



**Universidad  
Zaragoza**

## Trabajo Fin de Grado

Diseño de un dispositivo para la integración de  
juguetes convencionales en espacios aumentados  
digitalmente

Design of a device for integrating conventional toys  
in an enhanced digital space

Autor

**Belén Cebrián Martínez**

Director

Dr. Javier Marco Rubio

Ponente

Dra. Sandra Baldassarri

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2017







## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. Belén Cebrián Martínez

con nº de DNI 73011313K en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo

de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la

Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)

Grado \_\_\_\_\_, (Título del Trabajo)

Diseño de un dispositivo para la integración de juguetes convencionales en  
espacios aumentados digitalmente.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada  
debidamente.

Zaragoza, 31 de Agosto de 2017

Fdo: Belén Cebrián Martínez



# Agradecimientos

En primer lugar, quiero dar las gracias a Javier Marco y Sandra Baldassarri, por dirigir, coordinar y realizar conmigo este proyecto. Sin su ayuda e implicación este proyecto no hubiera sido posible.

A mis compañeros de clase, con los que he compartido horas de estudio, risas...Por ayudarme a superar los malos momentos y celebrar los buenos.

A mi familia, especialmente a mis padres, mi hermano y mi tía Geli, por su apoyo incondicional, motivación y cariño durante todo este tiempo.

A todos vosotros, muchas gracias.



# **Diseño de un dispositivo para la integración de juguetes convencionales en espacios aumentados digitalmente**

## **RESUMEN**

El objetivo de este trabajo es el desarrollo de un dispositivo electrónico que permita integrar juguetes electrónicos tangibles en el espacio interactivo aumentado digitalmente que gestiona el GIGA Affective LAB. Estos juguetes deben ser capaces de detectar las manipulaciones de los usuarios y realizar las órdenes recibidas del resto de elementos del espacio.

Inicialmente, se ha realizado un estudio de las tecnologías que están actualmente en uso para integrar objetos en aplicaciones basadas en Interacción Tangible. La Interacción Tangible es un campo emergente de la Interacción Hombre-Máquina, en el que el usuario percibe y controla el contenido digital de una aplicación informática a través de manipulaciones de su entorno físico.

Tras la elección de la tecnología, se han diseñado los bloques electrónicos y el software encargado de la comunicación entre el sistema de la Plataforma JUGUEMOS, implementada en el espacio interactivo, y el dispositivo diseñado en este trabajo, ‘ToyBox’.

Se ha implementado un prototipo con el fin de analizar el funcionamiento de ‘ToyBox’ dentro de la Plataforma JUGUEMOS. Para ello, se han diseñado una placa de circuito impreso, a la que se han conectado los componentes electrónicos del dispositivo, y un encapsulado, que contiene la placa y su alimentación.

Por último, se han realizado varios casos de uso que demuestran el correcto funcionamiento del dispositivo ‘ToyBox’. A partir de los resultados obtenidos, se obtienen unas conclusiones, y unas líneas de mejora que pueden utilizarse en trabajos futuros.



# Design of a device for integrating conventional toys in an enhanced digital space.

## ABSTRACT

The aim of this work is the development of an electronic device which allows the integration of tangible electronic toys in the digitally augmented interactive space managed by GIGA Affective Lab. These toys have to be able to detect manipulations of the users and to carry out the orders received from the rest of elements of the space.

A first step has been the research of the technologies suitable to integrate objects in applications based on Tangible Interaction. Tangible Interaction is an emerging field in Human-Machine interaction, in which the user perceives and controls the digital content of a computer application through manipulations of their physical environment.

Following the choice of the adequate technology, it's been designed the electronic blocks and the communication software between the JUGUEMOS Platform system, implemented in the interactive space, and the device designed in this work, named 'ToyBox'.

A prototype has been implemented in order to evaluate the 'ToyBox' functionality. To this end, it's been designed a printed circuit board, to which the electronic components of the device have been connected, and an encapsulation, which contains the board and its power.

Finally, it's been made several use cases that demonstrate the correct operation of the 'ToyBox' device. Based on the results, conclusions are obtained, and some improvement opportunities which can be used in future works.





# Índice general

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Motivación y contexto.....	1
1.2	Descripción del espacio interactivo .....	2
1.3	Objetivos y requisitos del proyecto .....	5
1.4	Estructura de la memoria .....	6
<b>2</b>	<b>Análisis</b>	<b>9</b>
2.1	Estado del arte: Interacción Tangible .....	9
2.2	Estudio comparativo de las tecnologías.....	13
<b>3</b>	<b>Diseño del Hardware</b>	<b>15</b>
3.1	Diagrama de bloques de ‘ToyBox’.....	15
3.2	Componentes de ‘ToyBox’.....	17
3.2.1	Módulo de comunicación WiFi: ESP8266.....	17
3.2.2	Optoacopladores .....	17
3.2.3	Alimentación de ‘ToyBox’ .....	18
3.3	Placa de circuito impreso .....	19
<b>4</b>	<b>Diseño del Software</b>	<b>21</b>
4.1	Integración software del dispositivo ‘ToyBox’ en la Plataforma JUGUEMOS.....	21
4.2	Programación del módulo ESP8266 .....	22
<b>5</b>	<b>Prototipado</b>	<b>25</b>
5.1	Prototipado electrónico.....	25
5.2	Prototipado del encapsulado .....	28

<b>6 Resultados obtenidos</b>	<b>31</b>
6.1 Dispositivo ‘ToyBox’ .....	31
6.2 Conexión de un objeto electrónico a ‘ToyBox’ .....	32
6.3 Demostración técnica del funcionamiento de ‘ToyBox’ .....	32
6.3.1 Integración de objetos electrónicos al espacio interactivo.....	33
6.3.2 Integración de objetos no electrónicos en el espacio interactivo .....	36
<b>7 Conclusiones y trabajo futuro</b>	<b>39</b>
7.1 Conclusiones.....	39
7.2 Trabajo futuro.....	39
<b>Bibliografía</b>	<b>41</b>
<b>Anexo A Esquemático y PCB</b>	<b>44</b>
<b>Anexo B Presupuesto</b>	<b>47</b>
<b>Anexo C Cronograma de planificación temporal</b>	<b>48</b>
<b>Anexo D Código</b>	<b>49</b>

# Índice de figuras

1.1	Logo GIGA Affective Lab . . . . .	1
1.2	Ejemplo de fiduciales añadidos a juguetes . . . . .	2
1.3	Espacio JUGUEMOS situado en Etopia . . . . .	2
1.4	Arquitectura de la Plataforma JUGUEMOS . . . . .	3
1.5	Sistema de las mesas NIKVision . . . . .	4
1.6	Plataforma electrónica para integrar objetos activos en el espacio interactivo . . . . .	4
1.7	Objetos activos conectados a la plataforma electrónica . . . . .	5
2.1	Phidget InterfaceKit 8/8/8 . . . . .	9
2.2	Arquitectura iStuff . . . . .	10
2.3	Módulo de la plataforma Atlas . . . . .	10
2.4	Sensor Smart-Its . . . . .	11
2.5	Placa Arduino Uno . . . . .	12
2.6	Raspberry Pi modelo B+ . . . . .	12
2.7	Módulo ESP8266 (modelo ESP-01) . . . . .	12
3.1	Nueva arquitectura de la Plataforma JUGUEMOS . . . . .	15
3.2	Integración de ‘ToyBox’ en la Plataforma JUGUEMOS . . . . .	16
3.3	Diagrama de bloques de ‘ToyBox’ . . . . .	16
3.4	Módulo ESP8266 (Modelo ESP-01) . . . . .	17
3.5	Optoacoplador 4N35 . . . . .	18
3.6	Bloque de alimentación del dispositivo ‘ToyBox’ . . . . .	18
4.1	Mensaje OSC de conexión . . . . .	21
4.2	Formato de mensaje OSC que envía el dispositivo ‘ToyBox’ . . . . .	22

4.3	Formato de mensaje OSC que envía el broadcaster a ‘ToyBox’ . . . . .	22
4.4	Diagrama de flujo de la programación del ESP8266 . . . . .	22
5.1	Prototipo experimental del circuito electrónico de ‘ToyBox’ . . . . .	25
5.2	Esquemático del circuito electrónico del dispositivo . . . . .	26
5.3	Diseño de la PCB del dispositivo . . . . .	26
5.4	Prototipo de la PCB fabricada . . . . .	27
5.5	Prototipo de la placa electrónica de ‘ToyBox’ . . . . .	27
5.6	Modelo 3D del prototipo electrónico de ‘ToyBox’ . . . . .	28
5.7	Modelo 3D del encapsulado . . . . .	29
5.8	Impresión realizada por la impresora 3D <i>Zortrax M200</i> . . . . .	29
5.9	Prototipo de encapsulado . . . . .	29
6.1	Dispositivo ‘ToyBox’ . . . . .	31
6.2	Conexión de un objeto electrónico al dispositivo ‘ToyBox’ . . . . .	32
6.3	Cerdito de marioneta . . . . .	33
6.4	Pulsador de la marioneta . . . . .	33
6.5	Introducción de la electrónica en la marioneta . . . . .	34
6.6	Demo con el <i>cerdito</i> de marioneta . . . . .	34
6.7	Espada . . . . .	35
6.8	Demo con la espada . . . . .	36
6.9	Regadera . . . . .	36
6.10	Demo con la regadera . . . . .	37

# Índice de tablas

2.1	Comparación entre los diferentes modos de comunicación inalámbrica . . . . .	13
2.2	Comparación entre las especificaciones que tiene cada una de las tecnologías . .	14



## Lista de abreviaturas

IoT: Internet of Things

PCB: Printed Circuit Board

OSC: Open Sound Control

UDP: User Datagram Protocol





# 1 Introducción

En este capítulo se describe el contexto de desarrollo del Trabajo Fin de Grado y su motivación inicial. A continuación se realiza una descripción del espacio interactivo y se presentan los objetivos y requisitos del proyecto. Por último, se presenta la estructura de la memoria, incluyendo capítulos y anexos.

## 1.1. Motivación y contexto

El grupo de investigación *GIGA Affective Lab* [GIG17] (Figura 1.1) de la Universidad de Zaragoza se encuentra involucrado en el proyecto JUGUEMOS: Juegos Pervasivos Basados en Interfaces Multimodales Emocionales y Agentes Sociales, proyecto TIN-2015-67149-C3-1R financiado por el Ministerio de Educación y Ciencia. Además, gracias a la colaboración del Instituto de Biocomputación y Física de Sistemas Complejos (BIFI) de la Universidad de Zaragoza con el Ayuntamiento de Zaragoza para la utilización de espacios y equipamiento con fines de investigación y difusión ciudadana de la investigación científica, el *GIGA Affective Lab* participa en el proyecto CeSAr-Etopía, mediante el cual dispone de un espacio en Etopía (Centro de Arte y Tecnología) ubicado en Zaragoza, y de equipamiento hardware para la implementación del espacio interactivo de juegos pervasivos necesario para el desarrollo del proyecto JUGUEMOS.



Figura 1.1: Logo GIGA Affective Lab

El objetivo del proyecto JUGUEMOS es el desarrollo de una plataforma hardware y software para el prototipado de juegos pervasivos en espacios interactivos. Los juegos pervasivos [MSW09] son aquellos que buscan eliminar la brecha entre el juego tradicional y el videojuego creando un continuo entre la experiencia del juego y la plataforma en la que se desarrolla. Para ello, el *GIGA Affective Lab* se ha basado en la Interacción Tangible [HB06], un campo emergente de la Interacción Hombre-Máquina, en el que el usuario percibe y controla el contenido digital de una aplicación informática a través de manipulaciones de su entorno físico. Esto es posible gracias a la integración de componentes electrónicos en objetos y entorno físico y mediante el uso de dispositivos, materiales y diversas tecnologías.

Los objetos físicos aumentados computacionalmente sirven como representaciones tangibles y directas de la información digital, y funcionan tanto de entrada como de salida, ofreciendo a los usuarios ciclos de retroalimentación paralela: expresadas mediante restricciones físicas en el objeto que informan al usuario que una manipulación se ha completado, y digitales, visual o auditiva, informando a los usuarios de la interpretación digital de dicha manipulación.

La interacción mediante objetos tangibles se lleva a cabo sobre la superficie de mesas interactivas tangibles llamadas tabletop NIKVision. Para ello, es necesario adherir a la base de

los objetos un marcador, fiducial, que actúa a modo de código visual fácilmente identificable mediante algoritmos de análisis de imagen (Figura 1.2). Los fiduciales permiten que el sistema sea capaz de saber qué objeto está sobre la mesa y cuál es su posición y orientación. El usuario controla el juego manipulando los objetos sobre la superficie de la mesa. A este tipo de objetos, cuya función se limita a ser manipulados por el usuario, se les denomina *pasivos*. Existe también otro tipo de objetos tangibles definidos como *activos* que son aquellos capaces de reaccionar a las manipulaciones de los usuarios mediante cambios físicos de sus propiedades, como movimientos o sonidos. Para la implementación de objetos activos en aplicaciones para el tabletop NIKVision, el *GIGA Affective Lab* dispone de una plataforma electrónica desarrollada en un trabajo fin de grado anterior [Vil17].

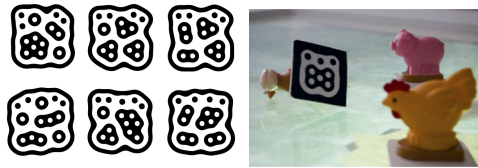


Figura 1.2: Ejemplo de fiduciales añadidos a juguetes

Dado que, en el contexto del proyecto JUGUEMOS anteriormente mencionado, se requiere que la interacción con los objetos físicos vaya más allá de las superficies del tabletop, surge la necesidad de crear y adaptar objetos tangibles activos para que estén integrados en el espacio interactivo de manera independiente, es decir, que el uso de objetos tangibles activos pueda cubrir todo el espacio interactivo *JUGUEMOS*.

## 1.2. Descripción del espacio interactivo

En el contexto de la colaboración en los Laboratorios CeSAr, el *GIGA Affective Lab* gestiona un espacio en el edificio Etopia (Centro de Arte y Tecnología) de Zaragoza. El espacio interactivo (Figura 1.3) que se ha creado está compuesto por cuatro mesas interactivas tangibles NIKVision, pantallas de proyección y diversos sensores capaces de identificar y seguir las manipulaciones y movimientos de los usuarios dentro del espacio. El espacio es un área irregular de unos 70 metros cuadrados.



Figura 1.3: Espacio JUGUEMOS situado en Etopia

La arquitectura diseñada (Plataforma JUGUEMOS) permite gestionar un número indeterminado y heterogéneo de dispositivos electrónicos embebidos e interconectados en un espacio interactivo. Estos dispositivos tienen como finalidad sensorizar y actuar sobre el espacio interactivo. En la figura 1.4 se representan aquellos dispositivos que están integrados actualmente

en la plataforma. Para cada tipo de dispositivo integrado es necesario implementar un gestor software, encargado del acceso al hardware del dispositivo. Se trata de procesos específicamente desarrollados para cada dispositivo, y que se ejecutan en ordenadores o microcontroladores a los que está directamente conectado el dispositivo. Con el fin de integrar el gestor software de cada dispositivo en la Plataforma, cada gestor implementa un proceso publicador, es decir, un proceso de comunicación cliente-servidor basado en el protocolo de red OSC [WF97], protocolo basado en *sockets* UDP. El proceso publicador envía información del estado de los sensores y recibe órdenes para ser ejecutadas en los actuadores. Por tanto, los procesos publicadores se basan en comunicación bidireccional, aunque no es necesario implementar siempre ambas direcciones. La integración de los diversos tipos de dispositivos en el espacio interactivo con la Plataforma se consigue a través de la estandarización del proceso publicador, es decir, todos implementan el mismo protocolo de comunicación, utilizando un formato común de mensaje OSC.

