Universidad de Alcalá
COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO

ACTA DE EVALUACIÓN DE LA TESIS DOCTORAL

(FOR EVALUATION OF THE ACT DOCTORAL THESIS)

Año académico (*academic year):* 2018/19

DOCTORANDO *(candidate PHD)*: **RIVERA PINTO, DIEGO**
D.N.I ./ PASAPORTE (*Id.Passport*): ******190S**
PROGRAMA DE DOCTORADO *(Academic Committee of the Programme)*: **D445-TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES**
DPTO. COORDINADOR DEL PROGRAMA *(Department)*: **TEORÍA DE LA SEÑAL Y COMUNICACIONES**
TITULACIÓN DE DOCTOR EN *(Phd title)*: **DOCTOR/A POR LA UNIVERSIDAD DE ALCALÁ**

En el día de hoy 14/01/19, reunido el tribunal de evaluación, constituido por los miembros que suscriben el presente Acta, el aspirante defendió su Tesis Doctoral **con Mención Internacional** (*In today assessment met the court, consisting of the members who signed this Act, the candidate defended his doctoral thesis with mention as International Doctorate*), elaborada bajo la dirección de (*prepared under the direction of*) BERNARDO ALARCOS ALCÁZAR // SUSEL FERNÁNDEZ MELIÁN.

Sobre el siguiente tema *(Title of the doctoral thesis)*: *CONTRIBUTIONS TO THE DATA ADQUISITION TECHNOLOGIES AND THE METHODS FOR AUTOMATIC DETECTION AND CLASSIFICATION OF ACTIVITIES IN A SMART TOYS ENVIRONMENT*

Finalizada la defensa y discusión de la tesis, el tribunal acordó otorgar la CALIFICACIÓN GLOBAL[1] de (**no apto, aprobado, notable** y **sobresaliente**) *(After the defense and defense of the thesis, the court agreed to grant the GLOBAL RATING (fail, pass, good and excellent)*: SOBRESALIENTE

Alcalá de Henares, a 14 de Enero de 2019

Fdo. *(Signed)*: JOSE F MARTINEZ Fdo. *(Signed)*: Iván María Mateos Fdo. *(Signed)*: MARIO [illegible] BARCAS

FIRMA DEL ALUMNO (candidate's signature),

Fdo. *(Signed)*: DIEGO RIVERA PINTO

Con fecha 21 de enero de 2019 la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado, a la vista de los votos emitidos de manera anónima por el tribunal que ha juzgado la tesis, resuelve:

☒ Conceder la Mención de "Cum Laude"
☐ No conceder la Mención de "Cum Laude"

La Secretaria de la Comisión Delegada

[1] La calificación podrá ser "no apto" "aprobado" "notable" y "sobresaliente". El tribunal podrá otorgar la mención de "cum laude" si la calificación global es de sobresaliente y se emite en tal sentido el voto secreto positivo por unanimidad. (*The grade may be "**fail**" "**pass**" "**good**" or "**excellent**". The panel may confer the distinction of "cum laude" if the overall grade is "Excellent" and has been awarded unanimously as such after secret voting.).*

INCIDENCIAS / OBSERVACIONES:
(Incidents / Comments)

Universidad de Alcalá

COMISIÓN DE ESTUDIOS OFICIALES
DE POSGRADO Y DOCTORADO

En aplicación del art. 14.7 del RD. 99/2011 y el art. 14 del Reglamento de Elaboración, Autorización y Defensa de la Tesis Doctoral, la Comisión Delegada de la Comisión de Estudios Oficiales de Posgrado y Doctorado, en sesión pública de fecha 21 de enero, procedió al escrutinio de los votos emitidos por los miembros del tribunal de la tesis defendida por *RIVERA PINTO, DIEGO*, el día 14 de enero de 2019, titulada *CONTRIBUTIONS TO THE DATA ADQUISITION TECHNOLOGIES AND THE METHODS FOR AUTOMATIC DETECTION AND CLASSIFICATION OF ACTIVITIES IN A SMART TOYS ENVIRONMENT*, para determinar, si a la misma, se le concede la mención "cum laude", arrojando como resultado el voto favorable de todos los miembros del tribunal.

Por lo tanto, la Comisión de Estudios Oficiales de Posgrado **resuelve otorgar** a dicha tesis la

MENCIÓN "CUM LAUDE"

Alcalá de Henares, 22 de enero de 2019
EL VICERRECTOR DE INVESTIGACIÓN Y TRANSFERENCIA

UNIVERSIDAD DE ALCALÁ · VICERRECTORADO DE INVESTIGACIÓN Y TRANSFERENCIA ·

F. Javier de la Mata de la Mata

Copia por e-mail a:
Doctorando: RIVERA PINTO, DIEGO
Secretario del Tribunal: IVAN MARSÁ MAESTRE
Directores de Tesis: BERNARDO ALARCOS ALCÁZAR // SUSEL FERNÁNDEZ MELIÁN

Universidad de Alcalá

ESCUELA DE DOCTORADO
Servicio de Estudios Oficiales de Posgrado

DILIGENCIA DE DEPÓSITO DE TESIS.

Comprobado que el expediente académico de D./Dª Diego Rivera Pinto reúne los requisitos exigidos para la presentación de la Tesis, de acuerdo a la normativa vigente, y habiendo presentado la misma en formato: ☒ soporte electrónico ☐ impreso en papel, para el depósito de la misma, en el Servicio de Estudios Oficiales de Posgrado, con el nº de páginas: 236 se procede, con fecha de hoy a registrar el depósito de la tesis.

Alcalá de Henares a 24 de septiembre de 2018

Purificación Reviejo

Fdo. El Funcionario

Contributions to the data acquisition technologies and the methods for automatic detection and classification of activities in a Smart Toys environment

Diego Rivera Pinto

Universidad de Alcalá

Universidad de Alcalá

Campus Universitario
Dpto. de Teoría de la Señal y Comunicaciones
Ctra. Madrid-Barcelona, Km. 36,6
28805 Alcalá de Henares (Madrid)
Telf: +34 91 885 88 99
Fax: +34 91 885 66 99

D. Diego Rivera Pinto ha realizado en el Departamento de Automática y bajo la dirección de Dr. D. Bernardo Alarcos Alcázar, Dra. Dña. Susel Fernández Melián y la tutoría de Dr. D. Juan Ramón Velasco Pérez, la tesis doctoral titulada "**Contributions to the data acquisition technologies and the methods for automatic detection and classification of activities in a Smart Toys environment**", cumpliéndose todos los requisitos para la tramitación que conduce a su posterior lectura.

Alcalá de Henares, 5 de septiembre de 2018.

Coordinador del Programa de Doctorado

UNIVERSIDAD DE ALCALÁ * DPTO. DE TEORÍA DE LA SEÑAL Y COMUNICACIONES *

Fdo.: Dr. D. Sancho Salcedo Sanz

Universidad de Alcalá

Campus Universitario
Dpto. de Automática
Ctra. Madrid-Barcelona, Km. 36,6
28805 Alcalá de Henares (Madrid)
Telf: +34 91 885 65 94
Fax: +34 91 885 69 23

Dr. D. BERNARDO ALARCOS ALCÁZAR, Titular de Universidad del Área de Conocimiento de Telemática de la Universidad de Alcalá, Dra. Dña. SUSEL FERNÁNDEZ MELIÁN, Profesor Ayudente Doctor del Área de Conocimiento de Telemática de la Universidad de Alcalá, y Dr. D. JUAN RAMÓN VELASCO PÉREZ, Catedrático de Universidad del Área de Conocimiento de Telemática de la Universidad de Alcalá,

CERTIFICAN

Que la tesis "**Contributions to the data acquisition technologies and the methods for automatic detection and classification of activities in a Smart Toys environment**", presentada por D. Diego Rivera Pinto, realizada en el Departamento de Automática bajo nuestra dirección, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y lectura.

Alcalá de Henares, 5 de septiembre de 2018.

Fdo.: Dr. D. Bernardo Alarcos Alcázar

Fdo.: Dra. Dña. Susel Fernández Melián

Fdo.: D. Juan Ramón Velasco Pérez

UNIVERSIDAD DE ALCALÁ
DEPARTAMENTO DE AUTOMÁTICA

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE AUTOMÁTICA

Programa de doctorado en Tecnologías
de la Información y las Comunicaciones

TESIS DOCTORAL

"Contributions to the data acquisition technologies and the methods for automatic detection and classification of activities in a Smart Toys environment"

Diego Rivera Pinto

2018

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE AUTOMÁTICA

Programa de doctorado en Tecnologías de la Información y las Comunicaciones

"Contributions to the data acquisition technologies and the methods for automatic detection and classification of activities in a Smart Toys environment"

Autor

Diego Rivera Pinto

Directores

Bernardo Alárcos Alcázar, Susel Fernández Melián

Tutor

Juan Ramón Velasco Pérez

2018

TESIS DOCTORAL

A mi familia

Agradecimientos

Me gustaría agradecer a mis directores de Tesis y a mi tutor por el apoyo recibido a lo largo de la realización de este trabajo de investigación. También quiero agradecer a todos los investigadores integrantes del proyecto EDUCERE, tanto de la Universidad de Alcalá como del centro universitario Cardenal Cisneros, la Universidad Autónoma de Madrid, la Universidad Politécnica de Madrid y el CEAPAT, ya que su dedicación y su colaboración en los distintos ámbitos de la investigación multidisciplinar ha sido fundamental para la consecución de la Tesis. Por otro lado, me gustaría también agradecer a las escuelas "Cuentos de Colores", "Gaia" y "Barbel Inhelder" y a sus respectivos responsables, por permitirnos llevar a cabo las pruebas piloto, y a todos los participantes en ella, tanto profesores como padres y niños.

Doy las gracias también a las personas con las que tuve la oportunidad de colaborar durante mi estancia en la Universidad Campus Biomédico de Roma por su hospitalidad y apoyo. Especialmente doy las gracias a Fabrizio Taffoni, por su ayuda y la oportunidad de aprender de su trabajo.

Un agradecimiento especial para mis compañeros en la Universidad de Alcalá, que me han apoyado y animado a lo largo de estos años, y especialmente a Luis de la Cruz, por su implicación y su colaboración desinteresada. También quiero agradecer a mi familia, especialmente a mis padres y a mi hermana el apoyo que me han brindado durante estos años, y a todos mis amigos, que han estado siempre ahí todo este tiempo, animándome siempre a continuar.

Finalmente, me gustaría agradecer la financiación ofrecida por el Ministerio de Economía y Competitividad para el desarrollo de este trabajo a través de las ayudas predoctorales para la formación de doctores (BES-2014-067912).

Resumen

El desarrollo de nuevos servicios y tecnologías basadas en Internet y asociadas a sensores ha permitido la aparición de nuevas oportunidades para el desarrollo de herramientas de adquisición y análisis de datos aplicadas a diversos campos. La posibilidad de integrar sensores de menor tamaño y consumo eléctrico hace que estos sistemas puedan ser utilizados en escenarios donde anteriormente no era posible. Uno de estos escenarios es el relacionado con la salud, donde se pueden obtener importantes beneficios en la calidad de vida de los usuarios. Como contraprestación, estos sistemas suponen la aparición de nuevos retos debidos a los requisitos de seguridad y privacidad de los datos tratados en ellos.

La evaluación del desarrollo infantil a través de la realización de juegos u otras actividades similares es un campo ampliamente estudiado. Los expertos en este tipo de evaluación (psicólogos, fisioterapeutas, pedagogos, etc.) han creado y mejorado a lo largo del tiempo escalas reguladas que permiten detectar, bajo determinadas condiciones, posibles dificultades en el desarrollo infantil desde edades tempranas. La detección temprana de estas dificultades es fundamental para que su seguimiento o los posibles tratamientos sean lo más efectivos posible. Actualmente, las actividades de evaluación del desarrollo requieren la recopilación de datos de forma manual y presencial durante la duración del juego, lo cual dificulta el análisis y la obtención de conclusiones por parte de los expertos.

La investigación llevada a cabo durante el desarrollo de esta Tesis se ha centrado en el estudio de las posibilidades de ofrecer soluciones tecnológicas para mejorar las actividades de juego utilizadas en la detección temprana de dificultades de desarrollo infantil. Para ello, se han abordado tanto cuestiones generales como la creación de una plataforma específica de Smart Toys, como cuestiones más específicas como el desarrollo de metodologías para el análisis autónomo y automático de cada movimiento

que compone la actividad. La plataforma se ha desarrollado siguiendo el paradigma del Internet de las Cosas (IoT) con el objetivo de integrar todos los actores que puedan intervenir en la evaluación (desde el diseño de juguetes con sensores incorporados a la definición de las comunicaciones entre dispositivos y los sistemas de seguridad y privacidad adecuados). El uso de estas tecnologías y métodos puede ofrecer una mejora significativa en cuanto a la precisión en las métricas de evaluación del desarrollo y abre la posibilidad de obtener nuevos parámetros de medida que no pueden recogerse de forma manual.

Se han definido tres objetivos específicos en esta Tesis: El primero de ellos es el diseño de la plataforma basada en una arquitectura IoT, así como los Smart Toys que la componen. Para ello se han estudiado las particularidades de los Smart Toys y se han propuesto soluciones específicas para los retos que presenta su integración en un entorno IoT. El segundo objetivo es el diseño y desarrollo de mecanismos de seguridad y privacidad personalizados para ofrecer la protección necesaria de los datos sensibles durante el proceso de intercambio a través de la plataforma. Finalmente, el tercer objetivo se centra en la creación y desarrollo de metodologías para la detección y clasificación de los movimientos realizados con los Smart Toys, que permitan obtener información valiosa que pueda ser utilizada por los expertos en desarrollo infantil.

Palabras clave: Internet de las Cosas, Juguetes inteligentes, Seguridad, Privacidad, Control de acceso, Clasificación de actividades, Reconocimiento de patrones.

Abstract

The development of novel Internet services and sensor technologies has allowed the appearance of new opportunities for the development of tools for the acquisition and analisys of data in various fields. The possibility of integrating smaller and with lower energy consompution sensors makes possible to use these sensor-based systems in scenarios where it was formerly not possible. One of these scenarios is the healthcare related systems, where important gains in terms of users' quality of live can be obtained. As a drawback, these systems raise new challenges related with the security and privacy of the data managed in them.

The assessment of children development through games or other similar playing activities is a well-studied field. Through time, experts in this type of evaluation (psychologists, physiotherapists, pedagogues, etc.) have created and improved regulated scales which allow them to detect, under certain circumstances, possible development difficulties in children from early ages. The early detection of these difficulties is crucial for its monitoring and possible treatment to be as effective as possible. Nowadays, the activities used for development assessment require the manual and in-person data compilation in the course of the game, which makes more difficult the analysis and obtaining conclusions from experts.

