviar Vuelta'.

Figura 64. Bloque Ida.

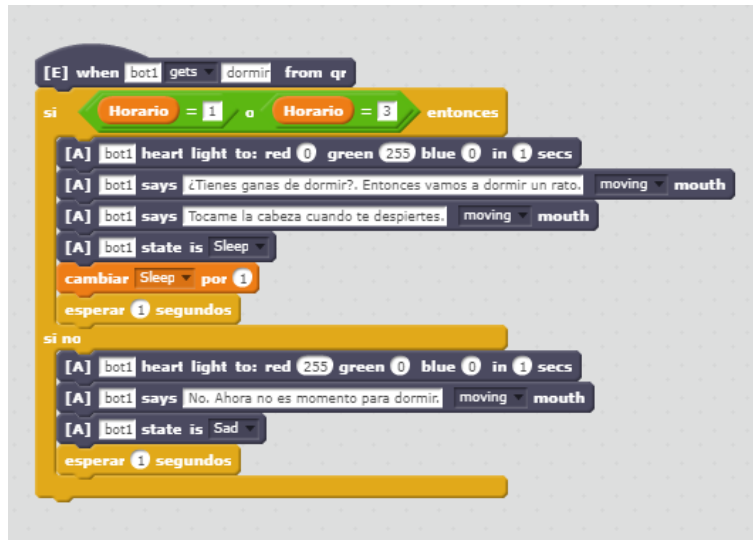


Figura 65. Ejemplo Bloque Temporal

Por otro lado, dentro del bloque temporal hay que contemplar los eventos de activación para las acciones dentro del mismo. Para su estructuración se ha optado por implementar una variable “Horario” que cambia de acuerdo a las franjas horarias delimitadas anteriormente. Junto al evento de detección QR asociado a cada acción, se ha incluido una comprobación condicional de dicha variable, alterando la respuesta del robot en caso de no estar dentro de la franja horaria requerida para la misma.

Bloque Atemporal.

En el bloque Atemporal, se encuentran aquellas acciones o actividades que el niño pueda necesitar realizar con indiferencia del momento del día.

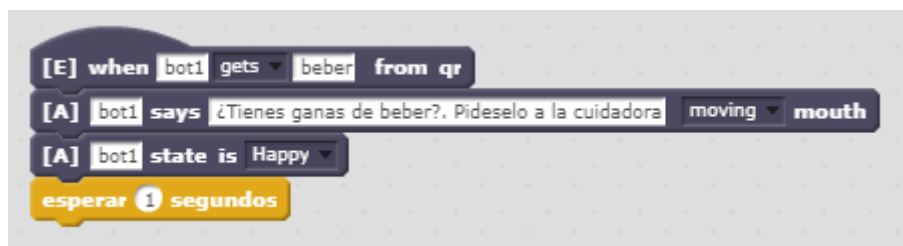


Figura 66. Ejemplo Bloque Atemporal.

Al no requerir de limitación temporal, su código se formará mediante el activador del evento QR correspondiente, y el accionamiento que queramos realizar. Se ha incluido acciones como: Ir al Lavado, Abrazar o Beber.

A continuación, se resume el contexto de la actividad:

- Nombre de la actividad: Acciones Rutinarias con Tarjetas y Horarios.
- Rango de edades recomendado: 5+.
- Objetivo principal: Desarrollar un grado de complicitad con el robot.
- Habilidades desarrolladas:
 - o Interacción social.
 - o Cognitiva.
 - o Comunicativa.
- URL contenido: <https://www.youtube.com/watch?v=Ra5Kc1cGCZU>

3.2.4 Demo Práctica 4. Expresiones y Estados de Ánimo Básicos. (Multi-nivel Específico a TEA)

En esta actividad, se pretende desarrollar un sistema mediante el cual, el niño con TEA pueda adquirir y/o entrenar una serie de facultades en relación a los estados de ánimo. Para ello, no solo nos apoyaremos en Códigos QR, sino que además incluiremos las capacidades ASR del robot.