La comunicación entre todos los procesos publicadores se realiza a través de un proceso *broadcaster*, que mantiene conexión OSC individualizada con cada proceso publicador. El proceso *broadcaster* puede gestionar un número ilimitado y dinámico de conexiones con los procesos publicadores. La función del proceso broadcaster es de mero tránsito y redireccionamiento de la información.

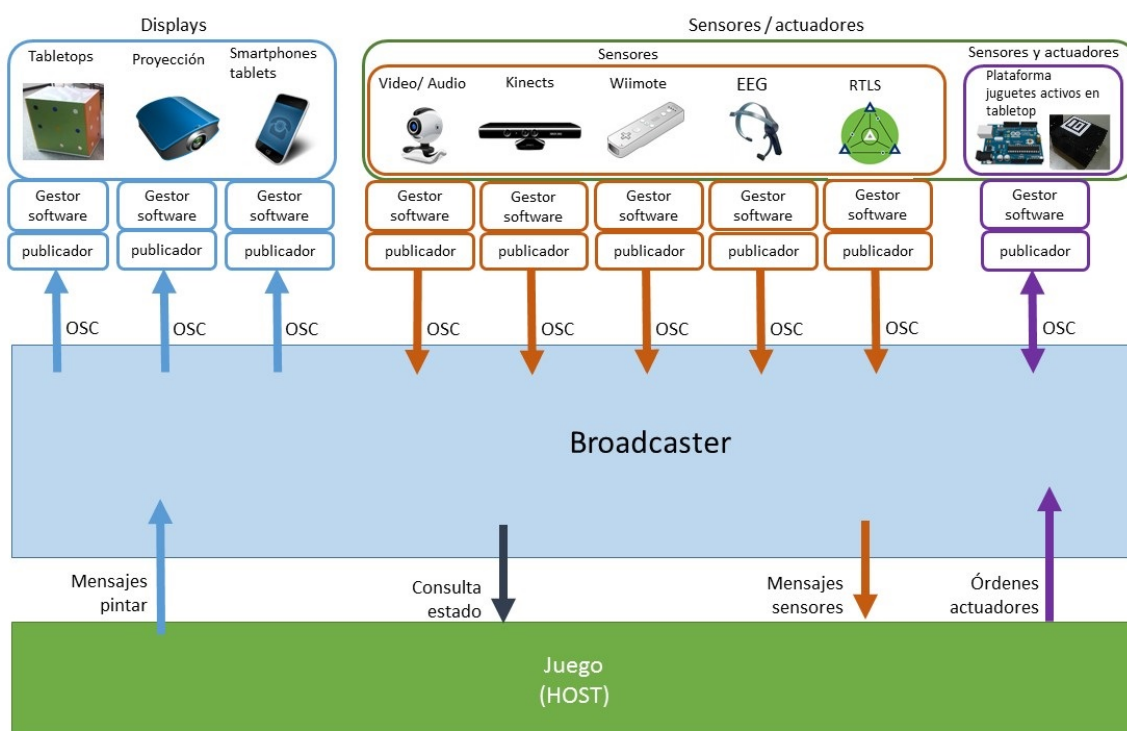


Figura 1.4: Arquitectura de la Plataforma JUGUEMOS

Uno de los elementos centrales del espacio interactivo *JUGUEMOS* son cuatro mesas NIK-Vision [MCB08]. Se trata de *tabletops* tangibles diseñados por el Grupo de Informática Gráfica Avanzada (GIGA) de la Universidad de Zaragoza. Un *tabletop* es una mesa convencional en la que se proyectan imágenes de ordenador sobre la superficie permitiendo que el usuario interactúe con sus manos u otros objetos. NIKVision permite diseñar juegos interactivos basados en Interacción Tangible a través de la manipulación de objetos colocados sobre la superficie de la mesa. Para detectar estas manipulaciones, el *tabletop* dispone de una cámara infrarroja que

capta un conjunto de blancos y negros que envía al sistema para que realice un análisis a través de algoritmos de tratamiento de imagen (Figura 1.5). Esto permite identificar qué objeto hay sobre la superficie de la mesa e identificar las manipulaciones realizadas. Para facilitar esta identificación, se añade un marcador (fiducial) a los objetos que actúa de código visual.

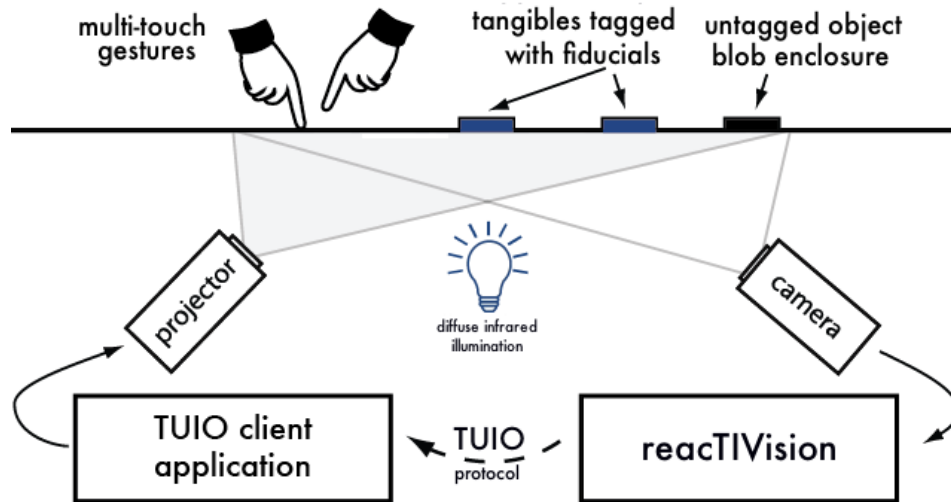


Figura 1.5: Sistema de las mesas NIKVision

Como se ha mencionado en el apartado anterior, la integración de objetos activos manipulados sobre la superficie de la mesa de forma inalámbrica es posible gracias a una plataforma electrónica (Figura 1.6) desarrollada en un Trabajo Fin de Grado anterior. Dicha plataforma permite el conexonado fácil de sensores y actuadores embebidos en piezas de juego manipulables sobre la superficie del tabletop. La transmisión de datos entre el *tabletop* y la plataforma se realiza a través de comunicación por infrarrojos. Para ello, se dispone de un transmisor IR controlado por un *Arduino Uno* que se encuentra conectado al ordenador de la mesa, y un receptor IR situado en el interior de la plataforma electrónica conectado a un *Arduino Nano*. Los objetos que se conectan a la plataforma no disponen de alimentación propia, sino que son alimentados a partir de la batería de la plataforma, por lo que, debido al consumo que la alimentación de la plataforma puede soportar, queda limitado el tipo de sensores y actuadores que pueden conectarse a ella. En la figura 1.7 se pueden ver ejemplos de objetos conectados a la plataforma electrónica.

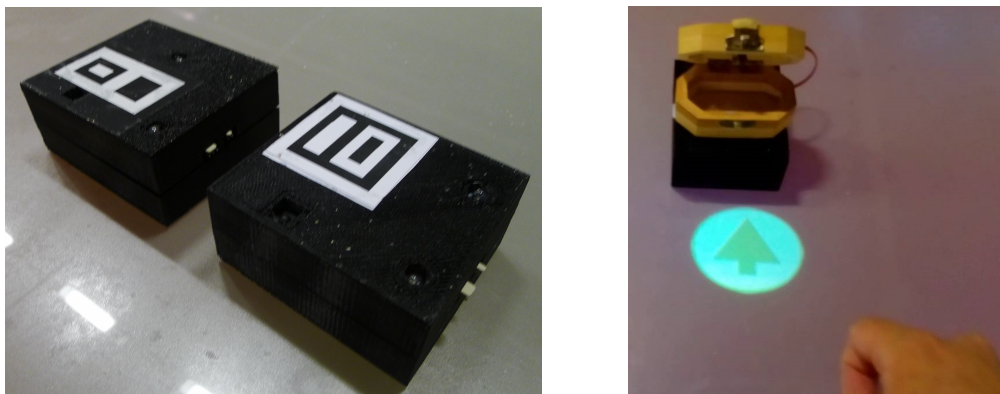


Figura 1.6: Plataforma electrónica para integrar objetos activos en el espacio interactivo

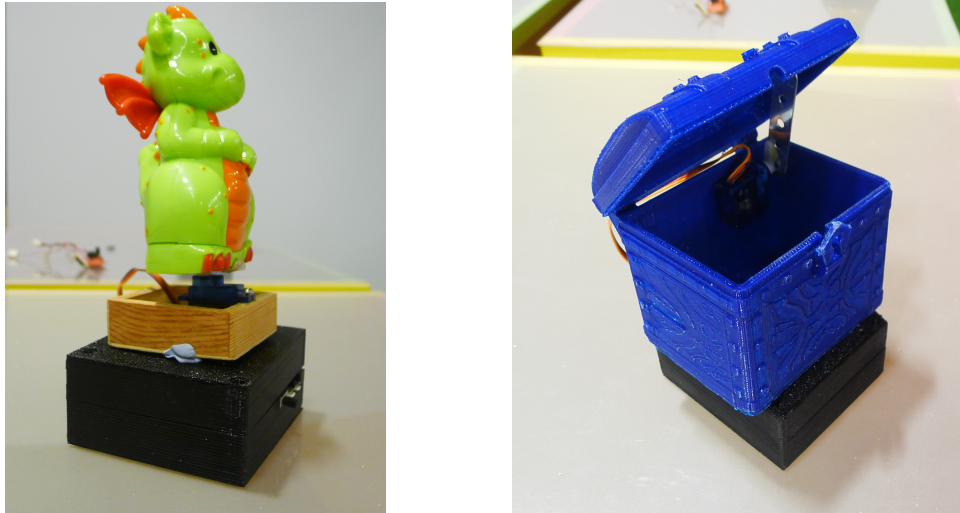


Figura 1.7: Objetos activos conectados a la plataforma electrónica

Esta plataforma electrónica dispone de comunicación por infrarrojos, por lo que no puede ser utilizada para integrar objetos activos en todo el espacio interactivo. Este modo de comunicación tiene un rango de funcionamiento pequeño y es muy sensible a objetos que interfieren o perturban la comunicación. Además el consumo de la alimentación de la plataforma limita el tipo de sensores y actuadores al que se puede conectar, ya que éstos se alimentan a través de ella.

### 1.3. Objetivos y requisitos del proyecto

En base al contexto planteado en el apartado anterior, el objetivo de este proyecto consiste en desarrollar un dispositivo, en adelante se denominará ‘ToyBox’, que permita integrar, de forma sencilla, objetos electrónicos activos en los juegos del espacio interactivo, de manera que el sistema del espacio interactivo sea capaz de conocer el estado de las manipulaciones de los usuarios con cada uno de los objetos integrados en el espacio (sensores) y, por otro lado, el sistema del espacio tiene que tener la posibilidad de enviar órdenes a cualquiera de los objetos integrados para que éstos reaccionen mediante cambios físicos percibibles por los usuarios (actuadores). Para ello, será necesario realizar un estudio y análisis exhaustivo de las tecnologías que actualmente están en uso para integrar objetos en aplicaciones tangibles.

El dispositivo debe establecer una comunicación inalámbrica con los elementos de la Plataforma JUGUEMOS. Además, es necesario que sea posible integrar un número indefinido de dispositivos funcionando simultáneamente.

‘ToyBox’ debe ser lo más universal posible, es decir, prácticamente cualquier tipo de objeto electrónico debe ser integrable en el espacio interactivo a través de ‘ToyBox’, por lo que el dispositivo tendrá una alimentación independiente de la del objeto al que se conecte.

Es necesario que se tenga acceso a la alimentación del dispositivo para poder sustituirla si fuera preciso y al microcontrolador para poder incorporar mejoras o modificaciones en su programación. Además debe poder ser utilizado por cualquier usuario, por lo que la conexión



de los objetos a ‘ToyBox’ debe ser sencilla.

Debe tener también el tamaño adecuado para no dificultar el uso del objeto. Para proteger los componentes electrónicos internos y facilitar el manejo del objeto por el espacio, el dispositivo debe estar protegido con un encapsulado y ser compacto.

Es necesario realizar un conjunto de casos de uso que demuestren el correcto funcionamiento de ‘ToyBox’ con diferentes juguetes convencionales.

Se pretende diseñar un dispositivo que cumpla con los objetivos establecidos. Para que esto sea posible, surge una serie de requisitos funcionales y no funcionales que ‘ToyBox’ debe cumplir.

### **Requisitos funcionales del dispositivo ‘ToyBox’**

- RF1 - Debe ser capaz de establecer una comunicación inalámbrica con el resto de elementos integrados en el espacio interactivo a través del protocolo de comunicación implementado en la Plataforma JUGUEMOS.
- RF2 - El dispositivo se debe poder utilizar en todo el espacio interactivo, de forma que no se produzcan interrupciones en la comunicación.
- RF3 - El objeto que se conecte a ‘ToyBox’ debe poder emplearse tanto como sensor como actuador, por lo que ‘ToyBox’ debe ser capaz de emitir y recibir mensajes.
- RF4 - Debe tener alimentación propia e independiente de la alimentación de los objetos que se le conecten.
- RF5 - Simplificar la conexión con el objeto, de manera que el usuario no necesite conocimientos técnicos en electrónica.

### **Requisitos no funcionales del dispositivo ‘ToyBox’**

- RNF1 - Debe ser pequeño para facilitar el uso del objeto por el espacio.
- RNF2 - Debe tener un encapsulado para proteger su circuito electrónico.
- RNF3 - Debe ser compacto y se debe poder acceder al circuito electrónico embebido en él para poder realizar modificaciones en su programación o sustituir su alimentación.

## **1.4. Estructura de la memoria**

Este documento se divide en dos partes: la primera parte contiene la memoria del proyecto, en la que se resume el trabajo realizado; la segunda contiene los anexos que explican con mayor detalle ciertos aspectos de la memoria.

La memoria contiene los siguientes capítulos:

- Capítulo 2 - Análisis: contiene un Estado del arte sobre la Interacción Tangible y un estudio de las tecnologías disponibles para su desarrollo.
- Capítulo 3 - Diseño del Hardware: contiene una explicación del diagrama de bloques del dispositivo y su integración en el espacio interactivo y una descripción de los componentes más significativos del circuito electrónico.
- Capítulo 4 - Diseño del Software: contiene una explicación sobre cómo se realiza la transmisión de los datos entre el dispositivo y el resto de elementos del sistema y la programación del módulo ESP8266.
- Capítulo 5 - Prototipado: contiene una explicación sobre el prototipado del dispositivo, tanto la parte electrónica del dispositivo como su encapsulado.
- Capítulo 6 - Resultados obtenidos: se muestra el dispositivo final diseñado, se explica cómo se conectan objetos al dispositivo y se muestran tres casos de uso que se han realizado a modo de demostración técnica del correcto funcionamiento del dispositivo.
- Capítulo 7 - Conclusiones y trabajo futuro: se recogen las conclusiones del proyecto y el trabajo futuro a realizar.

Los anexos son los siguientes

- Anexo A: contiene el esquemático del circuito electrónico, junto al diseño de la PCB.
- Anexo B: contiene el presupuesto del dispositivo desarrollado, donde se muestra el coste de cada componente y el coste total del dispositivo.
- Anexo C: contiene el cronograma de planificación temporal.
- Anexo D: contiene el código que ejecuta el módulo *ESP8266*.





## 2 Análisis

En este capítulo realiza un estudio sobre las tecnologías que están actualmente en uso para integrar objetos en aplicaciones basadas en Interacción Tangible. En base a este análisis, se realiza una comparativa para escoger la tecnología más adecuada para el trabajo que permita cumplir con los requisitos establecidos.

### 2.1. Estado del arte: Interacción Tangible

A continuación se estudian algunas de las tecnologías más utilizadas en el campo de la Interacción Tangible, en especial aquellas que permiten integrar objetos físicos, para ver cuáles son adecuadas a este proyecto.