The research carried out during the development of this Thesis has been focused on the study of possibilities for offering technological solutions to improve the playing activities used in the early detection of children development difficulties. To achive this goal, we have addressed both general issues such as the creation of a specific Smart Toys platform, and more specific issues such as the development of methodologies for the autonmous and automatic analysis of each movement composing an activity. The platform has been developed following the Internet of Things (IoT) paradigm, aiming to integrate in it every actor involved in the assessment (from the design of

sensor-embedded toys to the definition of the communications between devices, and the appropriate security and privacy mechanisms). The use of these technologies and methods can provide a significative improvement in terms of development assessment metrics measurement accuracy and can also opens the possibility of obtaining new measurement parameters which could not be obtained manually.

Three specific objectives have been defined in this Thesis: The first one is the design of the platform based on an IoT architecture, and the Smart Toys composing it. To achieve this objective, we have studied the distinctive features of Smart Toys and we have proposed specific solutions for the challenges raised by their integration in an IoT enviornment. The second objective is the design and development of custom security and privacy mechanisms which offer the neccesary protection during the sensible data exchange in the platform. Finally, the third objective is focused on the creation and development of methodologies for the detection and classification of movements performed with the Smart Toys, in order to obtain valuable information to be used by the children development experts.

Keywords: Internet of Things, Smart Toys, Security, Privacy, Access Control, Activities classification, Pattern recognition.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introduction

In this first chapter, we introduce the research work by providing its motivation and the scientific background in which the contributions of the Thesis are based (section 1.1). Then we detail the main general goal and specific objectives which were defined and then followed during the Thesis development (section 1.2). The main contributions that have allowed us to reach the objectives are summarized in section 1.3, and they are completely described in the next three chapters of this book. Finally, we include an overview of the book organization and its contents in section 1.4.

1.1. Motivation and background

Playing is one of the most important activities for children while they are growing up. Games not only are used as a tool for entertainment, but they play a crucial role in children learning and development processes [1].

The role of games as support for the learning processes has been widely studied [2]. There are many projects and research proposals which focus on using games and toys to help in the educational processes in many different areas [3].

Almost as studied as the educational component of games and toys is the relationship between playing actions and the child development in many different areas,

such as the cognitive, language, motor or sensory capabilities [4, 5]. It has been proven that, observing the way children play can provide useful information about their developmental status, and therefore, it can be used as a tool for the assessment of possible developmental delays or difficulties which could arose in any of the previously cited areas. This relationship relies on the fact that the development is materialized in diverse physiological and behavioral functions which can be observed during the child growth.

The activities used to identify this relationship are very diverse, and there are different approaches to test the performance in the literature. For instance, in [6], there is a review of sets of activities devised for the assessment of motor development. The actual activities can go from positioning the child in certain ways to measure the child reflex (used specially for young children from 0 to 10 months old) to more complex activities such as playing with a ball (kicking, catching or throwing it to a target for instance), manipulating specific toys such a rattle, or performing activities with sets of pegs or cubes (combining them, building structures, inserting shapes, etc.). Books like [7] define an extense and complete set of activities and the guidelines to use them to assess the children motor development.

An early detection of these difficulties is decisive to guarantee a good quality of life for children, as their future development will be strongly influenced by how these problems are issued and treated [8]. Moreover, the early development has an enormous influence in the future health of the children, in terms of postural control, verbal communication or social interactions, among many other aspects [9, 10]. For that reason, the early detection and treatment of any delay encountered is crucial. In general, the earlier the detection, the more is guaranteed the avoidance or prevention of additional pathologies and the better quality of life will be achieved [11].

Over the years, health specialists such as psychologists, physiotherapists, pediatricians, etc., have been able to take advantage of this relationship to develop tools which, when combined with other tests and diagnosis tools, would help them to detect and diagnose many difficulties at an early stage. There has been an effort to provide standard tools and guidelines to obtain regulated results in the tests carried out. Taxonomies about games used in health issues can be found in the literature. For instance, in [12], games are classified according their goal as self-ranking games, games for

rehabilitation, games for disease management, data collection games, etc. Moreover, regulated scales have been determined to help other experts in the evaluation of children through play. These scales work as guidelines for the experts, and provide both the activities and games that should be carried out, the description of the physical tools used in them (in some cases, toys can be purchased in combination with the scale), and of course, the description of the evaluation to perform whilst observing the activities. Examples of these scales are the previosly referenced Peabody scale [7] and also the Bayley [13] and Merrill-Pallmer [14] scales.

Despite the huge effort in early detecting development issues, there are still many cases in which is not possible to correctly identify the problems, and therefore they go unnoticed until later in the child life. This means that these children cannot receive the correct treatments or cares, causing a negative impact in their lives, as is stated in the "World Report on Disability" from the World Health Organization (WHO) [15].

More recently, studies, projects and proposals have focused on improving the processes used for the assessment of children development by using information systems [16] and new technologies [17]. Until recently, most of the evaluation has been performed traditionally using "manual" methods: Usually the playing activities are supervised by an expert which uses a visual examination of the performance (occasionally aided by a chronometer or other extra tool). Then performance is assessed by evaluating a series of metrics such as the time used to complete the game, the doubt when making movements, the way objects are grabbed, etc. [14]. New technologies can improve these methods by automatizing the information gathering process, by obtaining higher accuracy in the measurements, and by providing new information which cannot be retrieved manually, which could also lead to new evaluation processes and a improve the current difficulties' detection rates.

On the other hand, in recent years there has been an important improvement in the sensor-related technologies, which has leaded to the development of cheaper, smaller and more accurate sensors [18]. This progress in sensor technology, in combination with the growth of Internet has leaded to the definition of new paradigms like the "Internet of Things" (IoT) [19]. This concept, has been defined as the next evolution of what we call Internet nowadays. It proposes the massive interconnection of everyday objects which would be equipped with sensors and actuators to interact with the physical environment

where they are located. This would lead to new applications and services which could take advantage of the huge amount of data generated by billions of interconnected devices.

Currently, the construction of the "Internet of Things" is a work in progress, where there are many research and commercial proposals of platforms, devices and services related with it. For instance, the popularity of Wearable devices (microcontroller-driven electronic connected devices which can be put into clothes or worn like Smart Watches, activity tracking bands, etc.) can be seen as an important example of how the IoT systems are becoming more and more popular.

The possibilities of using the IoT paradigm as enabler technology for the design of systems which could contribute to healthcare and the general well-being of people have been studied [20], and it has been identified as one of the most promising contributions in the next years to improve the healthcare systems [21]. Using a IoT approach to design and build healthcare systems provide many advantages, among which it is possible to highlight that it would allow to offer users with more trustworthy and less intrusive interfaces that many proposed systems which rely on the complex installation of sensors in the patient's body or in the environment [22]. Even more important is the possibility of obtaining data from a high number of devices, which would lead to the definition of services for the analysis and aggregation of the data, obtaining new valuable information about the symptoms, treatments, patient evolution, etc. [23].

Nevertheless, using an IoT approach for healthcare raises important challenges in different aspects of their design and development. For instance, the interconnection of devices implies the exchange of a huge amount of data which could give away many personal information about the users, including personal data, their location, or even highly confidential health-related data. Moreover, the heterogeneity of the data, the devices and the interconnection networks make more difficult the protection of privacy and the systems access control.

Although this is a crucial issue in IoT systems, there is not yet a standard approach to provide security in terms of confidentiality, authentication, access control and other security-related tasks. The many different architectures and technologies available make it difficult to provide a generic security solution.

Another important question that must be issued in "Internet of Things" systems is how the data from sensors becomes valuable information for the users. The data from sensors is rarely useful as it is measured, and therefore, it needs some processing to generate useful information which can be consumed by users. The processing of the information depends on the type of data retrieved and the kind of information it should provide. Depending on the goal of the system, this processing can be as simple as calculating the average values (i.e. getting the average acceleration during a race) or as complex as performing operations which sometimes require the use of Artificial Intelligence techniques (i.e. automatically classify the different activities performed with a device).

Nowadays, it is possible to find an enormous variety of IoT platforms with the most heterogeneous purposes, from the construction of smart spaces to the logistic, social or health related goals. This heterogeneity favors the disparity of platforms and architectures, that often are not easily interoperable [24].

The use of an "Internet of Things" approach for improving the methods and techniques for the evaluation of children development through playing activities is a logical step in research, taking into account that IoT has been identified as a highly beneficial technological paradigm to be used in health related problems, and also that there exist tools for the assessment of children development that need a technological upgrade.

There are already works in the literature which are focused in providing Smart Toys for healthcare [25] or other purposes such as education or entertainment [26, 27]. Smart Toys are usually defined as any playing device with sensors and/or actuators embedded and any communications capabilities [28]. The term was popularized by the "Smart Toy Lab", a work group created in 1998 by the companies Mattel and Intel [29] to design toys which would take advantage of new technologies to be more innovative and attractive. Since then, many works have taken the approach of providing toys with sensing, communications, and "intelligence" capabilities. The Smart Toys can be seen as novel data acquiring devices which also provide new interactivity and feedback possibilites to enrich the playing act.

The combination between the concepts of Smart Toys and the IoT has led to coin a new term, the "Internet of Toys" (IoToys) [30], which would refer to any platform which

uses the IoT processes and protocols to interconnect sets of Smart Toys. The goals of this "Internet of Toys" are also heterogeneous, but it presents a series of common features, challenges and concerns in their design and development.

The "Internet of Toys" offers a very interesting opportunity for combining the potential of Smart Toys as novel data acquiring technologies and the possiblities of the distributed gathering of information for data analysis and processing which offer the IoT paradigm.

The work carried out in this Thesis has contributed to the IoToys paradigm in the definition of a specific Smart Toys platform which is devised for the acquisition of valuable data for the children development assessement. Specifically, and given that the available heterogeneous platforms do not cover the requisites of this particular project, the platform has been defined according to a reference architectonic model, and then a set of prototype Smart Toys have been built to work in it. Given the concerns related with the privacy and security of these technologies, a special effort has been made to define mechanisms to avoid the threats and risks which could arise from its use, protecting the generated data. Then, to study the extraction of valuable data from the built devices, we have defined and developed methodologies and mechanisms to automatically detect and classify the different movements performed while using the Smart Toys.

The results of this Thesis are expected to be useful as a novel set of tools (including the platform, the devices and the methodologies) for children development experts in their assessment tasks, while potentially determine new evaluation methods which could be derived from the information obtained from sensors.

The work is framed in the research project EDUCERE [31], which has been conceived as a multidisciplinary project to provide innovative technological solutions to children development experts which could lead to better and earlier developmental difficulties detections.

1.2. Objectives of the Thesis

The main goal of this Thesis has been the creation of new methodologies and processes in a Smart Toys enviornment for the generation of valuable information for experts in children development. To achieve this, the research has relied in the current tools and toys used by these specialists for the assessment of delays or difficulties in motor, cognitive, language or sensory development.

To achieve the goal, a set of three specific objectives have been identified. These objectives have determined the main tasks to perform during the Thesis work period, and the main contributions of the work. The objectives are:

1. The design, development and validation of a "Smart toys platform" for the collection and classification of data from Smart Toys which is based on the IoT paradigm.
2. The design, development and validation of custom security and privacy mechanisms for the Smart Toy platform in order to allow the secure and controlled communications of data.
3. The design, development and validation of methodologies and algorithms to automatically analyze and extract the data from the playing activities, generating valuable information for the experts.

The first objective has been defined to be able to count on a design framework and actual devices which could serve as support for the development of data acquisition methods, and to test the possible information to acquire when processing the sensor-based data. The use of an IoT approach is logical as the state of the art suggests, because it allows the development of an ecosystem of interconnected devices using standardized processes.

The study of the technological background has leaded us to define the second objective as a separate part of the research work. The privacy and security of information systems are one of the most important issues to address in IoT systems nowadays. Given the specific nature of the platform, the work has been specially focused on providing

confidentiality and access control to the data of the platform.

The next step after the definition of the platform is the definition of methodologies and mechanisms to provide actual useful information to the experts. To carry out this task, we have determined that one of the most important goals is to provide a method for the automatic classification of movements composing an activity performed using the Smart Toys. This allow the further analysis of each movement and improves the accuracy of the measurements in the metrics used by the experts, while providing possible new information to help in the assesssment process.

1.3. Contributions

The work carried out in the Thesis has been focused on fulfilling the objectives defined in section 1.2, and therefore, three main contributions have been proposed, each one trying to reach a specific objective. The main contributions of the Thesis are:

1. The design of a specific IoT platform for Smart Toys, following the guidelines of a well-known reference model and offering specific solutions for the application of the IoT technologies to the scope of this study. This allows the platform to be as generic as possible, which favors the interoperability and the possibility of adding new future designed Smart Toys to it.

   The platform is based on four main physical entities: The Smart Toys, gateways or data collectors, client applications or interfaces and Internet-based servers. Prototypes of each entity have been designed and implemented, focusing on designing and developing new prototype Smart Toys based on the tools used by experts in children development. At least four different toys have been conceptually designed: A set of stackable cubes, a pegboard, a ball and a rattle, and prototypes of the cubes and the pegboard have been built. The Smart Toys have been validated in pilot tests, which have offered results about the information which could be useful to extract from the sensors.

2. Two main custom methods have been defined to provide security to the platform. As it was stated in the definition of the objectives, the main concerns regarding

security are the privacy and control of the data generated in the platform.

To provide such capabilities to the platform, we have defined a novel symmetric key confidentiality mechanism based on standard encryption algorithms and taking advantage of the radio-frequency communications used by the Smart Toys. Then, we have proposed a unified access control mechanism based for the whole platform, which would see any communication (regardless of the communication scheme used in each case) as a resource to protect. Once the communication channels are modelled as resources, it is possible to apply access control profiles designed originally for Web-only environments to the IoT platform. We have built and tested the methods defined, focusing on the overhead that could imply its use in terms of time and energy consumption.

3. A novel methodology for the automatic detection and classification of movements performed with the Smart Toys has been proposed and tested, in order to reach the third main objective of the Thesis. The most common Artificial Intelligence approaches to this task (based on Machine Learning methods) have been proven to be not accurate enough to be used for our objective, and therefore, a different approach has been devised.

   Our proposal to achieve this objectives is using the acceleration signals of the sensors to obtain acceleration trends patterns for each different possible movement, and then compare the patterns with the acceleration sequences obtained from play activities with the Smart Toys. The comparison between this method and other popular methods for movement classification has stated that our proposal is more accurate in terms of number of correct movements detected and in terms of time and duration.

   Additionally, an optimization mechanism based on a Genetic Algorithm has been defined to obtain the optimum input variables to use in the classification mechanism. This is needed because each movement requires a set of specific variables, and obtaining the optimum values is not a trivial task. This mechanism guarantees that the method can be applied to any activity, by adapting the classifier to each different movement.