Por lo general, la conceptualización de las emociones básicas propias en los niños TEA no suele verse afectada, y se desarrollan alrededor de los 0 a 6 años de edad. En este punto, el niño es capaz de reconocer su propio estado de ánimo. Entre los 6 y 12 años, se puede empezar a trabajar con la percepción de estados anímicos en otras personas. Si bien parte de estas emociones en personas ajenas pueden ser aprendidas de manera innata, en algunos casos el desconocimiento de las mismas puede acarrear consecuencias negativas para el propio estado anímico del niño, produciendo angustia, estrés o rechazo.

Un niño con TEA es capaz de entender tres tipos de Estados Anímicos: Felicidad, Tristeza y Enfado. Fuera de estas tres Emociones, es posible en determinados casos que, con la atención pertinente, sean capaces de aprender a expresar otra serie de emociones, como puede ser Sorpresividad, Admiración y Miedo.

Por tanto, se podría plantear dos sub-objetivos para esta actividad.

- 1/ Construir una estructuración para el funcionamiento planteado.
- 2/ Desarrollar una demostración base para esa estructura.

Para esta actividad, nuestro escenario estará compuesto de los siguientes elementos:



Figura 67. Escenario Actividad 4.

Se pueden observar 3 listas, *Emociones*, *Base* y *Diálogos*.

La primera, *Emociones*, servirá para indicar las emociones que queremos que incluya nuestra actividad. Servirá además para indicarle la palabra codificada en los códigos QR y como ayuda en los diálogos del robot.

La segunda, llamada *Base*, sirve para indicar el nombre del niño objetivo de la actividad, es decir empleándola como pseudo-variable.

La Tercera, *Diálogos*, sirve para agrupar todas las frases que queramos que diga nuestro robot, otorgando un cierto grado de claridad a nuestro código.

Además, encontramos una Variable, *Current Emotion*, que indicará la Emoción que debe interpretar actualmente el robot.

Con Respecto a la estructuración del código, se han diseñado una serie de rutinas/sub-rutinas siguiendo el siguiente orden.

Inicio – Enseñanza – Ejercicio – Comprobación/Estimulación.

A continuación, se describirá el funcionamiento de estos bloques.

Inicio:

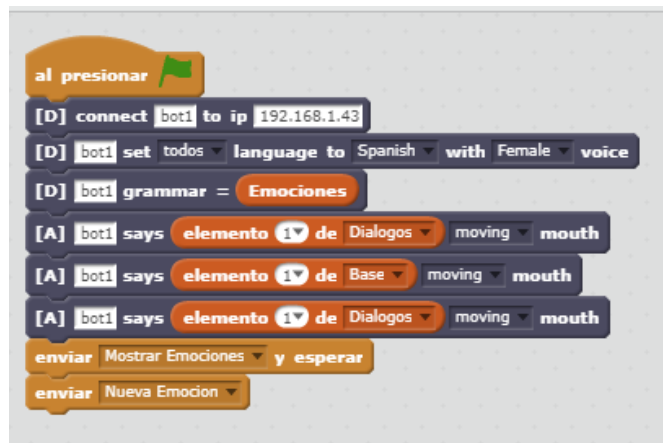


Figura 68. Bloque Inicio Actividad 4.

Esta rutina sirve para la Configuración Inicial del robot. Además, Inicia la conversación con el sujeto. Tras una breve presentación, pasará a la sub-rutina de enseñanza, enviando el mensaje global “*Mostrar Emociones*”.

La configuración inicial, compuesta principalmente por los bloques de configuración Connect, set y Grammar permiten la conexión y adecuación de variables internas para el funcionamiento de la actividad. Además, se ha desactivado las funcionalidades ASR, puesto que, en caso de estar siempre activas, existe la posibilidad que el robot reconozca sus propias palabras como secuencias activadoras.

Enseñanza:



Figura 69. Bloque Enseñanza Actividad 4.

En la subrutina de “Mostrar Emociones”, el robot va enseñando a Intervalos regulares las emociones de las que se le ha dotado y queremos que el niño practique, dejando un pequeño intervalo para que lo asimile. Tras acabar esta subrutina y volver al bloque Inicial, enviará un nuevo mensaje global, “Nueva Emocion” para Iniciar el Bloque de Ejercicio.

Ejercicio:



Figura 70. Bloque Ejercicio Actividad 4.

En esta subrutina, El robot seleccionará de manera aleatoria una de las emociones y la mostrará. Tras ello, mediante los Eventos programados, cuando se activen realizará una comprobación del resultado.