Como ya se ha dicho en el capítulo anterior, la Interacción Tangible es una forma de Interacción Hombre-Máquina que pretende unir el mundo digital con el mundo físico integrando objetos y entorno físico aumentados digitalmente. Para lograr técnicamente este objetivo, es necesario embeber dentro de los objetos físicos del entorno interactivo, dispositivos y componentes electrónicos que doten de capacidades de sensorización y actuación a dichos objetos. En los últimos años se han desarrollado distintos toolkits electrónicos que facilitan dicha tarea, citándose a continuación los más conocidos y usados en la actualidad.

Una de las tecnologías más utilizadas en este campo es *Phidgets* [GF01]. *Phidgets* (widgets físicos) es un conjunto de componentes electrónicos de bajo costo que son controlados por un ordenador. El hardware está diseñado para conectar sensores, como switches o acelerómetro, y actuadores, como LEDs o servomotores. La comunicación entre ordenador y *Phidget* se realiza a través del bus serie universal (USB). En la Figura 2.1 se puede observar un ejemplo de tarjeta Phidget, el cual se alimenta a través del USB y dispone de entradas analógicas, en las que se puede conectar sensores como temperatura o posición, entradas digitales, en las que se pueden conectar pulsadores o relés, y salidas digitales.



Figura 2.1: Phidget InterfaceKit 8/8/8

Otra de las tecnologías más utilizadas para la Interacción Tangible es *iStuff* [BRS03]. *iStuff* incluye un conjunto de herramientas físicas y una infraestructura de software flexible diseñada para simplificar la exploración de nuevas técnicas de interacción de múltiples usuarios, sistemas y aplicaciones que colaboran en un espacio interactivo. La comunicación se realiza a través del *Event Heap*, un proceso central que recibe eventos de los elementos del espacio y los

redistribuye a su destinatario. En la Figura 2.2 se muestra la arquitectura de *iStuff*, en la que se observa que *iStuff* está formado por dispositivos inalámbricos emparejados a una máquina conectada al *Event Heap* y que tiene un transceptor<sup>1</sup> y dispone del software necesario para ser utilizado como proxy en el espacio. El proxy encapsula datos en un evento (o extrae datos de él), haciendo *iStuff* independiente de cualquier protocolo o tecnología inalámbrica. Esta arquitectura requiere que cada elemento que se añade en el espacio interactivo esté conectado a un ordenador, el cual será el encargado del intercambio de mensajes con el resto de elementos del espacio a través del *Event Heap*.

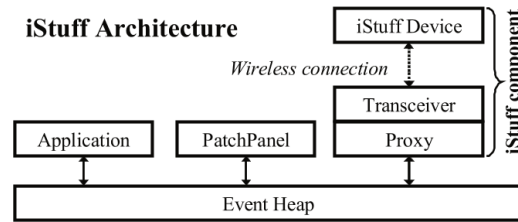


Figura 2.2: Arquitectura iStuff

Una de las diversas tecnologías empleadas en el desarrollo de espacios pervasivos es *Atlas* [KBY06]. *Atlas* es una plataforma de sensores y actuadores que permite crear espacios auto-integrativos y programables. Esta plataforma proporciona nodos físicos para conectar varios dispositivos heterogéneos, un sistema para traducir esos dispositivos en servicios de software, un sistema de mantenimiento de una librería de servicios de dispositivos y sus interfaces, y un entorno para acceder a servicios y diseñar aplicaciones. La figura 2.3 muestra uno de los diversos módulos de la plataforma *Atlas*, el cual dispone de entradas analógicas y comunicación por Ethernet. *Atlas* consiste en una plataforma para el desarrollo de juegos pervasivos en espacios interactivos y, dado que el *GIGA Affective Lab* ya dispone de una plataforma, no es posible su implementación en el espacio interactivo JUGUEMOS.



Figura 2.3: Módulo de la plataforma Atlas

Otra tecnología a mencionar es *Smart-Its* [HGK04]. *Smart-Its* consiste en computadores autónomos que se adhieren a los objetos cotidianos. Estos objetos aumentados permiten relaciones digitales dinámicas con usuarios. Son dispositivos pequeños formados por sensores y actuadores que se añaden a los objetos de manera que permite su integración en el espacio. Esta tecnología se basa en comunicación RF entre dispositivo y ordenador para la transferencia de información. Existen diferentes módulos implementables en estos objetos, como por ejemplo un módulo sensor (Figura 2.4). *Smart-Its* no está diseñado para integrar objetos electrónicos

<sup>1</sup>Un transceptor es un dispositivo que cuenta con un transmisor y un receptor que comparten parte de la circuitería o se encuentran dentro de la misma caja

en entornos inteligentes, ya que su función es incorporar sensores y actuadores a los objetos a los que se añade.

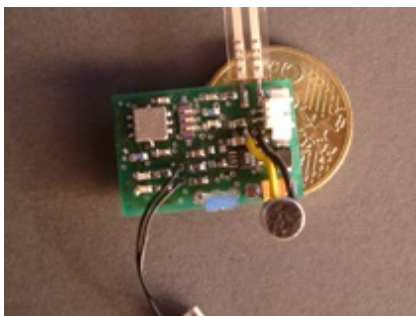


Figura 2.4: Sensor Smart-Its

Las tecnologías previamente presentadas son aquellas que más se están utilizando actualmente en el campo de la Interacción Tangible, pero no cumplen los requisitos de este proyecto. Por un lado, *Phidgets* no dispone de dispositivos con comunicación inalámbrica, principal requisito de este proyecto; *iStuff* y *Atlas* proporcionan plataformas para el desarrollo de aplicaciones tangibles en espacios interactivos, pero el *GIGA Affective Lab* ya dispone de la Plataforma JUEGUEMOS para el prototipado de juegos pervasivos integrada en el espacio interactivo, por lo que no es conveniente integrar dichas plataformas dentro de ésta; y *Smart-Its* está diseñado para integrar sensores y actuadores en objetos cotidianos, y este proyecto pretende integrar objetos electrónicos en el espacio interactivo. Por tanto, ninguna de estas tecnologías existentes en el campo de la Interacción Tangible es adecuada para la realización de este proyecto. Por este motivo, es necesario ampliar el estudio a otras áreas relacionadas en busca de una tecnología que cumpla con los requisitos del proyecto.

Dado que el objetivo de este proyecto es diseñar un dispositivo que permita integrar objetos electrónicos en espacios interactivos, existe cierta relación con el Internet de las cosas (IoT - *Internet of Things*) [GBM13], pues consiste en la interconexión digital de objetos cotidianos a través de Internet. Estos objetos constan de sistemas embebidos que les permiten la conectividad a Internet y realizar las tareas que les han sido asignadas remotamente. El creciente interés que está recibiendo el campo de *Internet de las cosas* ha propiciado la proliferación de diversas tecnologías que permitan (con cierta facilidad) integrar dispositivos electrónicos en un sistema digital que integre distintos dispositivos. Siendo los objetivos del *Internet de las cosas* similares a los objetivos de este proyecto, a continuación analizamos estas tecnologías que recientemente están surgiendo en este contexto:

Una de las tecnologías más utilizadas actualmente en el *Internet de las cosas* es *Arduino* [Ard]. *Arduino* es una plataforma electrónica de código abierto basada en hardware y software. Se trata de una herramienta sencilla para el prototipado rápido. Su hardware está formado por una placa que tiene integrados un microcontrolador, memoria y puertos de entrada y salida en los que se pueden conectar otros dispositivos. Existen diferentes modelos de *Arduino*, en la Figura 2.5 se puede observar el *Arduino Uno*. *Arduino* se ha convertido en una herramienta muy utilizada tanto por estudiantes y aficionados como por programadores y profesionales, llegando a tener gran importancia en diversos entornos, como en impresión en 3D o aplicaciones IoT. Esto se debe a que dispone de múltiples modelos y tamaños y una gran variedad de módulos y shields que complementan a estos modelos, dotándoles de características extras como, por ejemplo, comunicación inalámbrica.



Figura 2.5: Placa Arduino Uno

La siguiente tecnología es *Raspberry Pi* [Ras]. Se trata de un computador de placa simple (SBC) de bajo costo que contiene un procesador, memoria RAM, GPU, Ethernet y puertos de entrada y salida (Figura 2.6). En los últimos modelos se ha incorporado también la posibilidad de comunicación inalámbrica a través de WiFi o bluetooth. *Raspberry Pi* dispone también de un software optimizado para su hardware simplificando su empleo para el desarrollo de aplicaciones. Esta tecnología es muy utilizada en control domótico, debido a su gran capacidad de procesamiento.



Figura 2.6: Raspberry Pi modelo B+

Por último, el módulo *ESP8266* [ESP] es la tecnología más reciente de todas las nombradas anteriormente. Se trata de un chip integrado de bajo costo con conexión WiFi y compatible con el protocolo TCP/IP. El *ESP8266* tiene integrado un microprocesador, una memoria RAM y puertos de entrada y salida. Este módulo está teniendo mucha importancia en la comunidad de programadores y desarrolladores y, gracias a ella, dispone de firmwares y documentación. El campo donde más importancia está teniendo es en el Internet de las cosas (IoT), llegando a ser competencia directa del *Arduino*. Esta tecnología dispone de diferentes modelos, como el ESP-01 mostrado en la Figura 2.7.

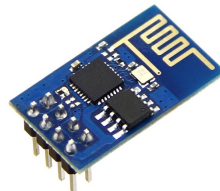


Figura 2.7: Módulo ESP8266 (modelo ESP-01)

Una vez realizado el estudio sobre las diferentes tecnologías que existen actualmente para el desarrollo de juegos pervasivos mediante Interacción Tangible, se realiza un estudio para elegir la tecnología a emplear para el desarrollo del proyecto en función de sus requisitos.

## 2.2. Estudio comparativo de las tecnologías

En este apartado se va a realizar un estudio comparativo sobre las tecnologías expuestas en el apartado anterior y se elegirá aquella que sea más apropiada en función de los requisitos especificados para la realización de este proyecto. Antes de elegir una tecnología es necesario decidir qué tipo de comunicación inalámbrica tendrá el dispositivo con el resto de los elementos del espacio interactivo. Existen cuatro principales posibilidades de comunicación inalámbrica:

- Comunicación por infrarrojos.
- Comunicación por Radiofrecuencia.
- Comunicación por bluetooth.
- Comunicación por WiFi.

	Ventajas	Desventajas
<b>Infrarrojos</b>	<ul style="list-style-type: none"> <li>- Consumo muy reducido.</li> <li>- Componentes de bajo coste</li> <li>- Transmite señales a altas velocidades.</li> </ul>	<ul style="list-style-type: none"> <li>- Funciona en pequeñas distancias.</li> <li>- Muy sensible a objetos que interfieren o perturban la comunicación entre emisor y receptor.</li> <li>- La luz solar o otras fuentes de luz brillante pueden interferir en la comunicación.</li> </ul>
<b>Radiofrecuencia</b>	<ul style="list-style-type: none"> <li>- Rango de funcionamiento grande.</li> <li>- Los obstaculos que puedan haber en el espacio no interfieren en la comunicación.</li> <li>- Consumo bajo</li> </ul>	<ul style="list-style-type: none"> <li>- Necesidad de integrar un dispositivo adicional que convierta los mensajes recibidos por radiofrecuencia en el formato en el que se basa la Plataforma JUGUEMOS, y viceversa.</li> </ul>
<b>Bluetooth</b>	<ul style="list-style-type: none"> <li>- Bajo consumo.</li> <li>- Fácil de implementar.</li> </ul>	<ul style="list-style-type: none"> <li>- Sólo se puede tener emparejados dispositivos por pares.</li> <li>- Es un modo de comunicación de corto alcance.</li> </ul>
<b>WiFi</b>	<ul style="list-style-type: none"> <li>- Permite establecer una red de dispositivos conectados de forma simultánea.</li> <li>- Tiene un alcance medio, pudiendo alcanzar 300 metros, en ausencia de interferencias.</li> <li>- La transmisión de mensajes en la Plataforma JUGUEMOS se realiza mediante Ethernet y Wi-Fi, siendo compatible con la creación de sockets UDP.</li> </ul>	<ul style="list-style-type: none"> <li>- Consumo elevado.</li> </ul>

Tabla 2.1: Comparación entre los diferentes modos de comunicación inalámbrica

En la Tabla 2.1 se encuentran las cuatro posibles comunicaciones inalámbricas junto con algunas de sus ventajas e inconvenientes. A partir de las características de la comunicación por infrarrojos citadas en la tabla y dado que uno de los principales objetivos del proyecto es que el dispositivo permita desplazarse por el espacio interactivo durante el transcurso del juego, este modo de comunicación queda descartado para la realización de este proyecto, ya que podría tener interferencias provocadas por los elementos del espacio o los usuarios. Queda también descartada la comunicación por Radiofrecuencia porque implicaría añadir al menos un elemento más a la Plataforma que se encargara de convertir los mensajes recibidos por radiofrecuencia en mensajes válidos para el sistema de la Plataforma, y viceversa.

Por tanto, los posibles modos de comunicación que cumplirían con los requisitos son Bluetooth y WiFi. El WiFi permite integrar múltiples dispositivos al espacio de una forma sencilla, mientras que el Bluetooth necesita emparejar los dispositivos a pares, pero el WiFi hace uso de

señales de potencia muy superior al Bluetooth, repercutiendo directamente en un mayor consumo de energía. A pesar de esto, como la comunicación dentro de la Plataforma JUGUEMOS se realiza mediante Ethernet o WiFi, si se escogiese establecer la comunicación por Bluetooth se necesitaría incluir otro elemento en el sistema que transformase los mensajes, por lo que se decide finalmente utilizar la comunicación por WiFi, haciendo posible tener una comunicación directa entre el dispositivo diseñado en este proyecto y el sistema de la Plataforma.

Una vez decidido que la comunicación se realizará por WiFi, el siguiente paso es decidir que tecnología emplear para que sea posible. Como se explicó en el apartado anterior, las primeras tecnologías que se describieron quedan descartadas porque no cumplen con los requisitos de este proyecto. Por tanto, las posibles tecnologías son : Arduino Nano 101 [Nan], Raspberry Pi 3 [Ras] y ESP8266 [ESP].

	Arduino Nano	Raspberry Pi 3	ESP8266 (ESP-01)
Conectividad de red (WiFi)	Sí (añadiendo módulo WiFi)	Sí	Sí
Microcontrolador	Sí	Sí	Sí
Voltaje de alimentación	5V	5V	3V a 3.6V
Memoria	Sí	Sí	Sí
Consumo	20 mA	800 mA (4.0 W)	0,9 mA (Standby) - 170 mA (transmitiendo)
Dimensiones	18x45 mm	85,6x53,98 mm	14,3x24,8 mm
Precio	20,00 €	46,59 €	4,50 €

Tabla 2.2: Comparación entre las especificaciones que tiene cada una de las tecnologías

En la tabla 2.2 se analizan las especificaciones necesarias para el desarrollo del proyecto. *Raspberry Pi 3* tiene una gran capacidad de procesamiento y almacenamiento, pero para la realización de este trabajo no es necesario. Además tiene un coste alto en comparación con las otras tecnologías y unas dimensiones elevadas, que provocaría un aumento del tamaño del dispositivo a diseñar y dificultaría el manejo del objeto durante el juego. Por lo tanto *Raspberry Pi 3* queda descartada para este proyecto.

El *ESP8266* se diseñó originalmente como un módulo para dotar de funcionalidad Wi-Fi a otros microcontroladores como *Arduino* o *Raspberry Pi*. Por esta razón, dispone de un limitada capacidad de procesamiento y almacenamiento. En una primera instancia se estudia como opción viable para la realización de este proyecto utilizar la combinación *Arduino Nano+ESP8266*, ya que otros módulos WiFi disponibles para el *Arduino* tienen mayor tamaño. Sin embargo, tras un estudio más detallado del módulo *ESP8266*, se determina que el módulo tiene suficiente capacidad de procesamiento y memoria para implementar en él los procesos gestor y publicador, sin necesidad de ningún otro dispositivo adicional. De esta forma, con un único módulo *ESP8266*, embebido dentro del dispositivo desarrollado en ese proyecto, es posible integrar objetos tangibles en el espacio utilizando el protocolo de comunicación de la Plataforma JUGUEMOS, sin depender de ningún otro elemento.