1.4. Thesis overview

This book has been structured following the three specific objectives defined in section 1.2 to define each one of the three contributions from section 1.3. Each chapter includes a separated state of the art section for its especific topic. This allow us to provide a more insightful view of each topic closer to the Thesis contributions. The chapters of the book are:

- **Chapter 1: Introduction**: This chapter provides a motivation and background of this work, and presents the objectives and contributions defined for the Thesis.
- **Chapter 2: Definition of a Smart Toy IoT platform**: Includes a study of related works focused in the design of IoT architectures and platforms, and in the different proposals found about Smart Toys in the literature. Then describes the design of the architecture of our platform in each one of the design views used for its definition. The second part of the chapter is dedicated to describe the prototypes built and the technologies used in their construction. A section is reserved to show the preprocessing possibilities on the prototype Smart Toys. Finally, it presents the pilot studies carried out with the prototypes and the experimental results.
- **Chapter 3: Security and privacy in a Smart Toy environment**: This chapter includes all the work carried out related with the security and privacy of the platform. It includes a state of the art section in which the concerns about privacy in IoT and specifically in Smart Toys are stated. It also covers the possibilities in the literature to provide access control to this kind of systems. Then, we study the main threats which could arise in the platform, and provide the mechanisms to avoid them, specifically describing the methods for confidentiality and authentication of the Smart Toys communications, and the proposal for the unified access control scheme, which can be used in all the communications in the platform. Finally, the implementations and tests carried out in each case are described, along the obtained results.
- **Chapter 4: Automatic detection movements with Smart Toys**: The state of the art of this chapter classifies the activities which usually are classified using

automatic methods, and describes the most used Machine Learning methods for this task. Then we include the description of the methods for acceleration pattern extraction from the input signals, the optimization of variables using a Genetic Algorithm and the method applied for the classification of movements. The proposal has been compared with other two commonly used detection and classification methods (signal similarity search and support vector machines), and we include the results of this comparison.

- **Chapter 5: Related publications**: This brief chapter presents a compilation of the scientific publications which have been generated from the work carried out in the Thesis, including papers in indexed journals, communications in international conferences, a book chapter and two patent applications.

- **Chapter 6: Discussion, conclusions and future work**: In the final chapter of this book we include a section for discussing the research results obtained in each one of the contributions, and provide conclusions about the work carried out. Finally, we define the main lines of future work risen from the the Thesis.

Capítulo 2

Definición de una plataforma IoT para Smart Toys

En este capítulo se muestra el diseño arquitectónico llevado a cabo para obtener la plataforma de Smart Toys que se ha diseñado, así como los diseños de prototipos de las entidades de la misma. A lo largo del capítulo se presenta el estado del arte relacionado con el diseño de la plataforma y los dispositivos Smart Toy (sección 2.2), el diseño de la arquitectura para la plataforma, así como de los módulos que la componen (sección 2.3. El caso de uso y el desarrollo de los dispositivos prototipo se verá en la sección 2.4, donde también se detallarán los experimentos realizados en las pruebas piloto y sus resultados. Finalmente, se incluye un resumen de lo expuesto en el capítulo en la sección 2.5.

2.1. Introducción

En este trabajo de investigación se ha tomado como uno de los objetivos principales la creación de una plataforma de Smart Toys que ofrezca a los expertos en desarrollo infantil una serie de herramientas adicionales basadas en la tecnología para la detección de posibles problemas de desarrollo de forma temprana.

Para poder contar con las herramientas necesarias para hacer frente a este objetivo, es necesario primero diseñarlas y construirlas, lo cual a su vez implica llevar a cabo primero el diseño de esta plataforma donde se puedan integrar los dispositivos, servicios y actividades que se desean desarrollar.

A la hora de desarrollar estas herramientas de evaluación, es necesario tener en cuenta su objetivo final y los tipos de usuario que van a utilizarla, es decir, tanto niños como otras personas que no necesariamente contarán con formación específica en el uso de tecnología. Esto implica que existen una serie de requisitos específicos en cuanto a la sencillez, seguridad, fiabilidad, etc.

Para que esta plataforma y los dispositivos que la componen cumplan con esos requisitos y otros relativos a su propia construcción (por ejemplo, cómo se modelan desde el punto de vista de la autonomía, las comunicaciones, el almacenamiento de datos, etc.), se ha llevado a cabo un estudio de plataformas, dispositivos y arquitecturas similares en la literatura. Este estudio del estado del arte permite concluir que existe un amplio número de arquitecturas, plataformas y tecnologías, no siempre compatibles entre sí, y que derivan de la utilización del paradigma IoT con objetivos muy diversos. Para el caso de uso planteado en esta investigación, no existe actualmente una arquitectura específica que cumpla con los requisitos planteados, por lo que se ha determinado que el camino a seguir es la definición de una arquitectura concreta y específica que permita diseñar los mecanismos genéricos sobre los que implementar la plataforma.

Para asegurar en la medida de lo posible la interoperabilidad y el cumplimiento de los requisitos de las arquitecturas IoT, se ha diseñado esta arquitectura basándose en un modelo de referencia que sirve de guía a la hora de definir las funcionalidades, módulos y estructuras de la plataforma.

A partir de estos diseños, en el marco de esta investigación, y en conjunción con el proyecto EDUCERE [32] en el cual se enmarca, se han diseñado y construido una serie de prototipos que han servido para evaluar de forma preliminar la plataforma. En estas pruebas, se han realizado experimentos utilizando los dispositivos en escuelas infantiles, recogiendo los datos obtenidos a partir de los sensores y procesándolos para obtener información que pudiera ser útil a los expertos.

Estos resultados muestran que, si bien existe una gran cantidad de información disponible a partir de las mediciones realizadas por los dispositivos, no toda la información tiene la misma utilidad cuando es comparada con las evaluaciones "tradicionales" de los expertos, y es necesario realizar un esfuerzo específico para transformar los datos en información útil. Esta conclusión refuerza además el trabajo descrito en el capítulo 4 de este mismo libro.

2.2. Estado del arte y trabajos relacionados

Para un correcto diseño de la plataforma de Smart Toys es necesario apoyarse sobre una arquitectura que asegure que las características y requisitos planteados para ella se cumplan, sin comprometer por otro lado los requisitos generales de fiabilidad, escalabilidad, interoperabilidad, seguridad, etc., que deben compartir estos sistemas. Por otro lado, para la correcta construcción de los dispositivos que se utilicen en las actividades de juego por parte de los niños, es también importante estudiar los ejemplos y propuestas de Smart Toys existentes. En los siguientes apartados se lleva a cabo un estudio de las arquitecturas y plataformas IoT en la literatura, así como de las propuestas de Smart Toys, sus objetivos y las tecnologías utilizadas en su construcción.

2.2.1. Arquitecturas IoT

En los últimos años han surgido un gran número de propuestas de arquitecturas y plataformas basadas en el Internet of Things, orientadas a múltiples objetivos y utilizando tecnologías y metodologías distintas [33]. Así, por ejemplo, se puede determinar una clasificación de los distintos tipos de arquitectura, tal y como se propone en [34]. Según este trabajo, se pueden clasificar atendiendo a cómo se dividen en capas, o a dónde se produce el procesado de los datos.

Las arquitecturas basadas en capas son las más habituales en la literatura. Se pueden dividir a su vez en dos aproximaciones distintas: Aquellas que utilizan tres capas y aquellas que utilizan cinco capas. Una arquitectura en tres capas [35] se suele componer de una capa de percepción (donde se sitúan los sensores que proporcionan

la información), una capa de red (encargada de las comunicaciones entre los elementos del sistema) y una capa de aplicación (formada por los servicios ofrecidos al usuario). Las arquitecturas de cinco capas [36], por otro lado, mantienen las capas de aplicación y de percepción y añaden otras tres capas: Capa de negocio, capa de procesado y capa de transporte. La primera determina los procesos de negocio y la gestión de la plataforma, la segunda define los procesos de análisis y procesado de los datos obtenidos de los sensores, y la tercera es la encargada de la transferencia de datos desde la capa de percepción a la capa de procesado.

Atendiendo al procesado de los datos, se puede hablar de arquitecturas más cercanas al concepto de Cloud Computing [37] o al concepto de Fog o Edge Computing [38, 39]. En el primer caso, se deriva el procesado de los datos generados por los sensores a plataformas basadas en Cloud, y el acceso a los mismos se realiza a través de aplicaciones que se conectan a dichas plataformas [40]. Por el contrario, en las arquitecturas más cercanas al Fog o Edge Computing, se realiza un procesado de los datos de forma distribuida en una serie de gateways encargados de este procesado, así como del almacenamiento temporal y la seguridad [41].

En cuanto a los objetivos de dichas arquitecturas y plataformas, se pueden identificar numerosos casos de uso y ejemplos. En [42] se dividen en diversos dominios de aplicación: Aplicaciones en transporte y logística [43,44], aplicaciones para la salud [23], para la construcción de entornos inteligentes (casas, oficinas, ciudades etc.) [45,46], para el ámbito personal o social (redes sociales, datos personales) [47], u otras futuras aplicaciones.

La alta heterogeneidad en las arquitecturas, tecnologías y aplicaciones de las plataformas IoT tiene una importante desventaja: La difícil interacción e interoperabilidad entre una y otra plataforma, pese a utilizar un paradigma común en su concepción [48]. Para evitar esta problemática, se han llevado a cabo una serie de proyectos e investigaciones cuyo objetivo es precisamente ofrecer marcos comunes para el diseño de futuras plataformas IoT. La Unión Europea ha sido especialmente proactiva en este sentido mediante la financiación de varios proyectos para la definición de marcos de referencia, guías de diseño y arquitecturas interoperables [49].

De entre estos proyectos, hay que destacar algunos como FI-WARE [50], un ambi-

cioso proyecto cuyo objetivo es diseñar el núcleo del futuro Internet. Para ello se basa en el concepto de Generic Enablers (GE), una definición de componentes reutilizables genéricos que se clasifican dependiendo de su funcionalidad (gestión de datos, recursos Cloud, servicios y aplicaciones, e IoT) [51]. El proyecto SENSEI [52] fue concebido con un objetivo similar, la construcción de una arquitectura abierta e interoperable que integrara múltiples redes de sensores heterogéneas en un marco global. El proyecto SPITFIRE [53] por otro lado, se centra en la definición de un modelo semántico para los escenarios IoT. Finalmente, el proyecto IoT-A [54] ha desarrollado un marco de referencia para el diseño de arquitecturas IoT con el objetivo de asegurar la compatibilidad entre ellas, sirviendo además de guía para un correcto diseño de plataformas para entornos concretos. Otro proyecto oneM2M [55], en este caso conformado por un gran número de compañías tecnológicas y otras organizaciones, se centra en obtener una estandarización de arquitecturas, protocolos y tecnologías IoT a nivel global [56].

2.2.2. Smart Toys

Además de la definición arquitectónica de la plataforma, en este capítulo se describen las implementaciones concretas llevadas a cabo para obtener prototipos funcionales de Smart Toys con los que alcanzar resto de objetivos planteados en este trabajo. Para construir estos dispositivos, se han estudiado las principales tecnologías utilizadas en IoT, y especialmente en la definición de sistemas de juguetes inteligentes. Este tipo de sistemas, que se ha venido llamando "Internet of Toys" o IoToys [57], son relativamente recientes, encontrándose trabajos para la interconexión aproximadamente a partir de mediados de la primera década de este siglo. Ejemplos son los trabajos mostrados en [58], [59], o [60]. Los tres trabajos muestran sistemas de comunicaciones específicamente diseñados para interconectar juguetes entre sí, siendo los dos primeros dos patentes. Este tipo de propuestas se pueden considerar la antesala de la aparición de un Internet específico para Smart Toys.

Los Smart Toys, tanto en su concepción como parte de un "Internet of Toys" o como dispositivos inteligentes independientes, han sido ideados con varios objetivos distintos en mente. Se pueden identificar principalmente tres líneas principales en la literatura:

- Smart Toys con fines educativos: Existen muchos trabajos cuyo enfoque es la construcción de dispositivos que, mediante la inclusión de tecnología, permitan favorecer la educación infantil [61, 62], ya sea mediante la proposición de actividades innovadoras [63], o bien mediante el uso de las posibilidades que ofrecen los sensores para enseñar distintas aptitudes [64].

- Smart Toys con fines de entretenimiento: Existen también enfoques cuyo principal objetivo es mejorar el entretenimiento que proporcionan los juguetes a los niños [65,66], normalmente aprovechando las posibilidades de realimentación que proporcionan los sensores [67] y las posibilidades de ampliar el juego a un entorno social mediante la interconexión de los dispositivos [68].

- Smart Toys relacionados con la salud: Aunque este es un enfoque relativamente menos explorado que los anteriores, existen en la literatura trabajos en los que, como en esta Tesis, se propone el uso de juguetes con fines sanitarios. En el trabajo [69], donde se estudia uso de tecnologías IoT para personas con discapacidades, se identifican Smart Toys como posibles herramientas para el aprendizaje en niños con discapacidad auditiva. En [70] o [71] por ejemplo, se proponen tecnologías para la detección de problemas de desarrollo psicomotor y niños en riesgo de padecer trastornos del espectro autista. También en [72] se puede encontrar una propuesta para la evaluación de las capacidades motoras de niños mediante Smart Toys.

En cuanto a las características tecnológicas de los Smart Toys propuestos en la literatura y en entornos comerciales, existe una gran diversidad como ya sucediera con las plataformas IoT en general. Muchos de los desarrollos comerciales en Smart Toys se basan en conexiones Wi-Fi [62], aunque existen plataformas basadas en Bluetooth [73] y otras alternativas en cuanto a las comunicaciones, como las comunicaciones mediante luz visible [74] o la propuesta de comunicaciones salto a salto definida en [75]. El uso de RFID (Identificación por Radio Frecuencia) es también habitual, por ejemplo, para ofrecer información extendida sobre los dispositivos [76].

Los sensores utilizados también son muy diversos, aunque en muchos casos se utilizan acelerómetros [77, 78] o sensores inerciales (conteniendo además giróscopos y magnetómetros) [79], sensores de presión [80], sensores de reconocimiento de voz [81], e

incluso cámaras [82], entre otros, dependiendo del objetivo de cada uno de ellos.

En algunas de las propuestas anteriores los Smart Toys son relativamente autónomos. Es decir, envían información a una plataforma, pero no ofrecen la posibilidad de utilizar la información de múltiples fuentes para generar servicios de análisis o agregación de la información. En otros casos, especialmente en los entornos comerciales, se han definido plataformas de este tipo. En [83] hay por ejemplo una propuesta para este tipo de sistemas interconectados basados en Cloud, que se acerca más al concepto original de Internet of Things, y en artículos como [84] se identifican las implicaciones del uso de este tipo de sistemas en niños.

2.3. Diseño de la plataforma

En esta sección se determinan los requisitos definidos para la plataforma y, a partir de ellos y del modelo de referencia utilizado en su diseño, se describe la arquitectura de ésta, desde cada una de las vistas de modelado utilizadas.