Comprobación/Estimulación.

Finalmente encontramos las subrutinas de Evento y Comprobación.

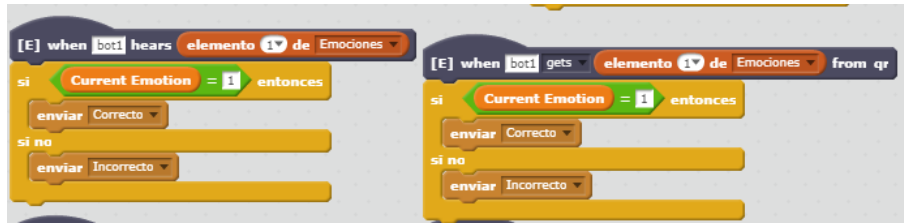


Figura 71. Ejemplo de Subrutinas de Evento.

En lo relativo a las Subrutinas de Evento, constan de un *accionador* ligado a la palabra y/o al código QR de los *pictogramas*. Tras recibir uno de estos accionadores, realizará una comprobación condicional múltiple que en caso de coincidir con la “*Current Emotion*”, enviará un mensaje de “*Correcto*”. En Caso contrario será “*Incorrecto*” lo enviado. Se ha incluido una subrutina de Evento por cada *Emoción* y cada *Accionador* empleado, siendo seis los bloques encontrados de este estilo en esta actividad.

Por otro lado, están las Subrutinas de Comprobación. Estas están ligadas a los mensajes de “*Correcto*” e “*Incorrecto*” citados anteriormente.



Figura 72. Subrutinas de Comprobación.

Los mismos constan del *Accionador* vinculado al mensaje correspondiente, y un pequeño diálogo felicitando al niño por su respuesta, o corrigiéndole en caso de fallo. Se ha incluido además otros estados no practicados a modo de condicionamiento, aunque podrían ser sustituidos por una *Señalización de la Luz Pectoral*.

Para finalizar, como en las otras actividades, se resumirá el contexto de la actividad en cuestión.

- Nombre de la actividad: Acciones Rutinarias con Tarjetas y Horarios.
- Rango de edades recomendado: 6-12.
- Objetivo principal: Realzar la conceptualización de las emociones.
- Habilidades desarrolladas:
 - o Interacción social.
 - o Imaginación - Simbólica.
 - o Comunicativa.

URL contenido: https://www.youtube.com/watch?v=5a2_DddI7P4

4 Conclusiones

A lo largo de la presente memoria referida al TFG realizado, se ha dado una base para la comprensión y realización de dos conceptos en particular.

Por un lado, se ha dado una breve explicación de lo que son los Trastornos de Espectro Autista, sus características y problemáticas, tanto para el afectado como para el tratamiento del referido trastorno. Además, se ha realizado un repaso a la historia de la Robótica y/o Automatización y su posible vinculación ya no solo a un marco educativo, sino terapéutico.

Por el otro, se ha diseñado un compendio, mediante el cual, un especialista en tratamiento de TEA, y con una cierta base con respecto a la informática, sea capaz de realizar Actividades focalizadas en el tratamiento de niños con la referida condición, de manera simple y eficaz.

Si bien, el desarrollo en esta ocasión ha sido mediante el Robot Aisoy1, cabe destacar que sería posible adaptar, en mayor o menor medida, lo contenido en esta memoria a la mayoría de robots disponibles actualmente en el mercado, pues las bases de control de los mismos son similares.

En lo relativo a resultados del trabajo realizado, el alumno opina que sería posible y positiva la incorporación de robots dentro de las rutinas generales de terapia para niños TEA, pudiendo llegar a producir resultados de una manera más presta y eficiente que con la metodología actual.

No obstante, actualmente el alumno observa dos inconvenientes fundamentales para la implementación objetiva de la idea.

En primer lugar, existe una barrera económica severa para la implementación. La mayoría de robots, con un grado de funcionalidad aceptable, rondan los 400€ en los modelos más comunes, siendo además relativamente poco robustos en cuanto a su resistencia, lo cual no hay que olvidar al estar tratando con niños. En este sentido, el Aisoy1 ha demostrado estar bastante bien preparado, siendo relativamente bastante completo en cuanto a funcionalidad además de robusto, con excepción de algunas partes muy específicas.