### 3 Diseño del Hardware

En este capítulo se describe el hardware del dispositivo diseñado, denominado ‘ToyBox’. Primero, se explica el diagrama de bloques del dispositivo ‘ToyBox’ y su integración en el espacio interactivo. A continuación, se mencionan sus componentes más significativos y, por último, la placa de circuito impreso (PCB).

#### 3.1. Diagrama de bloques de ‘ToyBox’

Como se ha explicado anteriormente, la función de ‘ToyBox’ es permitir la integración de objetos tangibles en el espacio interactivo. Además permite que estos objetos puedan ser sensores, es decir, aquellos que provocan algún cambio en el desarrollo del juego, o bien actuadores, es decir, que el juego provoca algún cambio en ellos. ‘ToyBox’ permite también que un mismo objeto pueda disponer de sensores y actuadores que participen simultáneamente en el desarrollo del juego.

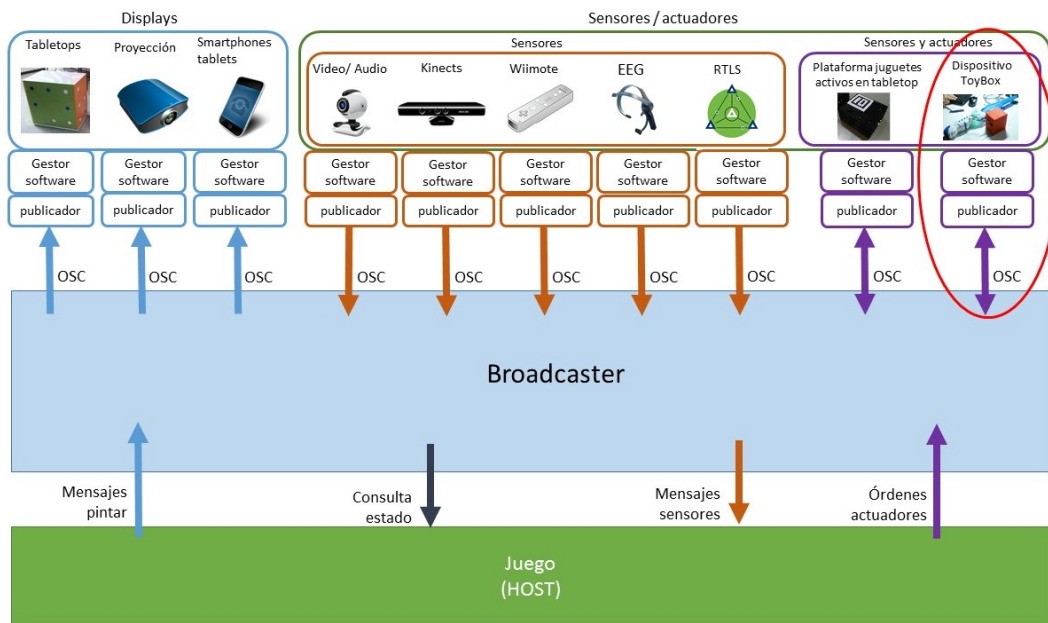


Figura 3.1: Nueva arquitectura de la Plataforma JUGUEMOS

En la Figura 3.1 se describe el esquema de la arquitectura de la Plataforma JUGUEMOS diseñada en el espacio interactivo, tras haber integrado el dispositivo ‘ToyBox’. Como se puede observar, ‘ToyBox’ se comunica con el resto de elementos del espacio mediante el protocolo de red OSC. Además contiene un gestor software, encargado del acceso al hardware del objeto al que se conecta el dispositivo, y un publicador, encargado de establecer la comunicación con el resto de elementos de la Plataforma. En la figura 3.2 se detalla cómo se integra el dispositivo ‘ToyBox’ en la Plataforma JUGUEMOS. Cuando se conecta un sensor al dispositivo, el gestor software captura los datos de éste y, cuando se produce algún cambio, el publicador envía la información de forma inalámbrica al broadcaster. Cuando se conecta un actuador, el publicador



recibe órdenes del broadcaster y éste se las comunica al gestor software, que se las transmite al objeto para que las ejecute.



Figura 3.2: Integración de ‘ToyBox’ en la Plataforma JUGUEMOS

El tipo de interacción tangible del objeto en el espacio interactivo está limitado, puesto que las manipulaciones que el usuario o el sistema pueden hacer en el objeto se limitan a estados activos o inactivos. Es decir, tanto el usuario como el sistema del espacio sólo pueden encender y apagar el objeto electrónico o activar y desactivar una parte de éste. Esto se traduce en que ‘ToyBox’ sólo puede detectar si se abre o se cierra el circuito del objeto conectado (sensor) y sólo puede abrir o cerrar el circuito del objeto (actuador). Esta restricción viene dada por uno de los principales requisitos del proyecto: la conectividad del dispositivo a objetos electrónicos tiene que ser fácil y no requerir conocimientos de electrónica. Para ello, se ha decidido aplicar esta restricción, permitiendo, simplemente recableando el objeto electrónico original, o bien detectar cuándo el usuario conecta o desconecta el objeto, o bien que el sistema pueda conectar o desconectar el objeto. Por ejemplo, si se conecta una espada, capaz de emitir luz y sonido al pulsar un botón, que actúe como sensor, el sistema será capaz de conocer cuándo el jugador ha pulsado el botón de la espada y, por tanto, provocar un cambio en el desarrollo del juego. Por el contrario, si se conecta al dispositivo como actuador, el sistema es capaz de encender o apagar las luces y sonido de la espada.

En la figura 3.3 se muestra el diagrama de bloques de ‘ToyBox’. Los bloques sensor y actuador contienen la electrónica necesaria para simplificar la conexión entre el dispositivo y el objeto por parte del jugador, de manera que el dispositivo dispone de 2 conectores (uno para sensor y otro para actuador). Por lo que el jugador únicamente tiene que conectar el objeto en el conector correspondiente del dispositivo. Además la alimentación de ‘ToyBox’ está conectada al resto del circuito a través de un interruptor, permitiendo al jugador encender y apagar el dispositivo ‘ToyBox’ fácilmente. El bloque de indicador contiene un LED que luce cuando ‘ToyBox’ está encendido. En el siguiente apartado se explican los componentes más importantes del circuito.

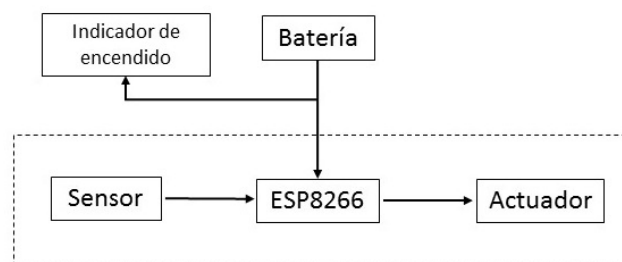


Figura 3.3: Diagrama de bloques de ‘ToyBox’



## 3.2. Componentes de ‘ToyBox’

A continuación se explican las características más importantes de los componentes electrónicos más significativos del dispositivo.

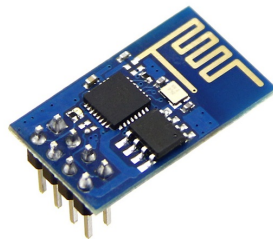
### 3.2.1. Módulo de comunicación WiFi: ESP8266

El módulo ESP8266, modelo ESP-01, mostrado en la Figura 3.4a es un módulo WiFi y que tiene también capacidades de procesamiento y almacenamiento. Dispone de un par de pines digitales disponibles (GPIO0 y GPIO2) y la antena se encuentra impresa en la PCB. Es un módulo pequeño y de bajo costo. Además, gracias a la gran comunidad de desarrolladores y programadores que tiene, dispone de librería para el entorno de programación Arduino y soporta el protocolo de red OSC, utilizado en la Plataforma JUGUEMOS para la transferencia de información entre los elementos del espacio interactivo.

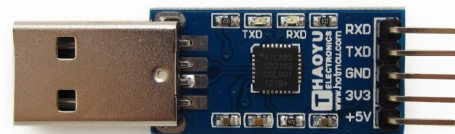
Para programar el procesador interno del ESP8266 se necesita conectarlo a un *Arduino* o a un adaptador USB. En este proyecto se ha utilizado el adaptador mostrado en la Figura 3.4b, el cual dispone de una salida de 3.3V para alimentar al módulo sin necesidad de incluir reguladores de tensión.

En el dispositivo ‘ToyBox’, el bloque sensor se conecta a la entrada GPIO2 y el bloque actuador se conecta a la salida GPIO0.

Las características del módulo ESP8266 (modelo ESP-01) se presentan en el *datasheet* proporcionado por el fabricante [DESP].



(a) Modelo ESP-01



(b) Adaptador USB a FTDI UART

Figura 3.4: Módulo ESP8266 (Modelo ESP-01)

### 3.2.2. Optoacopladores

Un optoacoplador [Opt] es un circuito integrado muy básico compuesto generalmente por un diodo LED y un fototransistor unidos de tal forma que cuando una señal eléctrica circula a través del LED haciendo que se ilumine, la luz que éste emite es recibida por la base del fototransistor, que empieza a actuar en modo saturación. Se suele utilizar para aislar eléctricamente a dispositivos muy sensibles.

En este caso se ha utilizado dos optoacopladores 4N35, formados por un LED y un fototransistor. Uno de ellos está colocado en el bloque sensor y el otro en el bloque actuador. Debido

a la restricción de que lo único que el dispositivo puede hacer es abrir o cerrar el circuito del objeto, o bien detectar cuándo esto ocurre, la función de los optoacopladores es aislar el objeto al que se conecte ‘ToyBox’ del resto del circuito, de manera que la alimentación del objeto no afecte al circuito y viceversa. En la Figura 3.5 se muestra el esquema de un optoacoplador 4N35.

Las características del optoacoplador 4N35 se encuentran explicadas en el *datasheet* proporcionado por el fabricante [DOpt].

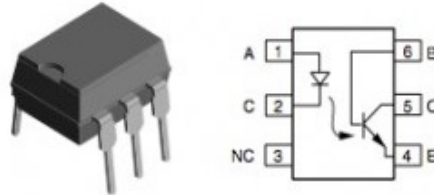


Figura 3.5: Optoacoplador 4N35

### 3.2.3. Alimentación de ‘ToyBox’

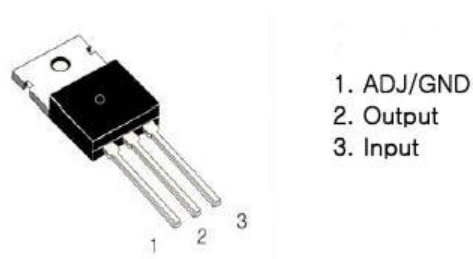
Como se ha dicho anteriormente, la alimentación del dispositivo ‘ToyBox’ es independiente de la alimentación del objeto al que se conecta, por lo que el dispositivo necesita una batería que alimente a los componentes del circuito. Además, el rango de tensión de alimentación del *ESP8266* es de 3V a 3.6V, por tanto es necesario una batería que proporcione un voltaje mayor a este rango y un regulador de tensión que tenga a la salida 3.3V.

La batería debe tener unas dimensiones compatibles con el tamaño especificado del dispositivo. Por esta razón, se decide alimentar el circuito con dos pilas botón de 3V conectadas en serie, pero, tras realizar pruebas durante el prototipado, se observa que no es suficiente para alimentar al módulo WiFi. Finalmente, se decide utilizar una batería de 9V de tipo pp3 (Figura 3.6a). El regulador elegido es LM1117-3.3V (Figura 3.6b). Se trata de un regulador de tensión que, para una tensión de entrada comprendida entre 4.75V y 10V, proporciona a la salida una tensión de 3.3V. Las especificaciones del regulador elegido se encuentran en el *datasheet* del fabricante [DReg].

El dispositivo contiene también un interruptor, que permite al usuario apagarlo y encenderlo sin necesidad de acceder al interior del encapsulado, y un LED que luce cuando ‘ToyBox’ está encendido.



(a) Batería de 9V tipo pp3



(b) Regulador de tensión LM1117-3.3V

Figura 3.6: Bloque de alimentación del dispositivo ‘ToyBox’

### 3.3. Placa de circuito impreso

La placa de circuito impreso, conocida como PCB, es un componente hardware fundamental ya que conecta eléctricamente a través de las pistas conductoras, y sostiene mecánicamente, por medio de la base, un conjunto de componentes electrónicos. De este modo, se obtiene un circuito electrónico más compacto.

Teniendo en cuenta la restricción del tamaño, la PCB se debe diseñar de manera que el circuito ocupe el menor espacio posible. Es importante también tener en cuenta que las pistas pueden tener anchuras diferentes, en función de la intensidad que pase por ellas. El módulo WiFi no estará soldado en la PCB, sino que se soldarán dos tiras de pines y el módulo se conectará a estos pines. De esta forma, se cumple el requisito de tener acceso al microprocesador para posibles modificaciones en el software.

El proceso de fabricación de la PCB junto con la explicación de su funcionamiento se describen en el capítulo 5.



## 4 Diseño del Software

En este capítulo se explica cómo se realiza la integración software del dispositivo en la Plataforma JUGUEMOS y la programación del módulo ESP8266.

### 4.1. Integración software del dispositivo ‘ToyBox’ en la Plataforma JUGUEMOS

La transmisión de datos entre el dispositivo y el proceso Broadcaster de la Plataforma JUGUEMOS se realiza a través de comunicación WiFi. Para ello, se utiliza el protocolo OSC, basado en el envío de mensajes a partir de la creación de sockets UDP, el cual es el mismo protocolo que utilizan el resto de elementos de la Plataforma JUGUEMOS. A continuación, se explica cuál es el proceso software para integrar el dispositivo desarrollado en este TFG en la Plataforma JUGUEMOS.

Para integrar un dispositivo en el espacio interactivo es necesario desarrollar un software encargado de la conexión directa con el hardware del módulo *ESP8266*. Este software se encarga de leer los datos recibidos del GPIO2 y escribir datos en el GPIO0. Una vez implementado el gestor software, hay que implementar un proceso publicador para la transferencia de datos con el broadcaster de la Plataforma JUGUEMOS. Lo primero que hace el publicador es conectarse al broadcaster enviando un mensaje OSC con el formato mostrado en la Figura 4.1), para lo cual es necesario importar la librería de comunicación OSC (*esp8266-OSC-master* [LOE]) al módulo *ESP8266*. En address el proceso declara la intención de conectar con el broadcaster especificando que se trata de un publicador asociado a un gestor sensor/actuador. En el parámetro se especifica el número de puerto UDP en el que el publicador escuchará los mensajes del broadcaster. Una vez establecida la conexión, el dispositivo puede enviar mensajes al proceso broadcaster y recibir aquellos mensajes que el broadcaster le envía.

Address (string)	Parámetro (int)
/sensor/connect	puerto

Figura 4.1: Mensaje OSC de conexión

Se distinguen dos tipos de mensajes: mensajes de sensor (envía estado del objeto al broadcaster) y mensajes de actuador (recibe órdenes desde el broadcaster para ser ejecutadas en el objeto). El formato del mensaje OSC a enviar para mandar el estado actualizado de un sensor conectado al dispositivo diseñado en este proyecto se muestra en la Figura 4.2. En el campo *address* se identifica el tipo de dispositivo que envía el mensaje, en este caso se trata de un dispositivo ‘ToyBox’. El campo *manipulación* indica el tipo de manipulación que el objeto detecta (pasivo) o realiza (activo), como el dispositivo ‘ToyBox’ sólo admite manipulaciones binarias, éstas son siempre de tipo ‘add’. El campo *ID objeto* permite identificar múltiples dispositivos de tipo ‘ToyBox’, permitiendo utilizar más de un dispositivo en un juego o aplicación en el espacio JUGUEMOS. Cada dispositivo ‘ToyBox’ tendrá un ID objeto distinto. El campo *ID sesion* se utiliza para otros elementos de la Plataforma JUGUEMOS, en este caso, como no

se utiliza para el tipo de dispositivo ‘ToyBox’, se pone el valor ‘0’ por defecto. Por último, el campo *Valor 1* puede ser 0 (circuito abierto) o 1 (circuito cerrado), en función del valor binario (0 ó 1) que el gestor software del dispositivo captura, de forma iterativa, en el pin GPIO2 del módulo *ESP8266*.