2.3.1. Requisitos, restricciones y términos de la plataforma

El diseño de esta plataforma depende de las especificaciones que se puedan derivar de los requisitos y restricciones que se han determinado a partir de los objetivos generales del proyecto y de las diversas fuentes disponibles. Estos requisitos servirán de punto de partida para el diseño de la arquitectura. Además, en esta sección se definen algunos términos que se utilizan a lo largo del diseño de la plataforma y del resto de este trabajo de investigación.

2.3.1.1. Requisitos

En este apartado se presentan los principales requisitos funcionales de la arquitectura a diseñar. Estos requisitos se han obtenido en base a: Entrevistas con expertos en desarrollo infantil, documentación sobre desarrollo infantil y detección temprana de trastornos en el desarrollo, el estudio del estado del arte en entornos y arquitecturas

IoT y las tecnologías disponibles.

A partir de estas fuentes, se han determinado una serie de requisitos clasificados en los siguientes grupos:

1. **Requisitos relacionados con el desarrollo infantil**

   a) La plataforma debe permitir la obtención de datos de actividades de juego.

   b) Los datos obtenidos deben ofrecer información adicional sobre la información utilizada actualmente en los procedimientos existentes para la evaluación del desarrollo infantil.

   c) Los dispositivos definidos en la plataforma deben tener en cuenta las herramientas y procedimientos existentes para la evaluación del desarrollo infantil y ser compatibles con ellos en la medida de lo posible.

   d) La plataforma debe favorecer la facilidad de uso para los expertos en desarrollo.

   e) La plataforma debe favorecer la facilidad de uso para los usuarios de los dispositivos (niños principalmente).

   f) La plataforma debe permitir el análisis de los datos obtenidos.

   g) Las actividades de juego no deberían ser interrumpidas por la autonomía de los dispositivos.

   h) Ni la plataforma ni los dispositivos deben ser intrusivos en el juego infantil: No se debería necesitar instalar ningún sensor, cámara u otro elemento extra en el cuerpo o en el entorno.

   i) La plataforma debe ser capaz de recopilar datos durante el juego del niño aunque no haya una conexión a Internet presente, al no poderse asegurar su disponibilidad en todos los lugares donde ésta estará disponible (escuelas infantiles, casas, etc.).

2. **Requisitos relacionados con los entornos y tecnologías IoT**

   a) La arquitectura debe ser fácilmente extensible a nuevos dispositivos, juegos, o herramientas que se diseñen en el futuro.

b) Varios modelos de comunicaciones deben coexistir en la plataforma.

c) El consumo energético se debe tener en cuenta en el diseño de la plataforma y sus dispositivos.

d) El precio de construcción de los dispositivos no debe ser excesivo.

e) Los dispositivos deben poder comunicar las lecturas de sensores de forma remota.

f) Los dispositivos deben ser capaces de llevar a cabo un preprocesado de los datos.

g) La plataforma debe ofrecer un mecanismo de almacenamiento de los datos y de la información obtenida a partir de ellos.

h) Los datos de la plataforma deberían ser accesibles desde cualquier lugar a través de Internet.

3. **Requisitos relacionados con la seguridad**

a) Los datos personales de los usuarios deben ser confidenciales, especialmente los de los niños.

b) Los datos obtenidos a partir de los dispositivos de la plataforma deben ser confidenciales y no poder ser interceptados durante su transmisión.

c) La plataforma debe permitir acceso a distintos datos dependiendo del tipo de usuario que solicite acceder.

d) Los mecanismos de seguridad deben ser lo más transparente posible para los usuarios

e) La activación de los dispositivos de la plataforma sólo debe poder llevarse a cabo por usuarios autorizados.

f) Los mecanismos de seguridad deberán amoldarse a los estándares existentes en la medida de lo posible.

4. **Otros requisitos de la plataforma**

a) La plataforma debe poder extender su uso a otros tipos de usuarios. Es decir, si se determina que la utilización de los dispositivos puede ser beneficiosa para otras personas (personas mayores, rehabilitación, etc.), debe poder ser adaptada fácilmente.

b) La plataforma debe permitir generar nueva información o servicios mediante la composición y procesado de los datos.

c) Se debe garantizar la posibilidad de cierta interoperabilidad. Debe ser posible obtener información de otras fuentes.

2.3.1.2. Definición de términos en la arquitectura

A lo largo del análisis de requisitos se han ido tomando una serie de decisiones sobre la utilización de algunos términos que permitan, durante el diseño arquitectónico, cierta precisión en la definición de cada módulo o vista. Además, estos términos han permitido, durante la recogida de requisitos, una mayor claridad en la determinación de cada uno de ellos. A continuación, se definen estos términos:

- Actividad: Cada uno de los posibles juegos a llevar a cabo por un niño a través de la plataforma.
- Dispositivo: Aquellos elementos hardware de la plataforma que permitirán la realización de actividades. También se habla indistintamente de "juguetes".
- Dato: Cada uno de los elementos de información obtenidos a través de un dispositivo o servicio de la plataforma. Se consideran datos tanto aquellos ofrecidos directamente por los dispositivos IoT (datos de sensores directamente) como aquellos que han sido preprocesados o procesados *a posterori* mediante su análisis, agregación, etc.
- Test: Cada una de las pruebas realizada con un niño y una actividad concreta.
- Sesión: Es cada uno de los periodos continuados de tiempo en los que se realizan varios tests con uno o varios niños.
- Experimento: Se define como un conjunto de tests realizados en una o varias sesiones con un conjunto de niños y que ofrece al final un conjunto de datos concreto para ser analizado con un objetivo común.
- Plataforma: Se define como el sistema descrito por la arquitectura diseñada en este capítulo. La plataforma de Smart Toys es la herramienta utilizada para

obtener y analizar los datos provenientes de actividades de juego.

- Experto: Definimos un experto como un tipo de usuario específico de la plataforma. Concretamente, serán los usuarios que accederán a los datos generados para analizarlos y sacar conclusiones de ellos. Habitualmente se tratará de psicólogos, fisioterapeutas, pedagogos, etc.

2.3.2. Diseño de la arquitectura

Los requisitos funcionales generales de la arquitectura, junto con la necesidad de definir una plataforma genérica, extensible e interoperable en la medida de lo posible, llevan a la conclusión de que uno de los enfoques más apropiados en cuanto al diseño es la utilización de un marco referencial que permita "estandarizar" el diseño en la medida de lo posible. Sobre este marco referencial, se han determinado las funcionalidades que necesita la arquitectura, y se han diseñado los elementos que la componen.

Así, después de evaluar las posibilidades existentes para este marco, se ha optado por la utilización de la arquitectura de referencia publicada por el proyecto IoT-A [85], citado en la sección 2.2. Este proyecto tenía como objetivo precisamente la definición de un marco arquitectónico que permitiera la definición posterior de arquitecturas específicas para cada problemática relacionada con un ecosistema IoT. El adherirse a este marco y a estas recomendaciones de diseño permite asegurar que las arquitecturas resultantes serán en gran medida compatibles entre sí, interoperables, escalables y extensibles, todas estas características imprescindibles para cualquier plataforma de este ámbito. La selección de este modelo se basa en la propuesta única ofrecida por este proyecto para la definición de arquitecturas IoT desde un alto nivel, sin centrarse en tecnologías o casos de uso específicos, dando por un lado la libertad necesaria para definir las estructuras de datos, de comunicaciones y tecnologías concretas necesarias en cada caso, pero asegurando la correcta definición de las funcionalidadees de cada plataforma diseñada.

La arquitectura de referencia de IoT-A se basa en lo que se conoce dentro de la Ingeniería del Software como vistas arquitectónicas. Una vista es una representación de uno o más aspectos estructurales de una arquitectura y define cómo la arquitectura

es capaz de abordar la problemática concreta a resolver [86]. Cada vista se compone a su vez de una serie de patrones, plantillas y convenciones utilizadas para construirlas en base a los requisitos que se determinen a priori. Se pueden definir al menos seis tipos de vistas:

- Vista Funcional: Define los elementos arquitectónicos del sistema que conseguirán ofrecer las funcionalidades deseadas.
- Vista de Información: Describe la forma en la que el sistema almacena, gestiona y representa la información.
- Vista de Entidades Físicas: En esta vista se definen los elementos físicos que componen el sistema y cómo se relacionan entre sí y con las funcionalidades que ofrece.
- Vista de Contexto: Describe las relaciones entre la plataforma y el entorno en el que se va a utilizar (personas, otros sistemas, etc.).
- Vista Operacional: Describe cómo se va a gestionar y administrar el sistema una vez desplegado.
- Vista de Despliegue: Describe el entorno donde se va a desplegar el sistema.

La arquitectura de referencia IoT-A se centra fundamentalmente en las dos primeras vistas, dado que no pretende definir de forma específica los detalles de cada arquitectura concreta físicamente, ni las tecnologías concretas que se usarán en el desarrollo y despliegue de las mismas. Más bien al contrario, el objetivo de IoT-A es permitir desarrollar de forma independiente cualquier arquitectura que, en base a las guías proporcionadas, pueda después ser definida completamente en base a los requisitos y restricciones específicas. En el caso de la arquitectura de Smart Toys definida en el presente trabajo, se han utilizado las vistas funcionales y de información como base para después desarrollar el resto de elementos arquitectónicos, como se verá más adelante.

En los siguientes apartados se van a definir cada una de las vistas anteriores. Para las dos primeras (Vista Funcional y Vista de Información) se han basado los diseños en

los modelos de referencia de IoT-A. En el resto de casos, se han seguido las recomendaciones del marco de referencia, y a partir de ahí se han aplicado las decisiones de diseño particulares derivadas de los requisitos específicos de la plataforma de Smart Toys que se han observado en el estudio de la sección anterior. No se han incluido descripciones de las vistas operacional y de despliegue ya que, al tratarse de una plataforma con un objetivo fundamentalmente experimental, no consideramos que sean esenciales en el proceso de diseño.

2.3.2.1. Vista Funcional

En IoT-A, la vista funcional se basa en una serie de componentes o grupos funcionales que forman la totalidad de la arquitectura. Cada componente identifica una serie de funcionalidades comunes, y por tanto están formadas a su vez por grupos funcionales donde se determinan las funciones específicas que se deben contemplar.

En la Figura 2.1 se puede ver un diagrama donde se descomponen los módulos funcionales que debería implementar un sistema según la referencia IoT-A.

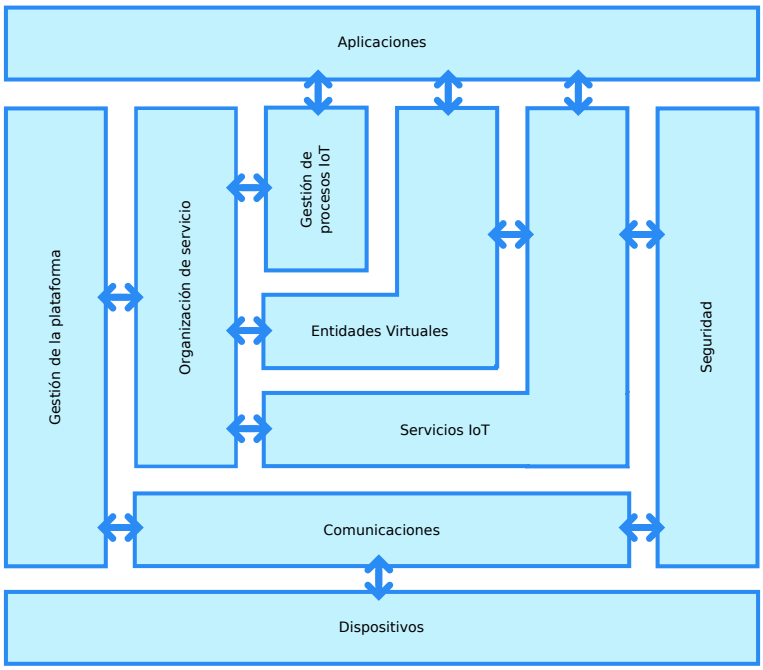

Figura 2.1: Diagrama de los grupos funcionales de la arquitectura de referencia IoT-A.

Cada uno de estos grupos funcionales tiene un objetivo y unas características concretas. No todos los grupos funcionales serán definidos en la arquitectura de juguetes inteligentes, al no ser necesarias algunas de las características que agrupan.

En la Figura 2.2, se pueden ver las relaciones entre los módulos funcionales de la arquitectura de juguetes inteligentes y sus homólogos de la arquitectura de referencia. A continuación, se presenta una descripción de cada uno de estos módulos y su relación con el modelo referencial. Como se puede ver en la figura, no todos los grupos funcionales se van a representar en el diseño de la plataforma de Smart Toys, agrupándose algunas de las funcionalidades en un único grupo. De esta forma, se incide con más detalle en aquellos módulos que más utilidad inmediata pueden ofrecer en el caso de uso que nos ocupa.

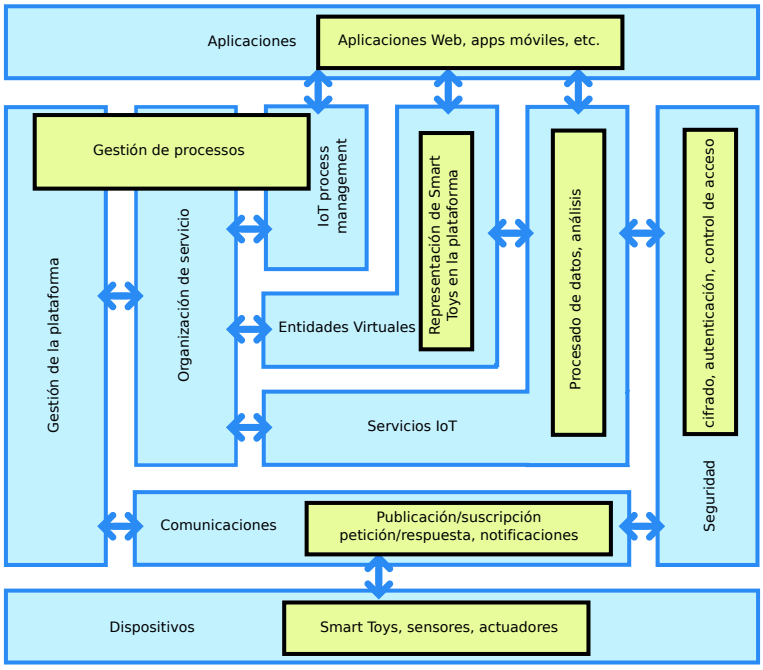

Figura 2.2: Diagrama de los grupos funcionales de la arquitectura de Juguetes Inteligentes y su relación con el modelo de referencia IoT-A.