El otro punto que puede dificultar la implementación, es la poca información a nivel base que existe con respecto a estos robots. Si bien existen multitud de estudios relacionados con los robots y el Autismo, no existe tanta información con respecto a cómo emplearlos. Este ha sido una de las carencias que pretende subsanar el presente TFG, sirviendo como una fuente para la iniciación de la materia en sí.

Aún con estos inconvenientes, se reitera que podría resultar posible y positiva la incorporación de los susodichos robots para intervenciones terapéuticas, siendo también positivas las reacciones obtenidas con las demostraciones realizadas al personal experto consultado para la materia.

5 Proyectos Futuros

Para finalizar esta memoria, comentar que las posibilidades con respecto al robot son aún mayores que las simplemente desarrolladas o citadas a lo largo de este documento, que debido a la naturaleza y tiempo disponible para el TFG en cuestión han sido descartadas para su desarrollo en el mismo.

En lo relativo al *Aisoyl* en particular, existen una serie de nuevos objetivos que sería interesante desarrollar, como podría ser:

- *Una Interconexión más profunda con el robot mediante la Raspberry Pi.*
Este punto vendría referido a un mayor entendimiento del contexto interno del robot, obteniendo información detallada de los sensores y estado general del robot. Para lograrlo se podría recurrir a software como MATLAB, que ofrece de por sí un paquete de interconexión con dispositivos Raspberry, o mediante herramientas creadas específicamente para ello.
- *Análisis de la efectividad de funcionamiento alcanzable mediante el resto de lenguajes disponibles y su viabilidad en el aprendizaje de los mismos.*
Como ya se ha especificado en la memoria, existen una serie de posibilidades de funcionalidad no alcanzables mediante Scratch. El objetivo de este punto vendría a centrarse en las posibilidades que ofrecería la programación del robot con el resto de lenguajes denotados, así como el valorar su grado de efectividad con respecto al lenguaje.

Por otro lado, en un nivel más general, existirían otra serie de objetivos que sería atractivo valorar:

- *Estudio de diversos robots y el grado de aceptación de los mismos con los niños.*
Aunque existen ya estudios similares, la oferta de los robots y su “anatomía” han sufrido una variación en la última década. El objetivo del estudio será analizar si existe alguna relación específica a la complejidad del robot en función a la patología del niño.
- *Estudio de Viabilidad de Implementación de la técnica con carácter general.*
En la presente memoria se ha trabajado con un robot y un lenguaje en particular. Tal y como se ha concluido, existen indicios que permiten afirmar que sería positivo el uso de los robots como herramienta terapéutica para niños TEA. El fin de este estudio sería analizar en profundidad la viabilidad de realizar dicha implementación, ya no solo desde el punto de vista de la técnica en sí, sino desde un contexto más general.
- *Estudio de la Posibilidad de Implementación en otros Ámbitos de Uso.*
Aunque el tratamiento terapéutico ha sido el motivo del trabajo realizado, existen otra serie de Ámbitos en los que podría resultar interesante la utilización de estos robots como una herramienta para tratar con niños TEA en particular, pero con niños a nivel general. Ámbitos como puede ser el legal, en los que haya que tratar con un niño con una afectación en la interacción, como los casos TEA, en supuestos en que haya que explorar o tratar con los mismos en sede judicial, supuestos hoy en día tan frecuentes como la separación o divorcio de sus progenitores, a la hora de la atribución del régimen de guarda y custodia del menor afectado o incluso tan relevantes, de exploración de niños con TEA víctimas de algún delito, dotando a los profesionales de medios con los que poder salvaguardar plenamente los derechos e intereses de estos niños, lo cual podría conseguirse mediante la implementación de tales tecnologías y la debida formación para su uso por parte de los profesionales.