Address (string)	Manipulación (string)	ID objeto (string)	ID sesión (string)	Valor 1 (float)
/sensor/0D/toybox	add			0/1

Figura 4.2: Formato de mensaje OSC que envía el dispositivo ‘ToyBox’

Address (string)	ID objeto (int)	ID sesión (int)	Valor 1 (float)
/actuador/0D/toybox			0/1

Figura 4.3: Formato de mensaje OSC que envía el broadcaster a ‘ToyBox’

El formato de los mensajes OSC que el proceso broadcaster envía al dispositivo ‘ToyBox’ es el mostrado en la Figura 4.3. En este caso, el campo *address* contiene la palabra ‘actuador’. El campo *Valor 1* indica si se debe cerrar el circuito del objeto (valor = 1) o si se debe abrir (valor = 0).

## 4.2. Programación del módulo ESP8266

A continuación se detallan los flujos de ejecución de los procesos gestor y publicador implementados en el módulo *ESP8266* para establecer la comunicación descrita en el apartado anterior. Estos procesos están desarrollados a través de la plataforma software *Arduino*. Para ello, se instalaron las librerías *esp8266-OSC-master* [LOE] y *esp8266-2.3.0* [LEA].

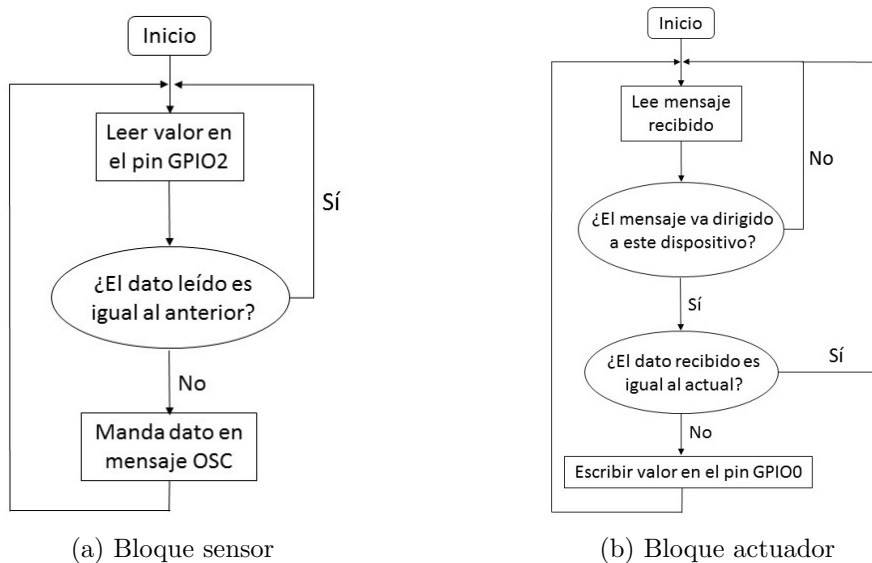


Figura 4.4: Diagrama de flujo de la programación del ESP8266

El módulo *ESP8266* ejecuta los procesos descritos en la Figura 4.4. El módulo ESP8266 lee el dato proporcionado por el sensor, si está conectado, comprueba si es diferente al leído anteriormente y, si es así, envía el nuevo dato al broadcaster a través de un mensaje OSC con el formato descrito en el apartado anterior. Además lee los mensajes recibidos y, cuando un mensaje va dirigido a él, compara el dato recibido con el valor actual y, si es distinto, proporciona ese valor al actuador, si está conectado, para que ejecute la orden.





## 5 Prototipado

En este capítulo se detalla el prototipado del dispositivo ‘ToyBox’. Para ello, primero se describe el circuito diseñado y el prototipado de la parte electrónica del dispositivo. Por último, se explica el prototipado del encapsulado del dispositivo.

### 5.1. Prototipado electrónico

Para llevar a cabo el prototipado de ‘ToyBox’, primero se debe diseñar el circuito electrónico que permita la integración de objetos electrónicos en el espacio interactivo, tal y como se ha explicado anteriormente, y cumpla con los requisitos establecidos. Para ello, se realizan una serie de pruebas a través de prototipos experimentales construidos en una protoboard, o placa de pruebas, (Figura 5.1).

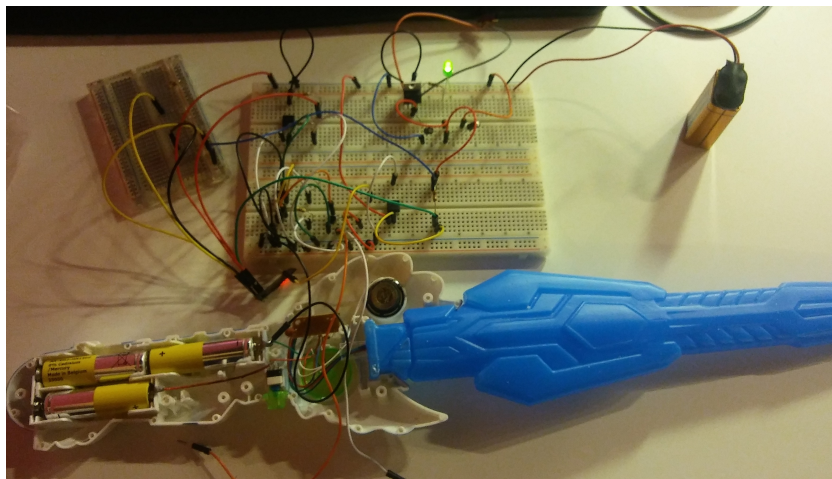


Figura 5.1: Prototipo experimental del circuito electrónico de ‘ToyBox’

A partir de estas pruebas, se deciden los componentes electrónicos necesarios para el correcto funcionamiento de ‘ToyBox’, obteniendo el circuito mostrado en la Figura 5.2. Como se ha comentado en el capítulo 3, la alimentación del dispositivo se controla a través de un switch y contiene un regulador que proporciona a su salida un valor de 3.3V. Además, ‘ToyBox’ dispone de un LED que luce cuando está encendido. Cuando se quiere detectar la manipulación que el usuario le hace a un objeto, éste se conectará al bloque sensor del dispositivo provocando que el LED del optoacoplador luzca, o no, en función de la manipulación realizada. Esto provocará que el GPIO2 del *ESP8266* reciba 3.3V o 0V. Cuando se produzca un cambio en el valor que recibe el GPIO2, el *ESP8266* enviará un mensaje OSC al broadcaster con el valor del dato recibido. Si se conecta un objeto al bloque actuador, el optoacoplador de este bloque lucirá, o no, en función de si el *ESP8266* recibe una orden para el objeto, provocando que el circuito del objeto conectado se cierre o se abra. Como la corriente que necesita el objeto para su correcto funcionamiento es mayor que la corriente máxima de colector del fototransistor que tiene el optoacoplador, se ha añadido tres transistores BC547 en cascada, obteniendo a la salida una corriente suficiente para que el objeto funcione correctamente. Como los bloques sensor y actuador son independientes y se conectan a distintos pines del módulo *ESP8266*, ‘ToyBox’

es capaz de detectar manipulaciones en un objeto y hacer que éste realice las órdenes que le manda el sistema, simultáneamente.

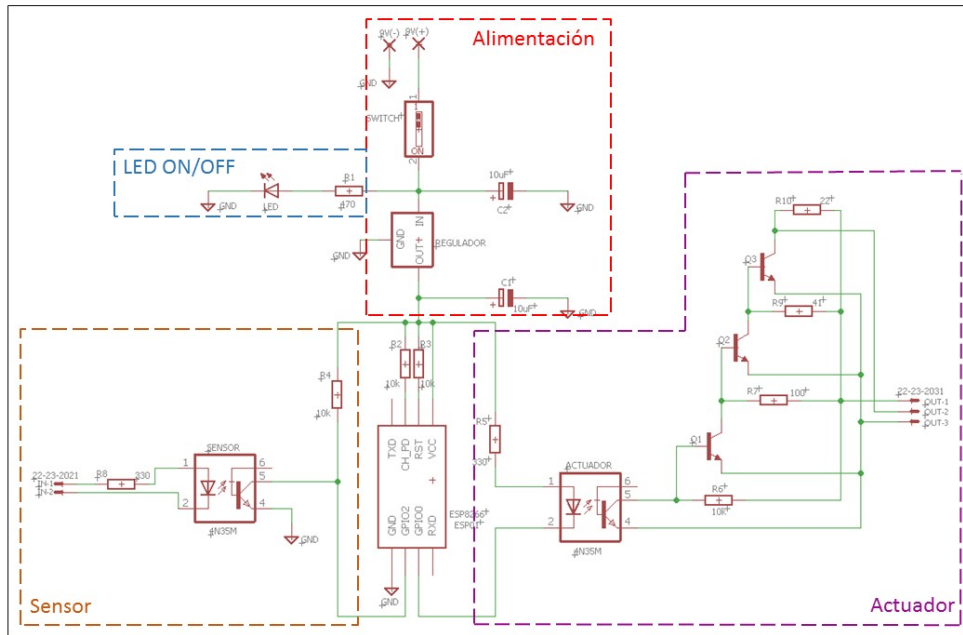
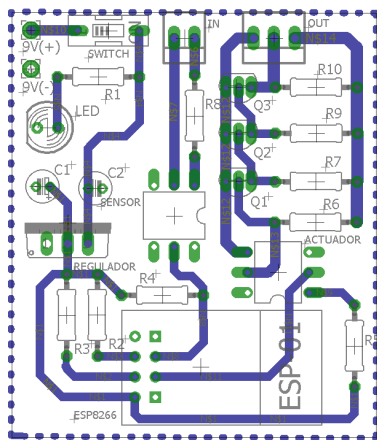


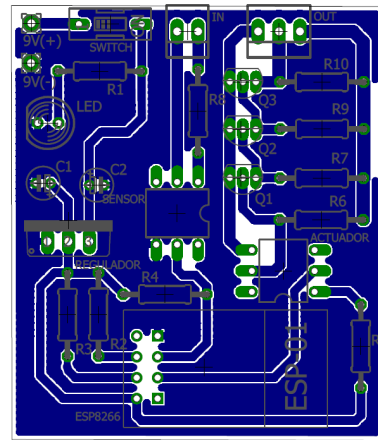
Figura 5.2: Esquemático del circuito electrónico del dispositivo

Una vez obtenido el circuito que cumple correctamente con los objetivos, se diseña a través del software *Eagle 8.2.2* una PCB que contenga el circuito diseñado. El principal requisito para el diseño de la PCB es que sea compacta, de manera que tenga el menor tamaño posible.

Además, se decide colocar los conectores y el switch en la misma cara del encapsulado, para simplificar su diseño y conseguir que el dispositivo sea lo más compacto posible. También hay que tener en cuenta a la hora de diseñar la PCB los tamaños de los componentes electrónicos del circuito y la distancia entre las pistas de la PCB. Para elegir el ancho de cada una de las pistas de la PCB se debe tener en cuenta la corriente que circula por cada parte del circuito y la anchura entre las patas de los componentes.



(a) Diseño de la PCB

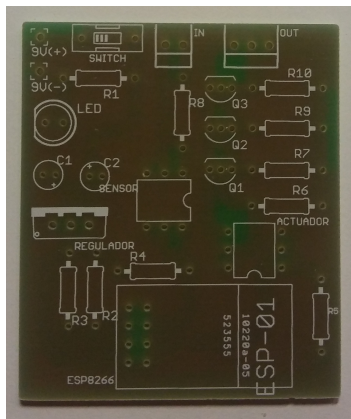


(b) Diseño de la PCB con el plano de masa

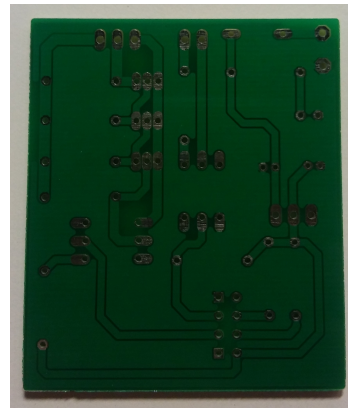
Figura 5.3: Diseño de la PCB del dispositivo

La Figura 5.3 muestra la PCB diseñada a través del software *Eagle 8.2.2*. Como se puede observar en la figura, todas las pistas se encuentran en la capa inferior de la PCB. De esta manera, quedan todos los componentes electrónicos del circuito en la cara superior y las conexiones entre ellos en la cara inferior.

Una vez terminado el diseño por software de la PCB, se generan los *gerbes*, que son unos archivos con extensión *.gbr* que contienen toda la información del diseño de la PCB. Estos archivos se envían a *Seed Studio* [See] para que fabriquen 10 PCB's con el diseño realizado. Además de los *gerbes*, se deben introducir algunas especificaciones de la PCB referentes a materiales, dimensiones y características de las capas y pistas. La altura de las pistas se establece en  $35\mu\text{m}$ . El motivo de esta elección de altura de pistas es que si se elige una altura inferior, las pistas pueden curvarse y despegarse de la PCB cuando se acerca la punta del soldador caliente para soldar los componentes. En la Figura 5.4a se puede observar la cara superior de la PCB ya fabricada y en la Figura 5.4b, la cara inferior.



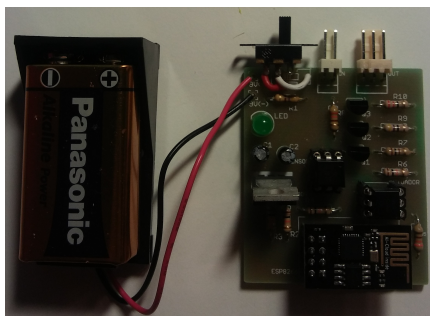
(a) Cara superior



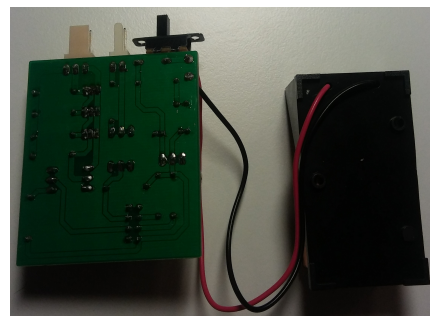
(b) Cara inferior

Figura 5.4: Prototipo de la PCB fabricada

La soldadura de todos los componentes a la PCB se realiza con un soldador de estaño JBC. Se utiliza un estaño que contiene en su interior una resina especial que facilita la distribución uniforme del estaño sobre las superficies a unir y evita, al mismo tiempo, la oxidación producida por la temperatura demasiado elevada del soldador. El módulo *ESP8266* se conecta a la PCB a través de pines, soldados en la placa. De esta manera, se puede extraer fácilmente para modificar su software. La Figura 5.5 muestra el prototipo de la placa electrónica de 'ToyBox', una vez soldados todos sus componentes.



(a) Cara superior



(b) Cara inferior

Figura 5.5: Prototipo de la placa electrónica de 'ToyBox'

## 5.2. Prototipado del encapsulado

Una vez comprobado el correcto funcionamiento, se comienza el diseño del prototipo del encapsulado del dispositivo ‘ToyBox’.