1. **Gestión de procesos**

   En este grupo funcional se incluyen todas aquellas funcionalidades relacionadas con los llamados procesos de negocio (*Business Processes*). Estos procesos determinan a alto nivel las secuencias de actividades o tareas que, ejecutadas de una

manera secuencial, producirán al final un producto o servicio. Por tanto, este grupo funcional debe definir las metodologías y definiciones de actividades que permitan llevar a cabo los procesos de alto nivel determinados por los requisitos del sistema. Además, debe determinar cómo se deben gestionar y actualizar estos procesos a lo largo del tiempo, con el fin de asegurar que siempre se obtengan los resultados esperados.

Para la definición de estos procesos, se utiliza habitualmente un sistema de notación conocido como Bussiness Process Model and Notation (BPMN) [87], una metodología estandarizada que ha sido especialmente diseñada para definir estos procesos de negocio.

Aunque este modelo funcional ha surgido para la definición de procesos de negocio y por tanto es fundamental en aquellas arquitecturas donde el objetivo final sea la obtención de un producto o dar un servicio determinado (esto es, habitualmente en arquitecturas pensadas para empresas u organizaciones privadas), es un sistema exportable a los servicios que se desean proporcionar en plataformas e-health o relacionadas con la salud [88] como es esta cuyo diseño se está describiendo.

Así, utilizando el sistema BPMN, se han determinado una serie de procesos que serán aquellos que deberán ser gestionados por esta entidad funcional (aunque su ejecución estará físicamente distribuida en los distintos módulos físicos que componen el sistema, como se verá más adelante).

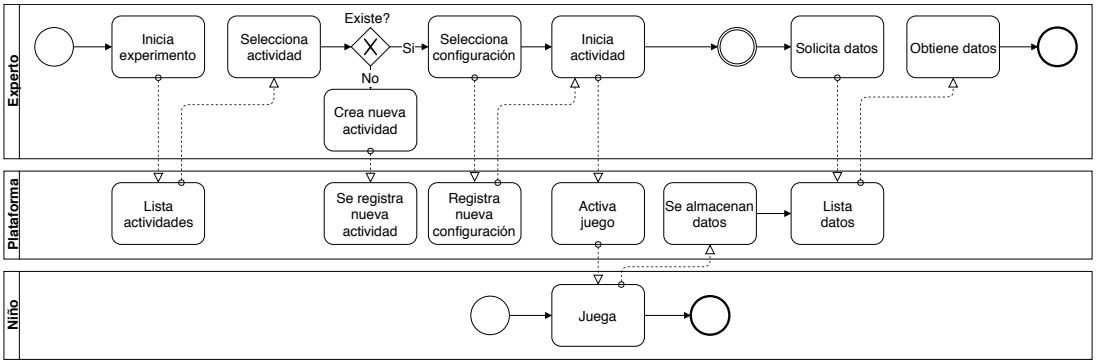

Figura 2.3: Diagrama de gestión de proceso para la realización de una actividad y la obtención de datos de una actividad.

En la Figura 2.3 se muestra el diagrama BPMN que describe las tareas a realizar

para la obtención de datos a través de la plataforma. El proceso comienza por parte de un experto (por ejemplo, un psicólogo o un psicopedagogo), que desea obtener nuevos datos desde los Smart Toys. Para lograrlo, a través de la plataforma pone en marcha un nuevo experimento. Inicialmente, se deberá determinar sobre qué actividad desea obtener los datos (véase la definición de actividad en la sección 2.3.1). Si no existe una descipción de la actividad deseada en la plataforma, se debe incluir antes de comenzar el experimento. En cualquier caso, con el experimento iniciado, el niño o niños que sean sujetos de la actividad podrán jugar con los dispositivos IoT. Estos dispositivos enviarán los datos a la plataforma, y el experto podrá recuperarlos.

Otro ejemplo de proceso a gestionar es el que se muestra en la Figura 2.4. En este caso, se define cómo un niño podría jugar con juguetes activos y los datos se almacenarían de forma transparente en la plataforma. El experto podría contar con esa información a través de servicios de alertas u otros mecanismos de suscripción.

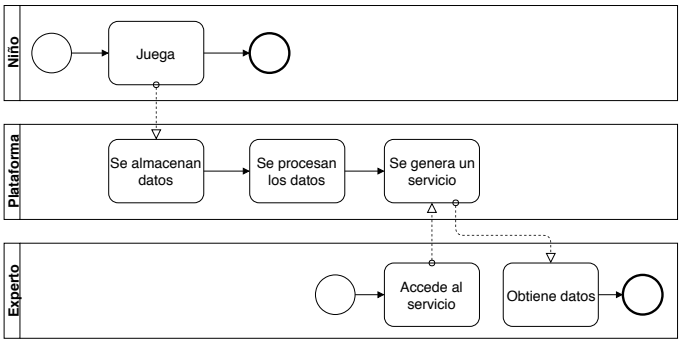

Figura 2.4: Diagrama de gestión de proceso para la obtención de datos a de la plataforma.

En la Figura 2.5 se tiene un tercer caso de proceso, en el que se identifica cómo añadir nuevas actividades de juego a la plataforma. Este proceso se realizará por parte del experto, que deberá introducir la configuración y datos a obtener en la misma. A partir de ese momento, el catálogo de actividades se ampliará con la nueva actividad.

El grupo funcional de organización de servicios en IoT-A tiene como objetivo ser el elemento a través del cual se comuniquen muchas de las otras entidades funcionales, concretamente aquellas que tienen relación con la generación y composición

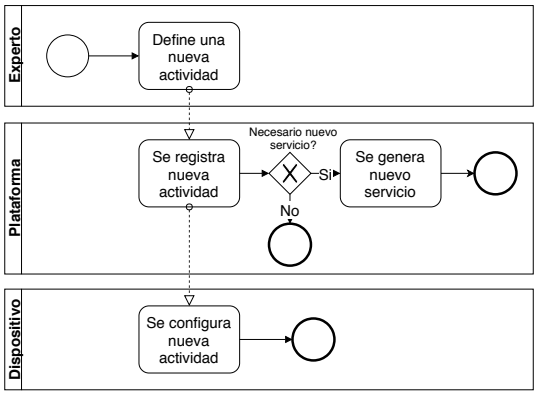

Figura 2.5: Diagrama de gestión de proceso para la generación de una nueva actividad en la plataforma.

de servicios que puedan ser utilizados posteriormente por los usuarios.

La definición de los servicios se abordará de nuevo en el punto 3 de esta lista (Servicios IoT). En el caso de la arquitectura, esta entidad funcional determinaría la metodología y normas a seguir para la orquestación y agregación de servicios. Esto es, la implementación de nuevos servicios a partir de los disponibles en la propia plataforma. Aunque existen varios procedimientos y estándares para la definición de los métodos de agregación y orquestación de los servicios ("Lifecycle Service Orchestration" (LSO), por ejemplo [89]), en nuestro diseño se ha optado por simplificar la definición de los servicios lo más posible, con el objetivo de tener una funcionalidad básica implementada de forma rápida para la experimentación inicial. Por tanto, no se han tenido en cuenta estos procesos en esta iteración del diseño. Este grupo funcional se ha integrado en el grupo de gestión de procesos, junto con el grupo funcional de gestión de la plataforma.

Finalmente, en el grupo funcional de gestión de la plataforma se determina cómo se van a definir las tareas de gestión de la propia plataforma, con el objetivo de evitar posibles incidencias en su uso. Dado el carácter experimental de la plataforma, se ha decidido que estas tareas se pueden integrar y definir desde el grupo funcional de gestión de procesos que engloba la gestión de procesos IoT, la organización de servicios y estas funciones.

2. **Entidades virtuales**

Las Entidades Virtuales son un elemento funcional clave en la arquitectura de

referencia IoT-A, ya que proporcionan una abstracción de los dispositivos IoT dentro de la propia arquitectura. Es decir, son un objeto abstracto que representa de una o más formas cada dispositivo conectado a la misma, de forma que sirve como elemento intermedio entre los usuarios, servicios y aplicaciones y los propios dispositivos.

Las Entidades Virtuales proporcionan las herramientas necesarias a los servicios para su descubrimiento y ayudan a la entidad funcional de organización de servicios a la composición de los mismos, en base a cómo están definidas.

En el caso de la arquitectura de Smart Toys, las entidades virtuales que representan los dispositivos IoT (es decir, los propios juguetes físicos), son elementos software que determinan cómo ofrecen su información los dispositivos concretos (desde el formato de datos a los protocolos de solicitud o suscripción), y cómo se puede acceder a ellos desde la plataforma o sus servicios.

Una Entidad Virtual puede por ejemplo proporcionar datos históricos de un dispositivo concreto, o definir una serie de operaciones de postprocesado sobre los datos del dispositivo para ofrecer a su vez datos agregados que den información más útil a los usuarios, servicios o aplicaciones que los usen.

Por ejemplo, un Smart Toy concreto será visto desde la plataforma como una estructura de datos que contendrá sus datos (tipo de juguete, características, actividades que puede realizar, etc.), los datos relacionados con él (historial de datos obtenidos a través de él, etc.) y los servicios que ofrece al usuario. Habitualmente la forma de acceder a los datos de la entidad virtual será a través de un servicio accedido a través de una API ("Application Programming Interface").

Estas Entidades Virtuales se definen mediante diagramas de entidad tal y como se verá en el primer apartado de la sección 2.3.2.2.

3. **Servicios IoT**

   En este grupo funcional se describen los servicios IoT, y se determinan sus funcionalidades específicas. Es decir, se determina cómo se realiza el acceso a un servicio y a la información que proporciona, normalmente definiendo el punto o puntos de acceso al servicio como recursos. Además, un servicio estará determinado por su identificación dentro del sistema (mecanismo de resolución del

servicio), que permitirá identificar y descubrir los servicios disponibles por parte de usuarios y aplicaciones.

En esta plataforma, los servicios se basan fundamentalmente en las Entidades Virtuales, definiéndose un punto de acceso por cada una de ellas, permitiéndose la inclusión posterior de nuevos servicios que ofrezcan acceso a datos post procesados o analizados en la propia plataforma.

Para el acceso a los servicios se definen dos tipos de interfaces básicas: Por un lado, se definen puntos de acceso a través de los cuales se pueden realizar operaciones de tipo REST ("REpresentational State Trasfer") que permitan su manejo y el de la información asociada a ellos. Es decir, mediante interfaces REST se permitirá el acceso, modificación, borrado y añadido de información a estas entidades.

Por otro lado, se define la posibilidad del uso de comunicaciones basadas en suscripción a servicios para obtener alertas e información desde éstos sin necesidad de una solicitud explícita. Para esto, se definirán canales basados en protocolos de intercambio de mensajes basados en un modelo de publicación/suscripción (Por ejemplo, AMQP ("Advanced Message Queuing Protocol"), MQTT ("Message Queue Telemetry Transport"), etc.).

Los servidores de la plataforma se definen para incluir el listado de los servicios disponibles, cuyo acceso se hará a través de URLs o direcciones basadas en el protocolo de suscripción correspondiente. También es posible el acceso a estos servicios de suscripción a través de WebSockets en los casos en los que se ofrezca una URL para ello.

El registro de nuevos servicios en la plataforma requiere la inclusión de información relativa a ellos que permita su búsqueda en los servidores.

4. **Comunicaciones**

   El grupo funcional de comunicaciones es aquel que determina cómo se establecen las comunicaciones entre cada elemento de la arquitectura. Define estos intercambios de información de manera abstracta, así como los protocolos a alto nivel para las comunicaciones, siempre permitiendo que las tecnologías subyacentes sean lo más amplias posibles, ya que no es en esta vista donde se debe decidir acerca de las tecnologías específicas a utilizar en la implementación. En la plataforma se pueden dividir las comunicaciones en tres subgrupos:

- Comunicaciones salto a salto: Se refiere a las comunicaciones de más bajo nivel, aquellas que permiten la comunicación entre dispositivos IoT o entre estos dispositivos y los dispositivos que sirvan de pasarela para con el resto de la red. Las características de las comunicaciones a este nivel deben definirse de forma que se permita transmitir la mayor cantidad de información posible con el mínimo coste, teniendo siempre en cuenta que se trata de comunicaciones cercanas, y que muchos de los dispositivos involucrados no serán capaces de implementar complejos protocolos de comunicación.

  Además, para la comunicación dispositivo-dispositivo se deben tener en cuenta los sistemas de enrutado este tipo de redes, así como la prioridad de unas comunicaciones con respecto a otras.

  En la plataforma de Smart Toys se permite la convivencia de varias tecnologías de comunicaciones punto a punto (Bluetooth, Wi-Fi, otras comunicaciones basadas en radio-frecuencia, etc.), dependiendo de los requisitos específicos de cada Smart Toy. En la sección 2.4.2, se explica cuales se han utilizado en la construcción de los dispositivos, pero en el diseño arquitectónico se deja abierta esta cuestión para poder incorporar nuevos sistemas en el futuro.

- Comunicaciones de red: Son aquellas que permiten la intercomunicación entre una red local y otra. Se trata de definir las necesidades de interconexión entre redes que pueden surgir, para asegurar la interoperabilidad en la mayor medida posible a nivel de red. Las pasarelas serán los módulos físicos que contendrán la mayor parte de las funciones de este grupo.

  En esta primera iteración se supondrá un acceso estándar entre redes a través de interfaces Wi-Fi o Ethernet, simplificando así las tareas de interconexión que pueden surgir.

- Comunicaciones extremo a extremo: Son las comunicaciones entre un extremo final de la comunicación, ya sean por ejemplo dos dispositivos conectados a través de Internet, un dispositivo y un servidor remoto, o un dispositivo y un usuario. Estas comunicaciones deben diseñarse de forma que sean seguras, fiables, etc. para asegurar la transmisión a través de varias redes (o presumiblemente, Internet).

  En el caso de la plataforma se utilizan los protocolos estándar de Internet

(HTTPS (“ Hypertext Transport Protocol Secure”), etc.) para asegurar en la medida de lo posible este tipo de comunicaciones, y se añaden sobre estos protocolos propuestas para el control de acceso y la privacidad, como se verá en el capítulo 3.5 de este libro.

5. **Seguridad**

   Uno de los elementos funcionales más importantes en este tipo de arquitecturas es la seguridad. De hecho, en la Figura 2.1 se puede ver cómo es un grupo funcional transversal que ha de ser tenido en cuenta a lo largo de todo el diseño.

   El diseño de seguridad debe consistir al menos en mecanismos para tener sistemas de autenticación, gestión de identidades y autorización, así como sistemas de cifrado e intercambio de claves.

   La importancia de estas funciones, así como la necesidad de detallar las aportaciones que se han realizado a las mismas en el presente estudio han determinado que el diseño de seguridad específico diseñado se haya separado en un capítulo concreto (el capítulo 3) donde se especificarán todas las decisiones tomadas en cuanto a estas funcionalidades.

2.3.2.2. Vista de Información

La vista de información permite poner el foco del diseño arquitectónico sobre cómo se representa la información dentro del sistema IoT. En esta vista, se determinan las representaciones de datos, qué componentes de la arquitectura se encargan de gestionar estos datos, y el manejo de la información durante todo su ciclo de vida.