6 Webgrafía

[1] José Luis Molina Marticorena, "Historia de la Robótica", Visitado el 01 de marzo de 2017, de:

<http://www.profesormolina.com.ar/tecnologia/robotica/historia.htm>

[2] Mariela Chavarría, Antonio Saldaño, "LA ROBÓTICA EDUCATIVA COMO UNA INNOVATIVA INTERFAZ EDUCATIVA ENTRE EL ALUMNO Y UNA SITUACIÓN-PROBLEMA" (2010), Visitado el 25 de abril de 2017, de:

<http://runachayecuador.com/refcale/index.php/didascalia/article/download/231/189>

[3] Mariela Chavarría, Antonio Saldaño, "Usos y beneficios de la robótica en las aulas" (2016), Visitado el 26 de abril de 2017, de:

<http://www.upc.edu/latevaupc/usos-y-beneficios-robotica-las-aulas/>

[4] The LEGO Group, "Lego Education WeDO. Guía del Profesor", Visitado el 12 de agosto de 2017, de:

<https://es.slideshare.net/proyectoste/gua-del-profesor-oficial-de-lego-wedo>

[5] Manuel Casanova, "LA DEFINICION DEL AUTISMO" (2013), Visitado el 21 de abril de 2017, de:

<https://corticalchauvinism.com/2013/05/02/la-definicion-del-autismo/>

[6] ADRIANA TAPUS, MAJA J. MATARIC', AND BRIAN SCASSELLATI, "Socially Assistive Robotics" (2007), Visitado el 13 de marzo de 2017, de :

<https://pdfs.semanticscholar.org/ae91/a82975dfcc381b12ddbcc02686b5104bc0dd.pdf>

[7] Tina R. Goldsmith, Linda A. LeBlanc, "Use of Technology in Interventions for Children with Autism" (2004), Visitado el 25 de Abril de 2017, de:

<http://psycnet.apa.org/fulltext/2014-52005-004.html>

[8] Juan Carlos Cruz Ardila, Yeliza Andrea Salazar, "APLICACIÓN ROBÓTICA PARA REALIZAR TERAPIAS EN NIÑOS CON AUTISMO" (2014), Visitado el 27 de abril de 2017, de:

<http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP026.pdf>

[9] John-John Cabibihan, Hifza Javed, Marcelo Ang Jr and Sharifah Mariam Aljunied, "Why Robots? A Survey on the Roles and Benefits of Social Robots for the Therapy of Children with Autism" International Journal of Social Robotics, 2013, 5(4), 593-618, doi 10.1007/s12369-013-0202-2, Visitado el 04 de marzo de 2017, de :

<https://arxiv.org/ftp/arxiv/papers/1311/1311.0352.pdf>

[10] Paola Pennisi, Alessandro Tonacci, Gennaro Tartarisco, Lucia Billeci, Liliana Ruta, Sebastiano Gangemi, and Giovanni Pioggia. "Autism and Social Robotics: A Systematic Review" (2016), Visitado el 13 de marzo de 2017, de:

<https://pdfs.semanticscholar.org/d815/d361be669f9e6d27f647ce754bb590b0731e.pdf>

[11] Laurie Dickstein-Fischer, Elizabeth Alexander, Xiaoan Yan, Hao Su, Kevin Harrington, and Gregory S. Fischer, “An Affordable Compact Humanoid Robot for Autism Spectrum Disorder Interventions in Children” (2011), Visitado el 21 de abril de 2017, de:

<https://pdfs.semanticscholar.org/f093/276d616127a75162656332ade5d0989d4014.pdf>

[12] Paola Pennisi, Alessandro Tonacci, Gennaro Tartarisco, Lucia Billeci, Liliana Ruta, Sebastiano Gangemi, and Giovanni Pioggia. “Autism and Social Robotics: A Systematic Review” (2016), Visitado el 13 de marzo de 2017, de:

<https://pdfs.semanticscholar.org/d815/d361be669f9e6d27f647ce754bb590b0731e.pdf>

[13] Aisoy Robotics, “Aisoy1 KiK - Especificaciones técnicas”, Visitado el 04 de agosto de 2017, de:

<https://www.aisoy.es/aisoy1-spec.html>

[14] Aisoy Robotics, “Aisoy1 KiK - Manual de Usuario”, Visitado el 04 de agosto de 2017, de:

https://aisoy.zendesk.com/hc/en-us/article_attachments/115001006373/A1KiK-Manual_de_usuario.pdf

[15] Scratch Wiki, “Scratch”, Visitado el 07 de agosto de 2017 de:

<https://wiki.scratch.mit.edu/wiki/Scratch>