En primer lugar, utilizando un calibre, se toman las medidas de la placa electrónica. A continuación, a partir de las medidas tomadas anteriormente y mediante el software *SketchUp*, se modela en 3 dimensiones una imagen de la PCB y sus componentes (Figura 5.6). En la figura se pueden distinguir los componentes más significativos del dispositivo: el bloque rojo corresponde con la pila que alimenta el circuito electrónico, el bloque verde representa la PCB, el bloque blanco corresponde con el resto de componentes de la placa y, en la parte frontal, se encuentran los conectores y el interruptor. Con el objetivo de que el tamaño final de ‘ToyBox’ sea el menor posible, se decide que la distribución en el interior del encapsulado será el mostrado en la figura, es decir, la pila estará debajo de la placa.

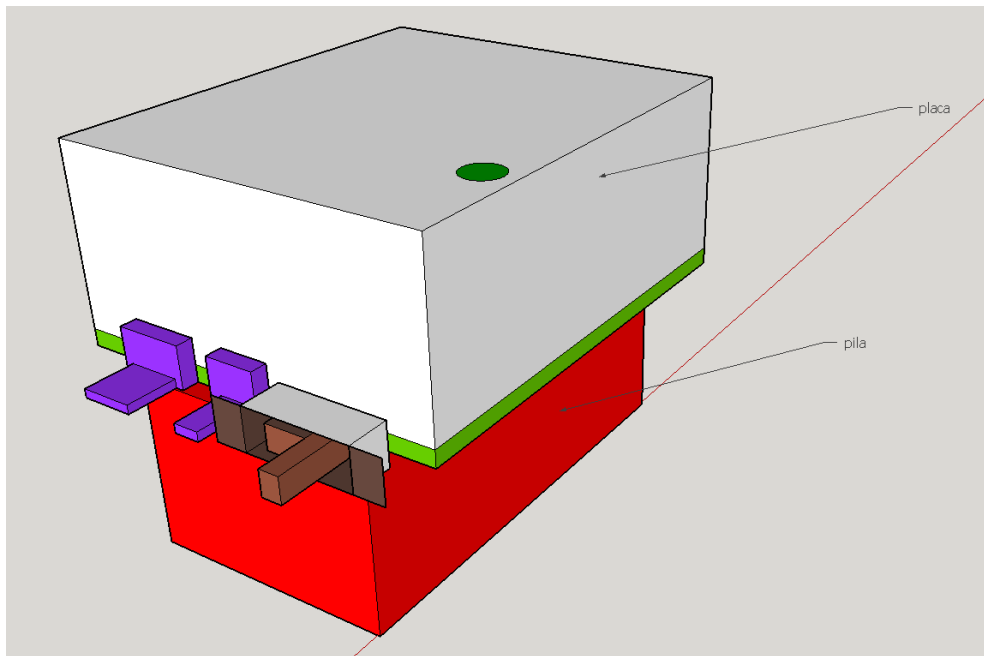


Figura 5.6: Modelo 3D del prototipo electrónico de ‘ToyBox’

Una vez modelados los componentes electrónicos del dispositivo, se diseña el encapsulado. El encapsulado está formado por una única pieza que se cierra con una tapa mediante un tornillo, consiguiendo que el dispositivo sea compacto. Además debe permitir al usuario conectar y desconectar los objetos que se desean integrar en el espacio interactivo y poder sacar los componentes para sustituir la pila o extraer el módulo *ESP8266* para modificar su programación. En la Figura 5.7 se muestra el modelo 3D del encapsulado. La cara delantera tiene una ventana (Figura 5.7a) en la que estarán ubicados los conectores e interruptor de la placa electrónica (Figura 5.7d). De esta forma, el usuario podrá conectar los objetos y encender o apagar el dispositivo fácilmente. El encapsulado dispone también de un pequeño agujero en la cara superior (Figura 5.7b) ubicado en la posición en la que se encuentra el LED de la placa electrónica. Además, el interior del encapsulado está dividido en dos secciones (Figura 5.7c) que separan la placa, ubicada en la parte superior, de la pila, ubicada en la parte inferior (Figura 5.7e).

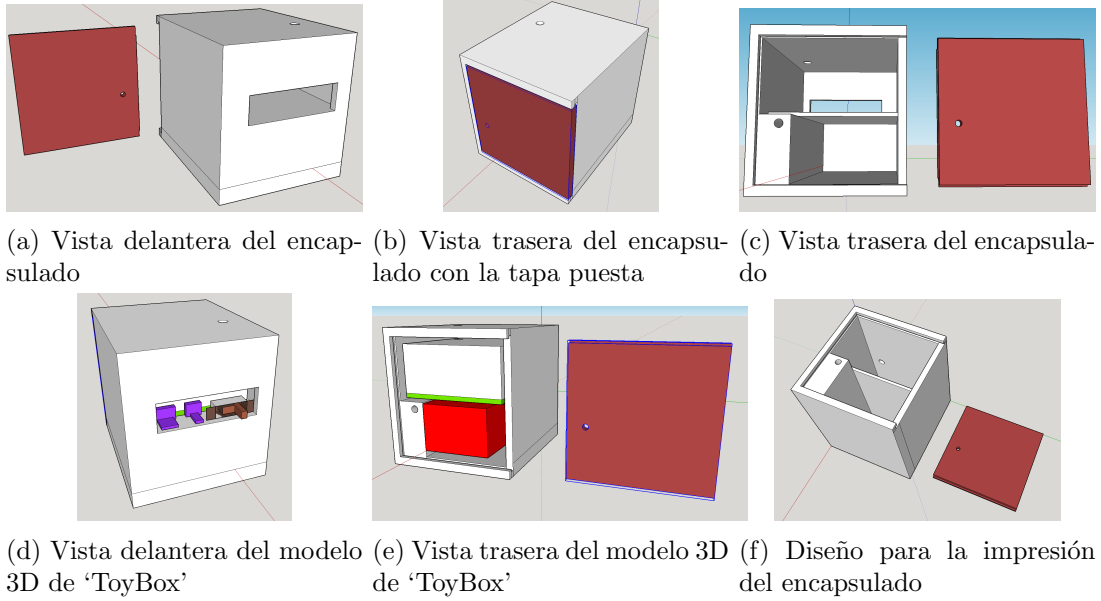
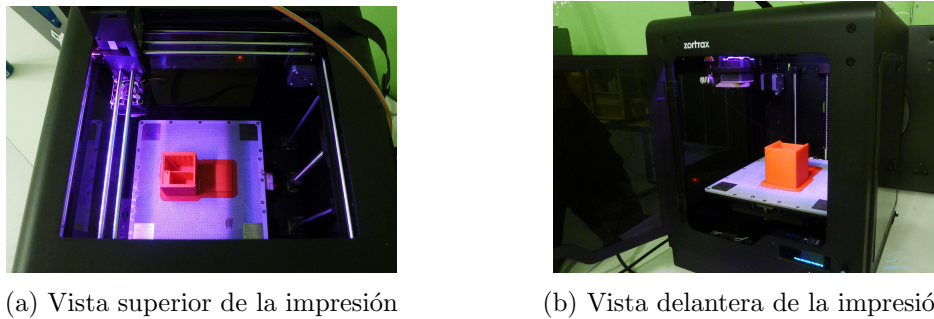


Figura 5.7: Modelo 3D del encapsulado

Una vez terminado el diseño del modelo 3D del encapsulado del dispositivo 'ToyBox', se procede a su fabricación. Para ello, se utiliza una impresora 3D *Zortrax M200* [Imp], a la que se le carga un archivo *.stl* con el diseño mostrado en la Figura 5.7f. A partir de ese archivo, la impresora imprime el encapsulado en *abs*, por ser este un material muy resistente (Figuras 5.8a y 5.8a).

Figura 5.8: Impresión realizada por la impresora 3D *Zortrax M200*

Tras varias pruebas y ajustes de tamaños, se obtiene el prototipo de encapsulado mostrado en la Figura 5.9.

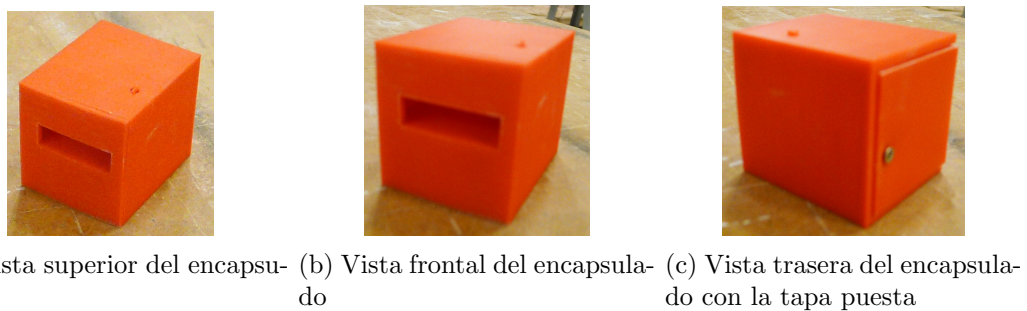


Figura 5.9: Prototipo de encapsulado



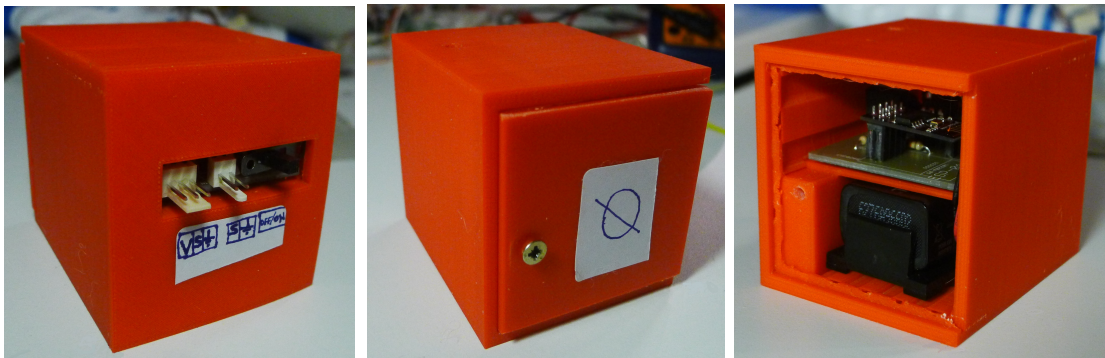


## 6 Resultados obtenidos

En este capítulo se presentan los resultados obtenidos en el desarrollo del proyecto. En el primer apartado, se muestra el dispositivo final ‘ToyBox’ diseñado, en el segundo apartado, se explica cómo se conectan objetos al dispositivo ‘ToyBox’ para ser integrados en el espacio interactivo y, en el tercer apartado, se muestran tres casos de uso que se han realizado a modo de demostración técnica del correcto funcionamiento del dispositivo ‘ToyBox’.

### 6.1. Dispositivo ‘ToyBox’

El dispositivo diseñado en este proyecto (Figura 6.1) tiene unas dimensiones de 5.5x5.9x6.6cm, lo que permite un fácil manejo del objeto conectado. Además, el dispositivo es compacto, gracias al encapsulado, y la conexión del objeto es sencilla (Figura 6.1a). También permite el acceso a la alimentación y al módulo *ESP8266*, ya que es posible acceder al interior del encapsulado retirando la tapa, la cual está sujeta con un tornillo (Figura 6.1b). La distribución de los componentes electrónicos en el interior del encapsulado se muestran en la Figura 6.1c). Se ha etiquetado el dispositivo, de manera que tiene indicado cómo se debe conectar los cables del objeto (Figura 6.1a) y el identificador del dispositivo dentro de la Plataforma JUGUEMOS (Figura 6.1b).



(a) Vista frontal de ‘ToyBox’ (b) Vista trasera de ‘ToyBox’ con tapa (c) Vista trasera de ‘ToyBox’ sin tapa

Figura 6.1: Dispositivo ‘ToyBox’

Para integrar el objeto conectado a ‘ToyBox’, simplemente hay que encender el dispositivo, a través del interruptor, y éste establece, automáticamente, la conexión con el broadcaster de la Plataforma JUGUEMOS. Esta conexión se realiza en 1 ó 2 segundos. A partir de ese momento el objeto está integrado en la Plataforma JUGUEMOS y se puede utilizar en el transcurso del juego. Dado que la comunicación entre ‘ToyBox’ y el broadcaster se realiza por WiFi, el objeto conectado al dispositivo puede desplazarse por todo el espacio interactivo durante el desarrollo del juego sin que se produzcan interrupciones en la comunicación.

Durante la realización de las pruebas de funcionamiento del dispositivo ‘ToyBox’ en el espacio interactivo JUGUEMOS, se observa que la comunicación no sufre interrupciones en ningún momento. Además, no existe *lag* apreciable entre el momento en el que el usuario realiza

una manipulación en el objeto y el momento en el que el broadcaster recibe la información.

## 6.2. Conexión de un objeto electrónico a ‘ToyBox’

Para conectar un objeto electrónico al dispositivo ‘ToyBox’ sólo es necesario recablear su circuito interno. Por tanto, si se desea detectar las manipulaciones del usuario y transmitírselas al broadcaster de la Plataforma JUGUEMOS se debe recablear el circuito del objeto como se muestra en la Figura 6.2a. El objeto se debe conectar a ‘ToyBox’ mediante el conector de dos pines, de manera que el pin del lado derecho (etiquetado como *tierra*) se conecta al polo negativo de la alimentación del objeto y el pin de la izquierda (etiquetado como *señal*) se conecta entre el elemento que actúa como interruptor y la electrónica del objeto. De esta forma ‘ToyBox’ detecta cuándo se cierra o se abre el interruptor.

Para que el sistema pueda mandar órdenes al objeto y éste las ejecute, el objeto electrónico se debe recablear como se muestra en la Figura 6.2b. El objeto se debe conectar a través del conector de 3 pines del dispositivo, de forma que el pin etiquetado como *tierra* se conecta al polo negativo de la alimentación del objeto, el pin etiquetado como *alimentación* se conecta al polo positivo de la alimentación del objeto y el pin etiquetado como *señal* se conecta en la parte positiva del circuito electrónico que será controlada por el sistema de la Plataforma JUGUEMOS.

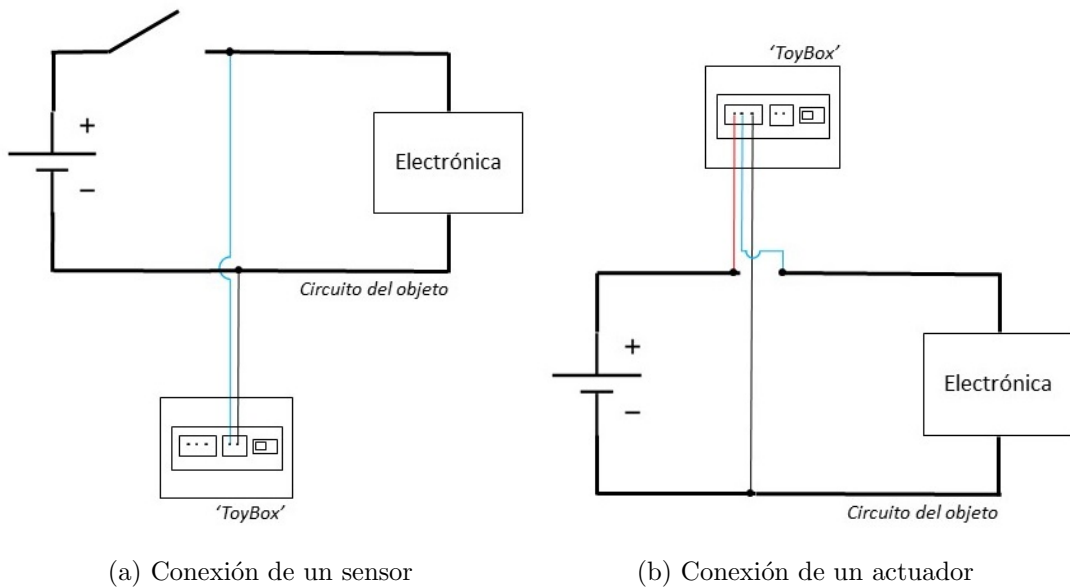


Figura 6.2: Conexión de un objeto electrónico al dispositivo ‘ToyBox’

## 6.3. Demostración técnica del funcionamiento de ‘ToyBox’

Para demostrar el correcto funcionamiento de ‘ToyBox’ en la Plataforma JUGUEMOS se han desarrollado tres casos de uso. ‘ToyBox’ permite integrar al espacio interactivo JUGUEMOS dos tipos de objetos: objetos electrónicos y objetos convencionales en los que se añade un circuito electrónico.