1. **Descripción de la información**

   Para determinar la forma en la que la información es descrita en la arquitectura, hay que acudir al concepto de Entidad Virtual, descrito anteriormente. La Entidad Virtual es el elemento que identifica cada dispositivo IoT en la plataforma, y por tanto se le asignará un identificador único. Cualquier operación que se desee realizar con un dispositivo físico, se hará a través de estas Entidades Virtuales. Para modelarlas, se pueden utilizar diagramas de entidades UML, tal y como se

ve en el ejemplo de la Figura 2.6. El modelado de las entidades sigue habitualmente un patrón jerárquico, donde se pueden determinar las relaciones que hay entre cada una dentro del sistema.

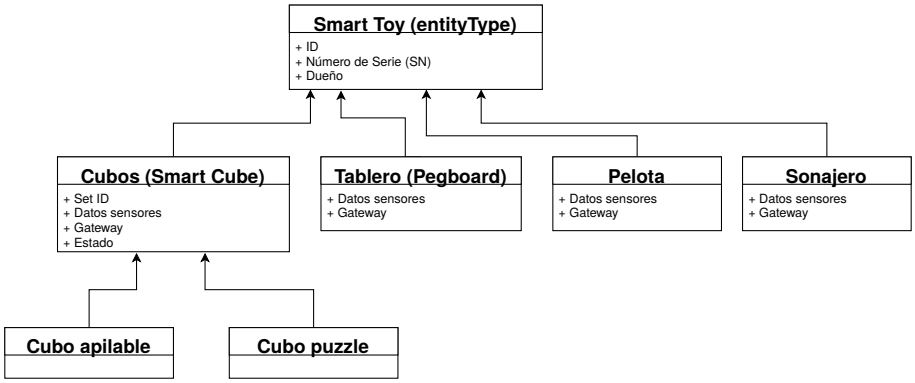

Figura 2.6: Diagrama entidad-relación jérarquico que define las entidades virtuales derivadas de Smart Toys.

Como se puede ver en la figura, la entidad virtual principal de la plataforma es el Smart Toy. A partir de esta entidad, que se identifica mediante un ID en la plataforma y mediante un Número de Serie único para cada dispositivo físico, se derivan todos los juguetes que se definen o se puedan definir posteriormente como dispositivos físicos. Entre ellos, se han definido inicialmente Smart Cubes (cuyo diseño se incluirá en la sección 2.4.2 donde se describen los prototipos fabricados), un tablero de espigas (también explicado en dicha sección), una pelota y un sonajero. Estas dos últimas entidades se incluyen como ejemplos de futuros diseños a incorporar a la plataforma. Desde la entidad que representa a los Smart Cubes, se derivan otras dos entidades más específicas, que identifican dos juguetes físicos que se pueden construir a partir del mismo Smart Cube. Uno es el conjunto de cubos apilables y otro es un conjunto de cubos para formar un puzle.

2. **Información y componentes funcionales**

Es importante determinar la relación de la información con cada componente funcional descrito en el apartado anterior, ya que es necesario determinar qué funciones de qué grupo se deben utilizar en la transferencia efectiva de la información desde un dispositivo a una aplicación. En la Figura 2.7 se pueden ver

los grupos funcionales por los que pasará un dato de aceleración obtenido en un juguete de la arquitectura, hasta poder ser leído en una aplicación cliente por parte de un usuario.

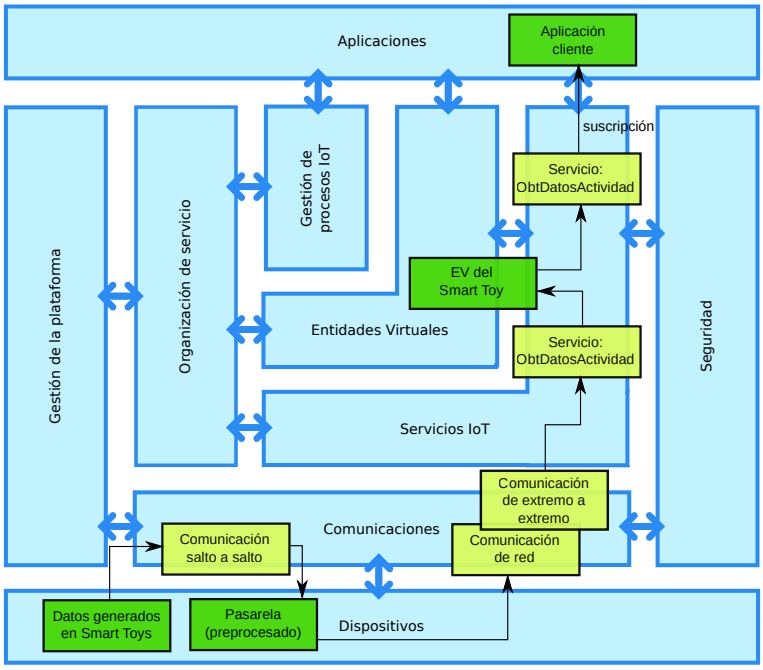

Figura 2.7: Flujo de información a través de grupos funcionales de la arquitectura (suscripción).

Además, el flujo de la información entre los diferentes componentes de la arquitectura puede tomar distintos patrones, y es necesario definir cuáles son los que se podrán utilizar en la arquitectura. Se han definido los siguientes mecanismos de comunicación a utilizar:

- Envío *push* de datos: En este tipo de comunicación, se tiene uno o varios mensajes unidireccionales, desde un elemento a otro de la arquitectura. Por ejemplo, una notificación de la generación de un cierto evento en un servidor, que es notificado a los clientes conectados al mismo. Este tipo de comunicación se utiliza habitualmente para los mensajes que son relativamente fijos en el tiempo.
- Patrón de petición y respuesta: En este caso, se tiene el esquema clásico utilizado tradicionalmente por las arquitecturas Web, en el que un cliente

solicita un recurso determinado a un servidor y éste responde con el recurso u otra respuesta. Se trata de un patrón de comunicaciones fundamentalmente síncrono, por lo que todos los datos son ofrecidos únicamente en base a las peticiones que se hagan sobre ellos. Aunque es un patrón de comunicaciones diseñado para la Web, se puede trasladar con ciertas condiciones a un ecosistema IoT (por ejemplo, mediante la definición de puntos de acceso REST, como se verá más adelante).

- Esquema de suscripción y notificación: Para evitar la problemática que puede surgir de un esquema basado únicamente en peticiones y respuestas, se puede utilizar este esquema de suscripción y notificación. Se podría ver este esquema como una mezcla de los dos anteriores, en el que un cliente se suscribe a un recurso determinado, pero no espera una respuesta inmediata síncrona, sino que espera a recibir notificaciones cuando el servidor esté listo para enviarlas. Este esquema permite no bloquear la comunicación a la espera de la respuesta, por lo que puede ser interesante en actividades donde se esperen datos de muchos dispositivos en tiempo real.
- Esquema de suscripción y publicación: Este esquema es similar al anterior en cuanto a la forma de envío de la información: Uno o varios clientes se pueden suscribir a uno o varios canales de información, y recibir las publicaciones que éstos hagan en ellos. La diferencia es que en este esquema se difuminan los roles de cliente y servidor de información, teniéndose un elemento intermedio (usualmente denominado *broker*) que se encarga de gestionar las suscripciones y publicaciones entre los elementos de la comunicación.

En el ejemplo de la Figura 2.7, se observa un flujo de datos basado en el modelo de comunicaciones mediante publicación/suscripción: El servicio publica los datos y la aplicación suscrita a este servicio es capaz de leerlos. Un esquema similar pero basado en solicitudes y respuestas se puede ver en la Figura 2.8. En la Figura 2.9, sin embargo, se define el flujo de la información para ser almacenado en la plataforma y accedido posteriormente.

3. **Ciclo de vida de la información**

   La persistencia de la información recogida por la plataforma puede ser variable. Dependiendo del tipo de información, ésta puede no llegar a ser almacenada, o

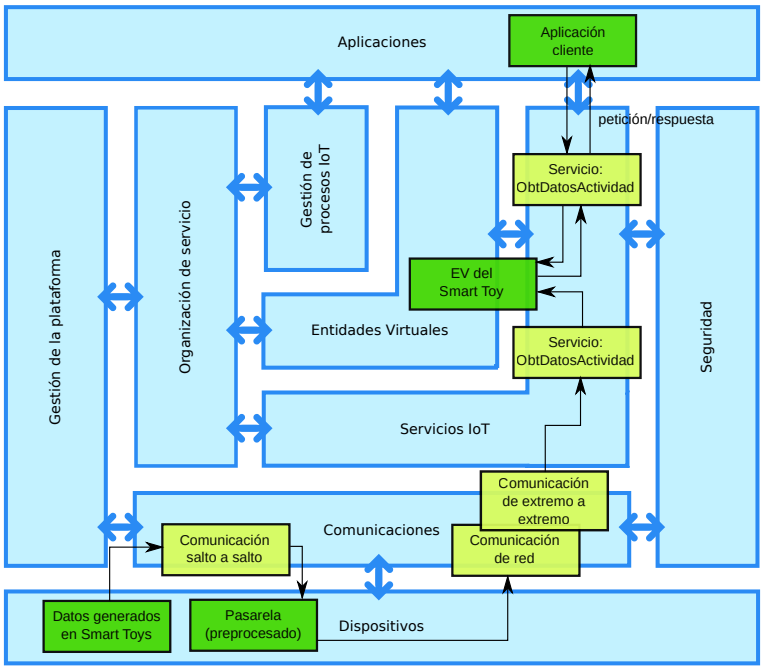

Figura 2.8: Flujo de información a través de grupos funcionales de la arquitectura (petición/respuesta).

ser almacenada brevemente (por ejemplo, datos puntuales de un sensor solicitados por algún cliente), o ser almacenada de forma persistente para obtener un histórico de datos o realizar un análisis general de los datos. Esta definición del tiempo que cada tipo de información debe permanecer en la plataforma permite además tener un control sobre los servicios y dispositivos activos en cada momento (se puede asegurar que servicios inactivos durante un periodo de tiempo dejarán de estar accesibles).

Se puede definir el ciclo de vida de la información a partir del diagrama de la Figura 2.10. Se identifican al menos tres posibles ciclos distintos, dependiendo de si los datos se almacenan en la plataforma de forma persistente o no. En el caso de que se almacenen, los datos pueden ser utilizados posteriormente y eventualmente borrados, o bien, pueden ser utilizados para componer nuevos servicios u obtener nueva información mediante la agregación de datos de distintos orígenes.

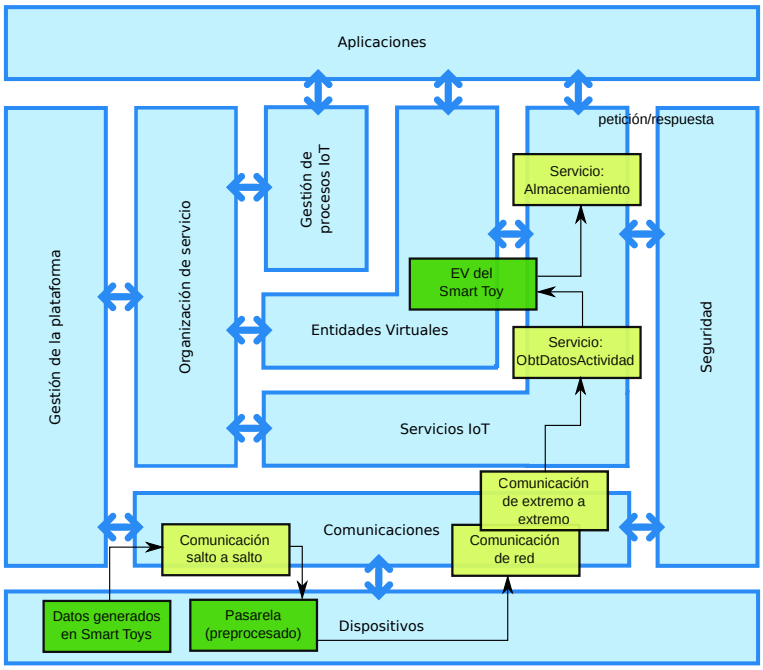

Figura 2.9: Flujo de información a través de grupos funcionales de la arquitectura (almacenamiento).

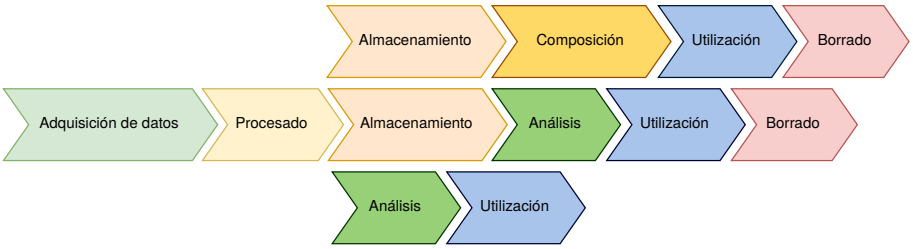

Figura 2.10: Ciclo de vida de la información en la plataforma.

2.3.2.3. Vista de Entidades Físicas

La vista de entidades físicas o vista física de la arquitectura tiene como objetivo el diseño de los elementos físicos que componen la misma, más allá de las funcionalidades o la gestión de la información que se haya definido anteriormente. Este tipo de vista es profundamente dependiente de la arquitectura concreta a diseñar, y está especificada en menor grado en la arquitectura de referencia IoT-A. En el caso que nos ocupa, y en base a los requisitos y restricciones planteados en la sección 2.3.1, se han determinado una serie de entidades físicas que compondrán los módulos de la plataforma basada en

la arquitectura.

En esta sección no se entrará en detalles de implementación sobre los dispositivos concretos a desarrollar, ya que la arquitectura se plantea como un sistema abstracto sobre el que se puedan utilizar diferentes tecnologías y parámetros (una descripción de la construcción de los prototipos se incluye más adelante en la sección 2.4.2). El objetivo de esto es definir cómo se debe desarrollar cada elemento físico, pero sin comprometer la interoperabilidad o, en la medida de lo posible, la introducción de nuevas tecnologías posteriormente. En esta sección se van a determinar estas entidades de forma genérica, y será en posteriores secciones donde se concretarán dispositivos y plataforma concretas utilizadas para evaluar la arquitectura mediante una implementación real.

Las entidades físicas que componen la arquitectura se han establecido en base a los componentes fundamentales definidos en los apartados anteriores. La idea básica de los módulos físicos es la simplicidad en su diseño, siempre implementando en ellos las funcionalidades definidas para la arquitectura completa. Así, se han determinado cuatro módulos físicos principales.

En la Figura 2.11 se pueden ver estos módulos y sus relaciones en cuanto a comunicaciones. A continuación, se define cada una de ellas:

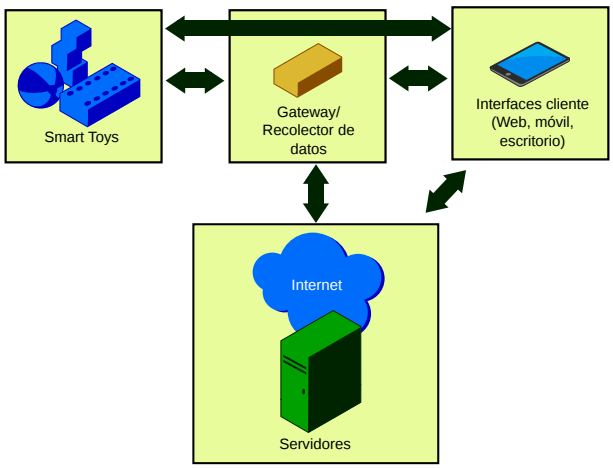

Figura 2.11: Diagrama básico de los módulos físicos que componen la Arquitectura de Juguetes Inteligentes.

1. **Smart Toys**: Son la prinicpal entidad física de la plataforma. Son dispositivos

diseñados específicamente para la plataforma dotados de capacidad sensorial y, en ocasiones, de actuadores para su interacción con el entorno en el que se sitúan. Dependiendo de su diseño, pueden ofrecer capacidades de procesado de los datos medidos, y en casos concretos pueden comunicarse entre sí. Se comunicarán con el resto de la plataforma a través de un gateway o, en ocasiones, directamente sobre clientes que puedan hacer las veces de gateway, dependiendo de la tecnología de comunicaciones utilizada.

2. **Gateway/Recolector de datos**: Este elemento se puede considerar parte de los conjuntos de Smart Toys. Permite mover parte de los recursos necesarios para comunicar los Smart Toys con el resto de la plataforma a un dispositivo externo sin las limitaciones en cuanto a recursos de los primeros, como dar formato a los datos, proveer de un mayor grado de seguridad, y sobre todo, permitir conectar las redes locales de Smart Toys a la plataforma a través de Internet.

   Estos dispositivos además permiten la utilización de los Smart Toys de forma autónoma, sin una conexión a Internet en tiempo real. En este sentido, los dispositivos actúan como recolectores de datos, con cierta capacidad de almacenamiento temporal de los datos obtenidos, que son volcados posteriormente sobre la plataforma a través de Internet.

3. **Cliente e interfaz de usuario**: La utilización de los Smart Toys por parte de sus usuarios habituales (niños) es transparente a la plataforma, pero para la configuración de los dispositivos, la gestión de las actividades realizadas, y la obtención de información de la plataforma, se utilizan aplicaciones cliente que ofrecen interfaces de usuario para los expertos. Estas interfaces se conectan normalmente a los gateways a través de una red local, y a los servidores de la plataforma a través de Internet. En ocasiones, pueden comunicarse directamente con los Smart Toys.

4. **Servidores en Internet**: Los servicios ofrecidos por la plataforma, que permiten la consulta, análisis y almacenamiento de los datos de los Smart Toys, así como los datos relacionados con éstos, se localizan en una serie de servidores desplegados de forma que sean accesibles a través de Internet.

2.3.2.4. Vista de Contexto

La vista de contexto es aquella que describe las relaciones, dependencias e interacciones del sistema con respecto a su entorno. Es decir, describe la relación entre el sistema y el mundo real en el que se utiliza.

En la Figura 2.12 se muestran las relaciones de la plataforma con su contexto. En la arquitectura diseñada, el entorno donde se ejecuta la plataforma se define en base principalmente a las personas y sistemas relacionados con éste. Así, en la plataforma se tiene una especial relación con los principales usuarios de los Smart Toys, que serían niños que interactúan con ésta en base a diversas actividades de juego. La realimentación que los niños pueden recibir por parte de la plataforma siempre será en forma de luces, sonidos u otros interfaces simples que le permitan por ejemplo modificar su comportamiento o premiar su rendimiento.

Por otro lado, los expertos son también uno de los principales actores que conforman el contexto de la plataforma. Los expertos se relacionan con ésta mediante la obtención y análisis de la información de los Smart Toys, y además incorporan a la misma sus consejos, diagnósticos, etc. Además, pueden incorporar nuevas actividades o formas de juego para la obtención de nuevos datos a incorporar al análisis.

En el entorno de la plataforma también se deben tener en cuenta tanto los padres como los profesores de los niños que la utilicen. En ambos casos, pueden proveer datos sobre los niños (datos personales, historial de comportamiento, etc.) para apoyar el análisis de los expertos. A cambio, de la plataforma pueden obtener realimentación a través de consejos, diagnósticos, alertas, y cualquier otra información determinada sobre los niños a su cargo.

De forma menos central, también se relacionan con la plataforma otro tipo de personas, “técnicos”, que se encargarán de la gestión técnica de la misma.

Por último, se define la posibilidad de interacción de la plataforma con otras plataformas o sistemas de salud que puedan ofrecer información adicional sobre los niños que la utilicen.

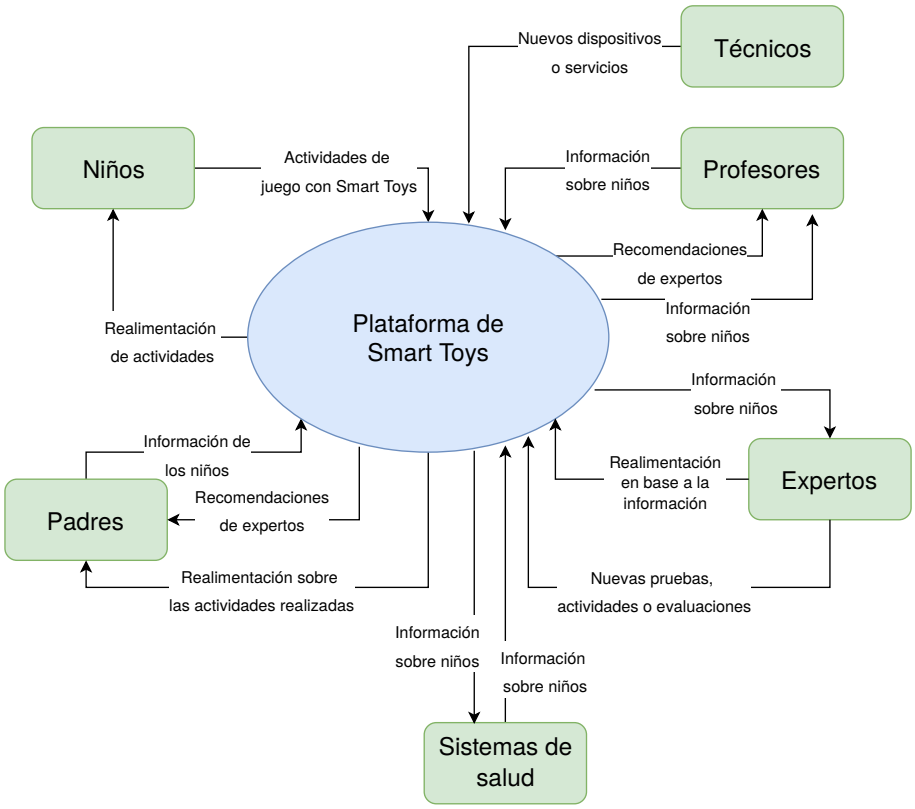

Figura 2.12: Diagrama de contexto que muestra las relaciones entre el sistema y su entorno.

2.4. Caso de uso y prototipos de Smart Toys

El objetivo principal del diseño arquitectónico llevado a cabo en las secciones anteriores es contar con herramientas que permitan el desarrollo de los dispositivos físicos para utilizar luego en un caso de uso real. Este caso de uso se basa en el proyecto de investigación en el que se enmarca la Tesis.

En esta sección se realiza una descripción del proyecto de investigación y sus objetivos, y a continuación se describen cada uno de los prototipos implementados para usarse en pruebas de validación que se han llevado a cabo en el marco del proyecto.

2.4.1. El proyecto EDUCERE

El proyecto EDUCERE (Ecosistema de Detección Ubicua, atenCión y Estimulación tempRana para niños con trastornos del dEsarrollo) [31] tiene como objetivo investigar, desarrollar y evaluar soluciones innovadoras para la sociedad que, mediante la interacción natural del niño con los juguetes, los objetos cotidianos y con otras personas, permitan detectar alteraciones en el desarrollo y realizar actividades de estimulación y atención temprana, en entornos reales como el hogar y la escuela.

Para ello, el proyecto ha contado con un equipo investigador multidisciplinar a través del cual se han llevado a cabo tareas para identificar cómo las nuevas tecnologías pueden ofrecer soluciones para dar apoyo en la detección precoz de trastornos en el desarrollo que pueden ir desde dificultades en el desarrollo cognitivo, en el desarrollo del lenguaje, en el desarrollo motor o sensorial.

Apoyándose en estas posibles soluciones tecnológicas, el proyecto pretende promover la salud integral del niño mediante el fortalecimiento de los procesos de prevención y atención temprana con valor añadido para su bienestar físico, mental y social.

Una de las metas del proyecto es precisamente el desarrollo e implementación de un prototipo de la plataforma de Smart Toys objeto de esta Tesis, y a partir de ella, la realización de pruebas para la validación y para la obtención de información útil para el objetivo final planteado.

2.4.2. Diseño de los prototipos

En el marco del proyecto EDUCERE, se han diseñado, implementado y probado una serie de prototipos de Smart Toys. En esta sección se describe cada uno de ellos con detalle, incluyendo las decisiones en cuanto a tecnologías concretas utilizadas en cada caso.

Una de las primeras tareas que se han llevado a cabo antes de comenzar el diseño específico de los prototipos ha sido la evaluación, en una serie de reuniones con los expertos participantes en el proyecto acerca de qué actividades serían algunas de las

principales a realizar con los niños, y qué herramientas serían necesarias para llevarlas a cabo.

2.4.2.1. Smart Cubes

Utilizando como base las diversas escalas existentes en psicología para la evaluación del desarrollo infantil (véase por ejemplo Merrill-Pallmer [14] o Bayley [13]), se ha determinado que un punto de partida interesante para la evaluación de los juguetes podría ser el desarrollo de una serie de cubos apilables, ya que son la base de numerosas actividades definidas en las escalas, y por tanto, favorece la posibilidad de utilizar el conocimiento de los expertos en su diseño, además de facilitar la experimentación, al ofrecer un sistema comparativo sobre el que hacer las pruebas.

Estos cubos, hechos habitualmente de madera y con unas dimensiones de una pulgada por lado, permiten realizar distintas actividades en presencia de un experto, que se encarga de medir y evaluar, de forma manual y mediante su propio criterio, el comportamiento del niño.

A partir de la selección del juguete, se ha planteado cómo dotar a éste de capacidades sensoras y de comunicación. El juguete diseñado se ha denominado cubo inteligente o Smart Cube.

Existen dos principales restricciones en el hardware del juguete, que serán determinantes en su diseño, su tamaño y su peso. Los cubos utilizados actualmente por los expertos miden exactamente una pulgada por cada lado, y al estar fabricados en madera, tienen un peso bastante ligero. Cuanto más se parezcan los cubos sensorizados a los cubos originales, más fiables serán las comparativas a realizar entre ellos posteriormente. Se ha determinado que la dimensión de cada lado del cubo no debería exceder los 2,5 a 3 cm. para no variar sustancialmente la actividad a realizar. Es importante por ejemplo que el agarre del cubo por parte del niño se pueda realizar con una única mano y sea capaz de levantarlo con ésta.

Además de estas restricciones, el prototipo debe cumplir las restricciones generales propuestas para los elementos de la arquitectura, y que se han detallado anteriormente.

En este caso deben al menos:

- Ofrecer información relevante sobre su uso: Al menos información sobre su movimiento y, finalmente sobre la forma en que se ha realizado la actividad.
- Ofrecer un alto grado de autonomía, ya que debe ser usado a lo largo del día sin necesidad de recarga: Para esta actividad, se ha determinado que al menos se deberían poder utilizar cuatro horas sin recargar la batería.
- El coste de fabricación debe ser lo más ajustado posible: Como con todos los elementos de la arquitectura en los que sea posible, se deben utilizar tecnologías de bajo coste y a ser posible abiertas.
- Tanto el diseño software como el hardware debe tener en cuenta su reutilización en futuros diseños: Teniendo en cuenta que las restricciones en cuanto a forma y peso pueden variar en otros dispositivos, se deben utilizar elementos y diseños que permitan la reutilización del diseño, con el objetivo de mejorar la interoperabilidad y ahorrar costes en el desarrollo de nuevos juguetes o actividades.

En base a lo anterior y teniendo en cuenta restricción en el tamaño de cada dispositivo, se debe realizar un diseño en el que se incluyan al menos los siguientes componentes:

- Un microcontrolador para el procesado y control de los sensores, así como para la implementación de los sistemas de seguridad y comunicaciones.
- Una batería y un circuito de protección para evitar fallos en los procesos de carga y descarga.
- Sensores: Al menos un acelerómetrogiróscopo para obtener datos del movimiento. Además, se incluyen una serie de sensores de luminosidad (*Light Dependant Resistors, LDR*) para apoyar la información del acelerómetro, indicando qué caras del cubo reciben más o menos luz.
- Un sistema de encendido y apagado, para prolongar la autonomía del dispositivo mientras no se está utilizando.
- Un sistema de comunicaciones para enviar los datos del sensor.

- Un sistema básico de actuadores para la interacción con el usuario (niño): En base a LEDs y un zumbador.

Como microcontrolador se ha seleccionado un ATMega328p [90], fabricado por la compañía ATMEL (Atmel Corporation (San Jose, CA, USA)). Este microcontrolador reúne unas características suficientes para gestionar las tareas que deben realizar los juguetes (recopilación y transmisión de datos, gestión de parte de las actividades, tareas de cifrado, etc.), y la sencillez de programación que supone la compatibilidad con plataformas como la ofrecida por Arduino. Puede funcionar a 16 y 8 MHz lo cual le garantiza una velocidad más que suficiente para ser utilizado en un Smart Toy, y tiene suficientes puertos de entrada y salida para la conexión de los sensores y el sistema de comunicaciones.

Sobre la batería, se ha decidido seleccionar la mayor posible cuyo tamaño le permita ser colocada dentro de cada cubo (teniendo en cuenta que el resto de componentes también deben estar colocados en el interior de cada cubo. Se ha seleccionado una batería de Polímero de Ión Lítio (LiPo) de 3.7V, al ofrecer una buena ratio entre su tamaño y su capacidad. La mayor batería de estas características que se ha podido incluir en el interior del cubo tiene una capacidad de 150mAh, que se ha demostrado más que suficiente para proporcionar la autonomía necesaria a cada dispositivo. Las baterías de tipo LiPo no deben descargarse nunca por debajo de aproximadamente 3.2V, por lo que se ha añadido al diseño un pequeño circuito de protección que impide que esto suceda, desconectando el dispositivo cuando se alcanzan valores bajos de batería (el umbral de parada se ha establecido en 3.15V). Este circuito se compone de un integrado MCP112 [91] (Microchip Technology Inc. (Chandler, AZ, USA)) que se encarga de medir el voltaje proporcionado por la batería, y un integrado AP2281 (Diodes Incorporated (Plano, TX, USA)) que se encarga de conectar y desconectar el circuito dependiendo de si el voltaje medido está por encima o por debajo del umbral.