### 6.3.1. Integración de objetos electrónicos al espacio interactivo

Para integrar objetos electrónicos en el espacio JUGUEMOS únicamente hay que recablear su circuito como se explica anteriormente. A continuación se muestran dos casos de objetos integrados en el espacio interactivo para la demostración el funcionamiento de ‘ToyBox’.

#### *Cerdito de marioneta*

El primer objeto que se quiere integrar en el espacio es una marioneta. Se trata de un *cerdito* que tiene un pulsador situado en la tripa, lugar donde se introduce la mano, de manera que cuando se pulsa suena el guarrido del cerdo (Figura 6.3). Se desea integrar este juguete en un juego para el espacio interactivo JUGUEMOS, para lo cual el sistema necesita detectar cuándo el usuario aprieta el pulsador del *cerdito*.



(a) Vista frontal



(b) Vista lateral

Figura 6.3: Cerdito de marioneta

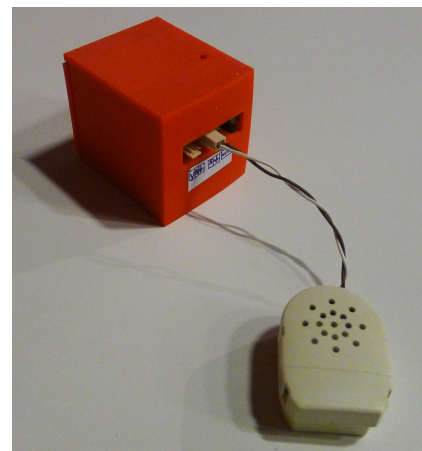
Para conectarlo a la entrada de sensor del dispositivo ‘ToyBox’, se han añadido dos cables al circuito del pulsador. El cable que se conecta a *tierra* en ‘ToyBox’ se ha añadido en el polo negativo de la alimentación del pulsador y el cable que se conecta a *señal* en ‘ToyBox’ se ha añadido a la entrada del altavoz. Estas conexiones se muestran en la Figura 6.4a. En la Figura 6.4b se muestra el pulsador con el conector añadido y en la Figura 6.4c se muestra el pulsador conectado a ‘ToyBox’.



(a) Pulsador recableado



(b) Pulsador con el conector



(c) Pulsador conectado a ‘Toy-Box’

Figura 6.4: Pulsador de la marioneta

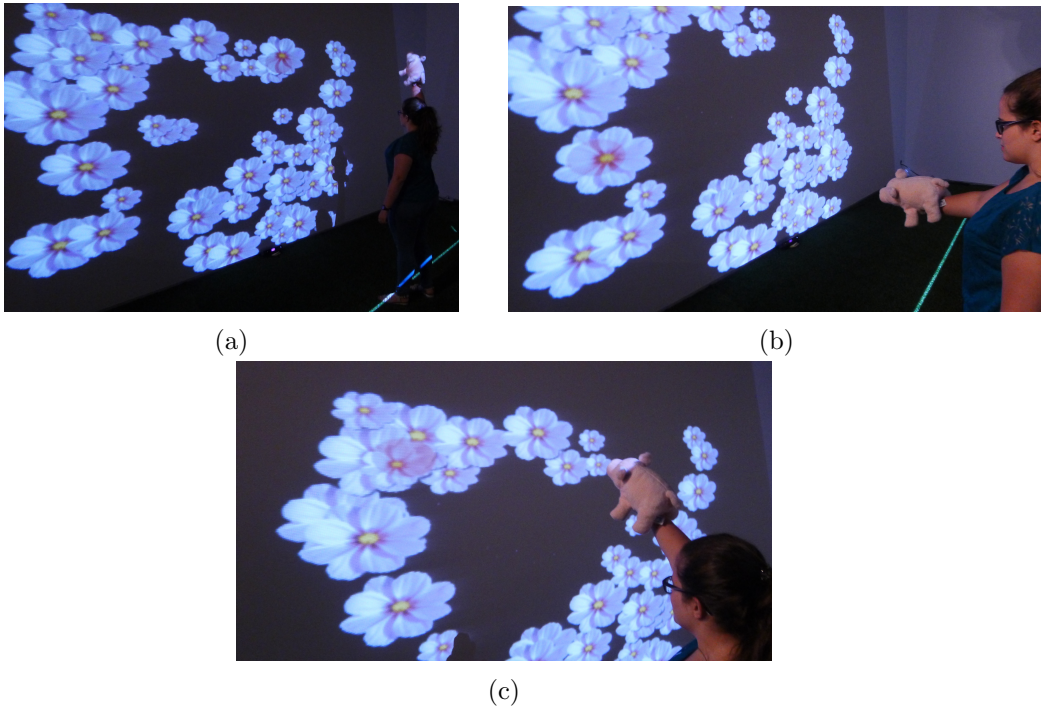
Para facilitar el manejo de la marioneta durante el desarrollo del juego, se introduce el dispositivo ‘ToyBox’ en el interior del *cerdito* (Figura 6.5a). A continuación se conecta el pulsador a ‘ToyBox’ (Figura 6.5b). Por último, se coloca el pulsador en la tripa del *cerdito*, donde se colocará la mano (Figura 6.5c). Una vez se encienda ‘ToyBox’, la marioneta estará integrada en el espacio interactivo.



(a) Dispositivo ‘ToyBox’ dentro del *cerdito*      (b) Conexión del pulsador a ‘ToyBox’      (c) Pulsador colocado en la tripa del *cerdito*

Figura 6.5: Introducción de la electrónica en la marioneta

Para demostrar la integración del juguete en el espacio interactivo JUGUEMOS, se ha desarrollado una pequeña demo en la que el *cerdito* se ‘come’ las flores que aparecen en una pantalla del espacio interactivo. Para ello, se muestra un círculo rojo en la pantalla que representa la posición de la marioneta en el espacio, dato conocido a través del sensor Kinect, y al darle al pulsador del *cerdito*, éste emite el guarrido del cerdo mientras se come la flor que se encuentre dentro del círculo rojo. En la figura 6.6 se muestran momentos del juego.



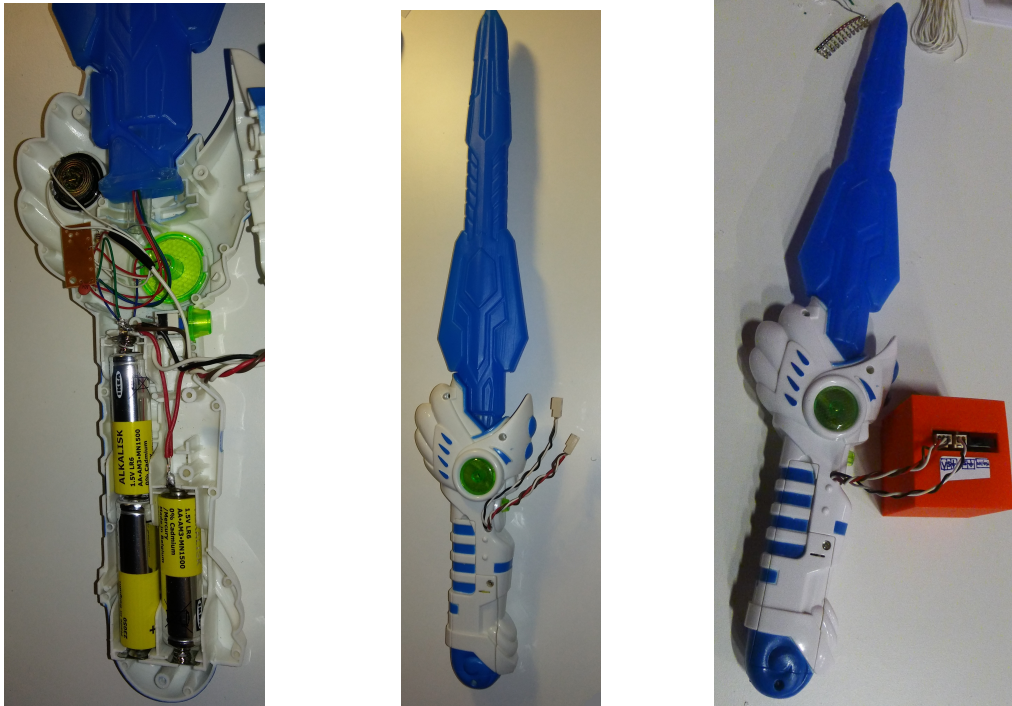
(a)      (b)      (c)

Figura 6.6: Demo con el *cerdito* de marioneta

### Espada

El siguiente objeto electrónico que se quiere integrar en el espacio interactivo JUGUEMOS es una espada que emite luz y sonido al pulsar un botón. Se desea integrar este juguete en el espacio JUGUEMOS, de forma que el sistema pueda, por un lado, detectar cuándo el usuario ha pulsado el botón de la espada y, por otro lado, ordenar a la espada que emita luz y sonido.

Para ello, se ha modificado la electrónica de la espada, separando el botón del resto del circuito electrónico, de manera que al pulsar el botón no se enciende la espada. Además se han añadido cables para que pueda funcionar de sensor y actuador (Figura 6.7a). Para actuar como sensor, se ha añadido el cable de *tierra* al polo negativo de la alimentación de la espada y el cable de *señal* al botón, de manera que el sistema sabrá cuándo pulsa el usuario el botón de la espada. Para la parte de actuador, se ha añadido el cable de *alimentación* al polo positivo de la alimentación de la espada, el cable de *tierra* al polo negativo y el cable de *señal* a la parte del circuito donde estaba conectado el botón. Así, la Plataforma JUGUEMOS podrá controlar cuándo se encienden las luces y sonido. En la Figura 6.7b se muestra la espada con los conectores y en la Figura 6.7c se muestra la espada conectada a ‘ToyBox’.



(a) Circuito de la espada modificado (b) Espada con los conectores (c) Espada conectada a ‘Toy-Box’

Figura 6.7: Espada

Se ha desarrollado una pequeña demo que demuestra su integración en el espacio. La demo consiste en encontrar un monstruo que está escondido detrás de uno de los arbustos proyectados en pared y mesas NIKVision (Figura 6.8a). Mediante el sistema RTLS de Ubisense, se localiza la posición del usuario dentro del espacio interactivo y, de esta manera, el sistema de la Plataforma JUGUEMOS puede conocer a qué arbusto se acerca el usuario. Cuando el usuario esté cerca del arbusto en el que está el monstruo, la espada emitirá luz y sonido (Figura 6.8b), indicando que el monstruo está en ese arbusto. Cuando el jugador pulsa el botón, el arbusto se abre, dejando ver lo que hay detrás (Figura 6.8c).



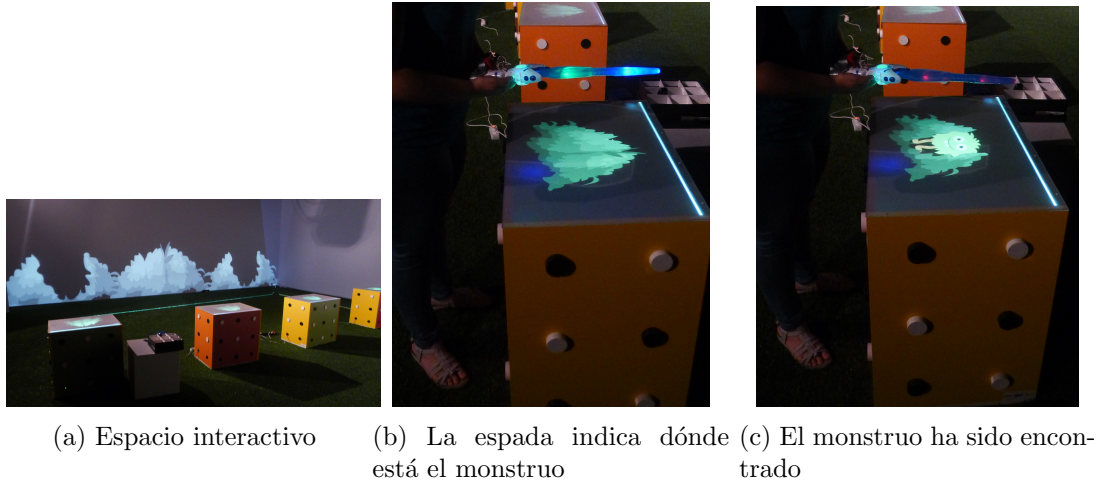


Figura 6.8: Demo con la espada

### 6.3.2. Integración de objetos no electrónicos en el espacio interactivo

Además de integrar objetos electrónicos en el espacio interactivo JUGUEMOS, ‘ToyBox’ permite integrar objetos que, originalmente, no disponen de electrónica propia, pero que se le añade un circuito electrónico para convertirlos en interactivos dentro del espacio JUGUEMOS. A continuación se muestra un ejemplo de un juguete no electrónico y cómo se integra en el espacio JUGUEMOS.

#### *Regadera*

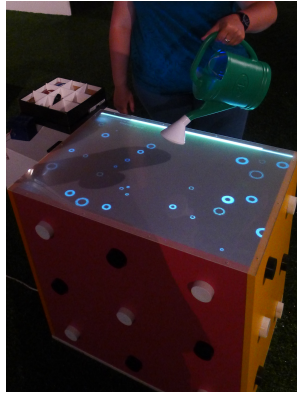
El juguete que se desea integrar en el espacio interactivo es una regadera (Figura 6.9a).

Primero, se diseña el circuito que se añadirá a la regadera (Figura 6.9b). Este circuito está formado por un *Arduino Micro* y un acelerómetro, alimentados con una pila de 9V. El Arduino se ha programado de forma que, cuando el acelerómetro se inclina unos determinados grados, el pin 12 de la placa Arduino se pone en *alto*. Este pin es el que se conectará al pin de *señal* del dispositivo ‘ToyBox’. El circuito, por tanto, se encarga de detectar cuándo se inclina la regadera hacia delante (posición de regar). Una vez conectado a ‘ToyBox’, se introduce en el interior de la regadera y se coloca el acelerómetro en la parte trasera (Figura 6.9c)

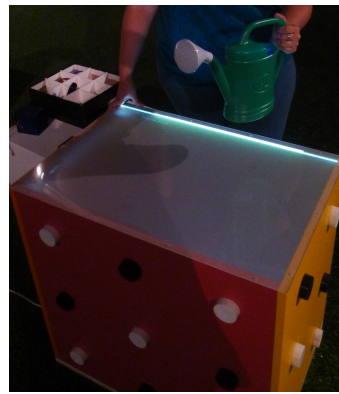


Figura 6.9: Regadera

Se ha desarrollado un pequeño juego a modo de demostración de integración de objetos no electrónicos en el espacio interactivo. El juego consiste en regar las plantas que aparecen proyectadas en las mesas NIKVision. Para ello, se debe inclinar la regadera sobre las mesas. El sistema RTLS de Ubisense se encarga de detectar sobre qué mesa está la regadera, mientras que la electrónica de arduino+acelerómetro se encarga de detectar cuándo se inclina la regadera lo suficiente para estar en la posición de echar agua. Durante el juego, aparecen gotas en la pantalla (Figura 6.10a), si se inclina la regadera, o está la pantalla en negro, si se mantiene recta (Figura 6.10b).



(a) Regadera inclinada



(b) Regadera recta

Figura 6.10: Demo con la regadera



## 7 Conclusiones y trabajo futuro

En este último capítulo se resumen la consecución de objetivos y el trabajo futuro que se puede realizar a partir de este Trabajo Fin de Grado.

### 7.1. Conclusiones

Se ha realizado un análisis exhaustivo de las tecnologías que están en uso actualmente en el campo de la Interacción Tangible. A continuación se ha realizado un estudio comparativo de las tecnologías, escogiendo aquella más adecuada para la realización de este trabajo. Seguidamente, se han diseñado los bloques electrónicos y el software del dispositivo y se han fabricado tres dispositivos funcionales ‘ToyBox’. Por último, se ha desarrollado varios casos de uso que demuestran el funcionamiento técnico de ‘ToyBox’ con diferentes juguetes en el espacio interactivo JUGUEMOS.

A modo de resumen, el dispositivo ‘ToyBox’ construido cumple con los siguientes aspectos:

- Permite integrar objetos electrónicos, y objetos a los que se le ha añadido un circuito electrónico, en el espacio interactivo JUGUEMOS.
- Es posible integrar, simultáneamente, un número indefinido de objetos, cada uno conectado a un ‘ToyBox’, en el espacio interactivo.
- La comunicación inalámbrica entre ‘ToyBox’ y el broadcaster se realiza a través del Protocolo OSC, protocolo que utilizan todos los elementos del espacio para transmitir información. Además, el objeto se puede desplazar por todo el espacio interactivo, sin que se produzcan interrupciones en la comunicación.
- Permite la conexión de objetos de una manera sencilla, haciendo posible que un usuario sin conocimientos de electrónica pueda integrar objetos electrónicos en el espacio interactivo.
- El prototipo diseñado es pequeño y compacto, lo que permite un manejo cómodo del objeto conectado.
- Dispone de una alimentación propia, independiente de la alimentación del objeto conectado. Esto permite el circuito del dispositivo quede aislado del circuito del objeto.

### 7.2. Trabajo futuro

A continuación se destacan posibles líneas de trabajo futuros que podrían complementar o ampliar las posibilidades interactivas del dispositivo ‘ToyBox’.

Actualmente, el dispositivo sólo permite conectar un sensor y/o un actuador, limitación dada por el módulo ESP8266. Una posible línea de mejora sería modificar el dispositivo para poder conectar varios sensores y actuadores en el mismo ‘ToyBox’.

Además, el tipo de interacción tangible del objeto en el espacio interactivo está limitado a estados activos o inactivos. Por lo que una posible línea de trabajo futuro sería ampliar el tipo de interacción tangible, de manera que permitiese conocer más datos sobre el estado del objeto conectado. Por ejemplo, en el caso de la regadera descrita en el capítulo anterior, ‘ToyBox’ reconoce si la regadera está recta o inclinada. Ampliando el tipo de interacción sería posible conocer el grado de inclinación de la regadera.

Por último, el tamaño final de ‘ToyBox’ se ve afectado por la alimentación. Se podría buscar otra alimentación alternativa que tuviese unas dimensiones menores que la elegida en este trabajo. De esta forma se reduciría el tamaño del dispositivo.



## Bibliografía

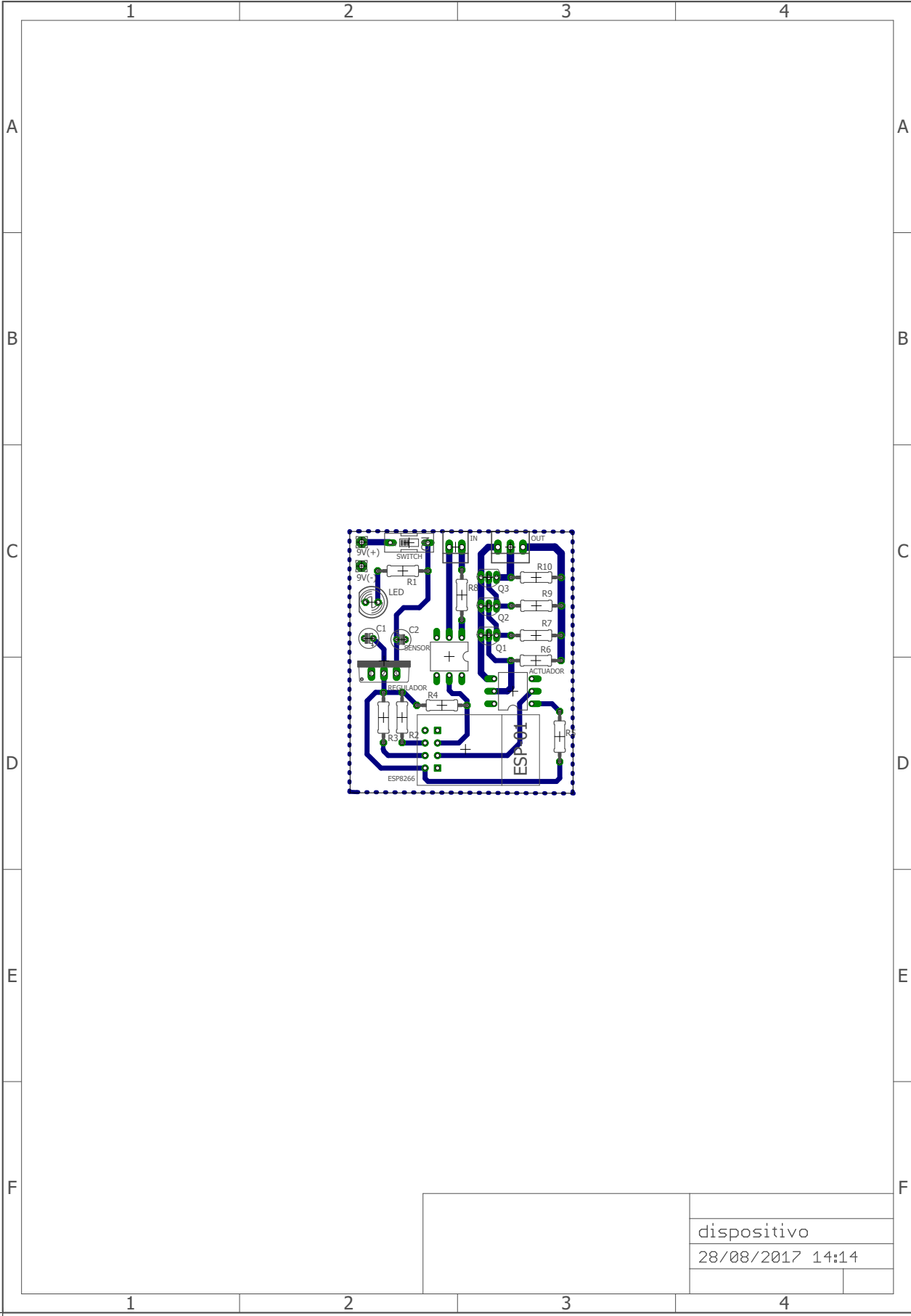
- [Ard] Arduino: <https://www.arduino.cc/>
- [BRS03] Ballagas, R., Ringel, M., Stone, M., & Borchers, J. 2003. *iStuff: a physical user interface toolkit for ubiquitous computing environments*. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 537-544). ACM.
- [DESP] Datasheet de módulo ESP8266: <http://www.microchip.ua/wireless/esp01.pdf>
- [DOpt] Datasheet del optoacoplador 4N35: <https://measure.feld.cvut.cz/groups/edu/osv/4N35.pdf>
- [DReg] Datasheet del regulador LM1117t-3.3V: <http://www.ti.com/lit/ds/symlink/lm1117.pdf>
- [DTran] Datasheet del transistor BC547: [http://pdf.datasheetcatalog.com/datasheets/150/128380\\_DS.pdf](http://pdf.datasheetcatalog.com/datasheets/150/128380_DS.pdf)
- [ESP] Módulo ESP8266: <http://www.prometec.net/modelos-esp8266/>
- [GBM13] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. 2013. *Internet of Things (IoT): A vision, architectural elements, and future directions*. Future generation computer systems, 29(7), 1645-1660.
- [GF01] Greenberg, S., & Fitchett, C. 2001. *Phidgets: easy development of physical interfaces through physical widgets*. In Proceedings of the 14th annual ACM symposium on User interface software and technology (pp. 209-218). ACM.
- [GIG17] GIGA Affective Lab: <http://giga.cps.unizar.es/affectivelab/>
- [HB06] Hornecker, E., & Buur, J. 2006. *Getting a grip on tangible interaction: a framework on physical space and social interaction*. In Proceedings of the SIGCHI conference on Human Factors in computing systems (pp. 437-446). ACM.
- [HGK04] Holmquist, L. E., Gellersen, H. W., Kortuem, G., Antifakos, S., Michahelles, F., Schiele, B., ... & Mazé, R. 2004. *Building intelligent environments with smart-its*. IEEE Computer Graphics and Applications, 24(1), 56-64.
- [Imp] Impresora 3D Zortrax M200: <https://zortrax.com/printers/zortrax-m200/>
- [KBY06] King, J., Bose, R., Yang, H. I., Pickles, S., & Helal, A. 2006. *Atlas: A service-oriented sensor platform: Hardware and middleware to enable programmable pervasive spaces*. In local computer networks, proceedings 2006 31st IEEE conference on (pp. 630-638). IEEE.
- [LOE] Librería OSC para módulo ESP8266: <https://github.com/sandeepmistry/esp8266-OSC>
- [LEA] Librería de ESP8266 para software de Arduino: <https://github.com/esp8266/Arduino/releases>

- [MCB08] Marco, J., Cerezo, E., Baldassarri, S., Mazzone, E., & Read, J. C. 2009. *Bringing tabletop technologies to kindergarten children*. In Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology (pp. 103-111). British Computer Society.
- [MSW09] Montola, M., Stenros, J., & Waern, A. 2009. *Pervasive games: theory and design*. Morgan Kaufmann Publishers Inc..
- [Nan] Arduino Nano: <https://store.arduino.cc/arduino-nano>
- [Opt] Optoacoplador: <https://es.wikipedia.org/wiki/Optoacoplador>
- [Ras] Raspberry Pi: [http://www.diotronic.com/raspberry-pi-model-3-b-sbc\\_29805/](http://www.diotronic.com/raspberry-pi-model-3-b-sbc_29805/)
- [See] Seeed Studio: <https://www.seeedstudio.com/>
- [Vil17] Villalba, Adrian. 2017. Plataforma electrónica para incluir juguetes convencionales en juegos de realidad mixta.TFG.
- [WF97] Wright, M., & Freed, A. 1997. *Open SoundControl: A New Protocol for Communicating with Sound Synthesizers*. In ICMC.

# **Anexos**

## A Esquemático y PCB





## B Presupuesto

Componente	Cantidad	Fabricante	Proveedor	Precio /Unidad (€)	Total (€)
Resistencia 10K	4	TE Connectivity	Diotronic	0,02	0,08
Resistencia 330Ω	2	TE Connectivity	Diotronic	0,02	0,04
Resistencia 470Ω	1	TE Connectivity	Diotronic	0,02	0,02
Resistencia 100Ω	1	TE Connectivity	Diotronic	0,02	0,02
Resistencia 47Ω	1	TE Connectivity	Diotronic	0,02	0,02
Resistencia 22Ω	1	TE Connectivity	Diotronic	0,02	0,02
Optoacoplador 4N35	2	Texas Instruments	AESA	0,25	0,5
Interruptor	1	C&K	Diotronic	0,13	0,13
ESP8266 (ESP-01)	1	Espressif	Prometec	4,5	4,5
Adaptador USB a FTDI UART	1	-	Prometec	3,7	3,7
Conector macho 2 pines 90º	1	Molex	Diotronic	0,1	0,1
Conector macho 3 pines 90º	1	Molex	Diotronic	0,15	0,15
Regulador LM1117t-3.3v	1	Texas Instruments	Diotronic	0,44	0,44
Transistor BC547C	3	-	AESA	0,02	0,06
LED verde 5mm	1	-	Diotronic	0,11	0,11
Portapilas	1	-	Diotronic	0,79	0,79
Batería 9V	1	Duracell	Bauhaus	4,5	4,5
Condensador 10μF	2	Huang	Diotronic	0,04	0,08
Tira de contactos hembra	2	Winslow	AESA	2,12	4,24
PCB	1	Seeed Studio	Seeed Studio	0,49*	0,49
Coste total componentes prototipo					19,99

\* Precio de producción (gastos de envío no incluidos). Precio por unidad en paquete de 10 unidades. A partir de 10 unidades, el precio por unidad varía.

## C Cronograma de planificación temporal

Id.	Nombre de tarea	Inicio	Fin	Duración	abr.2017				may.2017					jun.2017				jul.2017				ago.2017					
					2/4	9/4	16/4	23/4	30/4	7/5	14/5	21/5	28/5	4/6	11/6	18/6	25/6	2/7	9/7	16/7	23/7	30/7	6/8	13/8	20/8	27/8	
1	Estudio de tecnologías	03/04/2017	29/04/2017	4s																							
2	Desarrollo del software	01/05/2017	16/06/2017	7s																							
3	Desarrollo de sistema electrónico	15/05/2017	16/06/2017	5s																							
4	Montaje en protoboard	14/06/2017	23/06/2017	1s																							
5	Diseño de PCB	26/06/2017	12/07/2017	3s																							
6	Fabricación de prototipo	24/07/2017	08/08/2017	3s																							
7	Evaluación del dispositivo	09/08/2017	25/08/2017	2s																							
8	Redacción del proyecto	02/06/2017	31/08/2017	13s																							



## D Código

```
/* Código del módulo ESP8266
 * Se realiza la conexión con el Broadcaster de la Plataforma JUGUEMOS, se envían los datos recibidos por
 * GPIO2 y se reciben las órdenes, dando valor al GPIO0
 */

#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <OSCMessages.h>
#include <OSCDData.h>

char ssid[] = "nikvision";    // network SSID (name)
char pass[] = "";            // network password

// A UDP instance to let us send and receive packets over UDP
WiFiUDP Udp;

const IPAddress outIp(192, 168, 0, 2); // remote IP
const unsigned int outPort = 32000; // remote port
const unsigned int localPort = 12001; // local port to listen for UDP packets
OSCErrCode error;
unsigned int cont = 0;
int sensor = -1;
int sensorAnt = -1;
float valor = 0;

////////////////////////////////////
int id = 0; // Identificador de ToyBox para actuador
char * idG = "0"; // Identificador de ToyBox para sensor
////////////////////////////////////

void setup() {
  pinMode(2, INPUT);
  pinMode(0, OUTPUT);

  digitalWrite(0,HIGH);

  // Connect to WiFi network
  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  Udp.begin(localPort);
  connect();
}

void connect() {
  OSCMessage msg("/sensor/connect");
  msg.add(localPort);
  Udp.beginPacket(outIp, outPort);
  msg.send(Udp);
  Udp.endPacket();
  msg.empty();
}
```

```
void disconnect() {
    OSCMessage msg("/sensor/disconnect");
    msg.add(localPort);
    Udp.beginPacket(outIp, outPort);
    msg.send(Udp);
    Udp.endPacket();
    msg.empty();
}

void mensajeRecibido(OSCMessage &msg, int offset) {
    if(msg.getInt(0) == id) {
        if(msg.getFloat(2) == 0.0)
            digitalWrite(0,HIGH);
        else
            digitalWrite(0,LOW);
    }
}

void loop() {
    int size = Udp.parsePacket();
    OSCMessage msg;

    if (size > 0) {
        while (size-- > 0) {
            msg.fill(Udp.read());
        }
        if (!msg.hasError()) {
            msg.route("/actuator/0D/toybox", mensajeRecibido);
        } else {
            error = msg.getError();
        }
    }

    sensor = digitalRead(2);
    if(sensor != sensorAnt) {
        OSCMessage mensaje("/sensor/0D/toybox");
        mensaje.add("add");
        mensaje.add(idG);
        mensaje.add("0"); // idS -> valor por defecto -> 0
        if(sensor == 0) //Cuando se pulsa llega cero, pero hay que mandar 1 para indicar boton pulsado
            valor = 1;
        else
            valor = 0;

        mensaje.add(valor);
        Udp.beginPacket(outIp, outPort);
        mensaje.send(Udp);
        Udp.endPacket();
        mensaje.empty();
    }
    sensorAnt = sensor;
    delay(1000);
}
```