Como mecanismo de encendido y apagado, y teniendo de nuevo en cuenta las reducidas dimensiones del dispositivo, se ha optado por no utilizar elementos mecánicos, sino dotar al circuito de un relé Reed. Este tipo de interruptor se activa en base a la aplicación de un campo magnético cercano y por tanto se puede utilizar como interruptor posicionando un pequeño imán sobre el dispositivo.

En cuanto al elemento utilizado para las comunicaciones, se ha determinado utilizar un emisorreceptor de radiofrecuencia NRF24L01+ [92] (Nordic Semiconductor ASA (Oslo, Norway)). En la Tabla 2.1 se puede ver una comparativa de distintas tecnologías que pueden ser utilizadas para el tipo de comunicación de corto alcance requerido en este caso.

Tabla 2.1: Comparativa de distintas tecnologías a utilizar en las comunicaciones de los prototipos.

Tecnología	**Bluetooth**	**BLE**	**Wi-Fi**	**6LoWPAN**	**NFC**	**Zigbee**	**NRF24**
Consumo	Bajo (<30 mA)	Muy Bajo (<15 mA)	Medio-Alto (>20 mA)	Bajo (<30 mA)	Muy Bajo (<15 mA)	Bajo (<20 mA)	Muy Bajo (<15 mA)
Cobertura	50/100 m	50 m	100 m	30 m	0.2 m	100 m	30 m
Estándar	802.15.1 (inicialmente)	802.15.1 (inicialmente)	802.11	802.15.4	ISO/IEC 18092 ISO/IEC 14443-2,3,4	802.15.4	-
Adopción del mercado	Muy Alta	Alta	Muy Alta	Baja	Alta	Alta en entornos industriales	Baja
Tasa máxima	24 Mbps	1 Mbps / 2 Mbps	300 Mbps	250 Kbps	424 Kbps	250 Kbps	1 Mbps / 2 Mbps

Como se puede ver en la Tabla, las comunicaciones basadas en Bluetooth [93] y la más actual Bluetooth Low Energy (BLE) son las más populares para la comunicación de dispositivos en redes locales o redes PAN ("Personal Area Networks"). BLE es una variante de Bluetooth que permite la conexión de dispositivos con un menor consumo pero con una menor tasa de transmisión. Wi-Fi [94] por su parte es una de las tecnologías más utilizadas para la creación de redes inalámbricas locales, aunque en realidad hace referencia a un conjunto de protocolos agrupados en el estándar 802.11 de IEEE. Tiene un rango de cobertura mayor que el resto, pero a cambio de un mayor consumo energético. 6LoWPAN [95] es una propuesta de uso de IPv6 para IoT y redes de sensores. NFC ("Near-Field Communication") want2011near tiene como objetivo la comunicación de dispositivos situados a muy corta distancia, por lo que no cubre todos los requisitos de comunicación de nuestro caso de uso. Zigbee [96] es una alternativa a 6LoWPAN y Bluetooth. Es muy utilizado en entornos industriales.

La selección de un dispositivo de comunicaciones basado en NRF24 se basa en la posibilidad de tener unas características similares a BLE en cuanto a consumo, cobertura y tasa de transmisión, pero con un coste menor. La decisión de utilizar este sistema en lugar de otros más estandarizados para comunicaciones de dispositivos IoT debido fundamentalmente al ahorro en cuanto a coste en el momento del diseño, así como a la facilidad de integración software con el microcontrolador ATMega328p. En

cualquier caso, siempre se ha tenido en cuenta que, en el diseño de futuros dispositivos, sea posible modificar el sistema de comunicaciones para utilizar por ejemplo un adaptador BLE o Bluetooth si se necesitara una mayor tasa de transferencia de datos.

Este dispositivo de radiofrecuencia implementa hasta un nivel de enlace, incluyendo gestión automática de ACKs y retransmisión de tramas perdidas, con una tasa de transmisión máxima de 2Mbps. Este esquema de comunicaciones sencillo es suficiente para implementar sobre él un protocolo simple de comunicaciones entre dispositivos y gateways o recolectores.

La programación del dispositivo se lleva a cabo a través del interfaz ICSP. Esto implica que es necesario utilizar un programador hardware externo para cargar nuevos programas en el dispositivo, pero esto no es un gran inconveniente tratándose de un prototipo que, una vez programado podrá ser utilizado experimentalmente sin necesidad de volver a ser programado. Además, el uso de un programador permite ahorrar recursos en cuanto a memoria (no es necesario cargar un *bootloader*) y energía en el dispositivo.

Como sensores se han seleccionado un acelerómetro de 9 ejes MPU-9150 [97] (InvenSense (Sunnyvale, CA, USA)) y una serie de sensores de luz LDR NORPS-12 [98] (Silonex Inc. (Montreal, QC, Canada)). Estos últimos se conectan en serie formando un divisor de voltaje que a su vez se conecta a los puertos analógicos del microcontrolador, mientras que el acelerómetro se conecta a través del bus I2C. El sensor MPU-9150 está en realidad compuesto por un acelerómetro, un giróscopo y un magnetómetro, todos ellos de 3 ejes (se trata en realidad de un dispositivo *System in Package (SiP)* que contiene un acelerómetro/giróscopo MPU-6050 [99], un magnetómetro y además un procesador integrado que permite realizar cierto preprocesado de los datos en el propio sensor. Aunque el sensor de 6 ejes puede ser suficiente para una mayoría de las mediciones, se ha decidido utilizar la versión que incluye el magnetómetro para disponer de la mayor cantidad y diversidad de información en el dispositivo.

El circuito integrado donde se coloca el microcontrolador y que provee al dispositivo de las conexiones con el resto de componentes se ha diseñado de forma que deje el mayor espacio posible para éstos. Para ello, se ha realizado el circuito PCB en 6 pequeñas piezas cuadradas que, una vez ensambladas, forman la propia figura del cubo. Cada

tres piezas se conectan primero entre sí mediante la soldadura de las pistas y mediante conectores, se pueden a su vez ensamblar las dos piezas resultantes para formar el cubo completo. Una vez ensamblado, el circuito integrado mide 2.54 cm de lado, dentro de los márgenes impuestos por los requisitos del prototipado. Además, el hueco interior que deja el circuito permite que los sensores y el resto de componentes puedan ser soldados y se alojen en él.

Mediante impresión 3D se ha diseñado finalmente una carcasa formada a su vez por dos piezas de plástico que recubre el circuito y permite que sea utilizado sin ningún peligro.

En la Figura 2.13-a se pueden ver las dos piezas formadas por tres partes del PCB. En la Figura 2.13-b se muestra el cubo ensamblado, y finalmente, en la Figura 2.13-c se puede ver la carcasa que lo recubre.

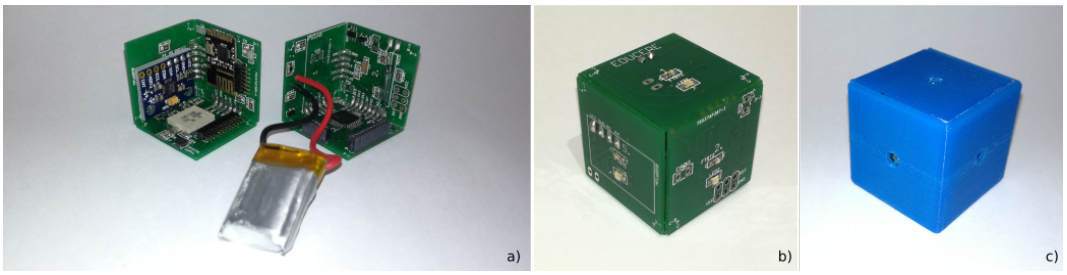

Figura 2.13: Fotografías del prototipo de Smart Cube. Interior del PCB (a), cubo ensamblado (b) y carcasa impresa en 3D (c).

Para comprobar que el diseño cumple el requisito de autonomía, se han realizado tests con dos configuraciones distintas del reloj del microcontrolador: 1MHz y 8MHz. Los resultados pueden verse en la gráfica de la Figura 2.14

En el eje x de la gráfica se muestra el tiempo medido durante la descarga en minutos, mientras que el eje y representa el voltaje medido en la batería en cada momento. Según las mediciones, se puede observar que utilizando el reloj de 8 MHz se obtienen aproximadamente 650 minutos de autonomía (unas 10 horas y 30 minutos). Si se rebaja la frecuencia de reloj a 1 Mhz, se pueden llegar a 960 minutos de autonomía. Hay que tener en cuenta que estas mediciones se han hecho simulando el funcionamiento continuo de los cubos, incluyendo la lectura de datos en los sensores y su comunicación a través del emisor de radiofrecuencia, por lo que, en un escenario real, donde no se

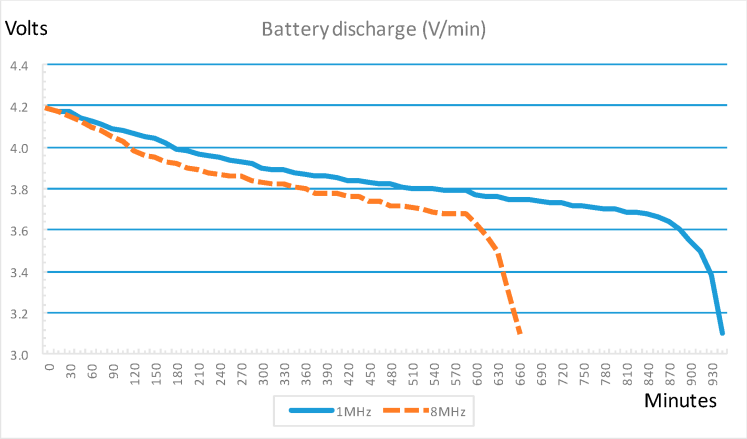

Figura 2.14: Gráfica mostrando la evolución de la carga de la batería incorporada en los prototipos de Smart Cube.

producen lecturas y emisiones continuas de datos, el resultado en cuanto a autonomía será siempre mayor. Esta prueba certifica que se cumple también la restricción en cuanto autonomía de cada dispositivo.

2.4.2.2. Tablero de espigas

A partir de este primer prototipo, se han añadido nuevos diseños al sistema, con el fin de ampliar las actividades disponibles para la experimentación a la vez que determinar la interoperabilidad entre dispositivos. Inicialmente, además de los Smart Cubes, se ha determinado la creación de una tabla de espigas, una pelota y un sonajero. Los dos últimos diseños no han sido aún objeto de una implementación, pero sí se ha construido un prototipo del primer juguete.

Las tablas de espigas son una herramienta muy común en la evaluación psicológica [100]. Consiste en una tabla, generalmente de madera en la que se tienen una serie de hileras de agujeros sobre los que se pueden introducir un conjunto de espigas. Se utiliza en evaluaciones del desarrollo de la destreza manual [101] y muchas otras evaluaciones, como por ejemplo la predicción de posibles lesiones cerebrales [102].

La tabla prototipo diseñada tiene como objetivo poder ser utilizada con las tablas tradicionales, por lo que se mantienen las restricciones en cuanto a dimensiones y

forma: Debe medir menos de 31cm de longitud y debe estar compuesta al menos por dos filas de agujeros. Cada espiga y cada agujero debe tener aproximadamente 10mm de diámetro, y la distancia entre agujeros debe ser suficiente para que los niños puedan recoger y soltar cómodamente las espigas.

Internamente, el Smart Toy se ha diseñado utilizando tecnologías similares a las de los Smart Cubes: Se ha incluido un microcontrolador ATmega328p, y un emisor/receptor NRF24L01+, por lo que ambos Smart Toys serán completamente compatibles desde el punto de vista hardware. En este caso, se ha dotado al dispositivo de una batería de 12000 mAh, dado que el tamaño de este dispositvo permite baterías más grandes.

Para la detección de la inserción o retirada de una espiga de cada agujero, se han utilizado sensores fotointerruptores LTH-301-32 [103] (Lite-On Inc. (California, USA)). En el interior de cada agujero se ha añadido un sensor de este tipo, de forma que emitan una señal si el haz de luz que emiten se interrumpe cuando una espiga se introduce en el agujero. Los sensores se han conectado en serie y después se han utilizado registros de desplazamiento de 8 bits (CD4021B [104] (Texas Instruments (Dallas/Texas/USA))) para conectarlos al microcontrolador, de forma que se minimice el número de pins necesarios para su conexión, generando así un mapa de bits con el estado de la tabla en cada instante.

Además, como interfaz para los niños que utilicen el juguete, se han añadido una serie de LEDs RGB asociados a cada agujero de la tabla, que puede ofrecer información visual sobre el estado y permitir la definición de nuevas actividades, como la introducción de clavijas siguiendo patrones de colores, etc.

En la Figura 2.15 se puede ver el interior de la tabla prototipo.

2.4.2.3. Gateway/Recolector de datos

Se ha diseñado un módulo gateway para los juguetes antes definidos con el objetivo de ser el punto centralizado de conexión con los juguetes. Este elemento es imprescindible para separar las comunicaciones locales de los Smart Toys de las comunicaciones

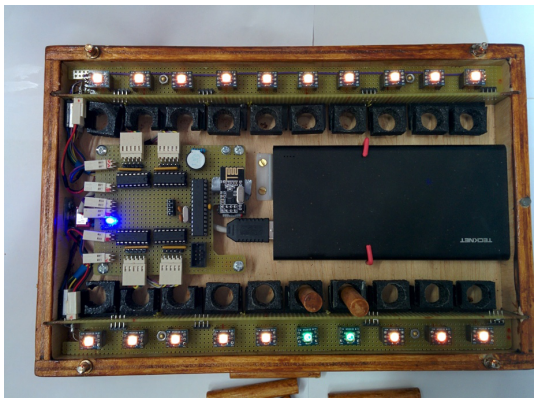

Figura 2.15: Interior de la tabla de espigas prototipo.

remotas a través de Internet. Además, dado que es posible que en muchos escenarios no exista la posibilidad de una conexión directa a Internet en tiempo real, el gateway se ha diseñado para servir como dispositivo de almacenamiento temporal seguro de los datos generados, que pueden ser volcados posteriormente en la plataforma.

La conectividad directa de los dispositivos a Internet no se ha considerado como una opción viable debido a las restricciones en cuanto a recursos disponibles en los mismos (consumo de energía, memoria y capacidad de procesamiento necesarios, etc.), así como debido a la imposibilidad de asegurar que en todos los entornos donde se va a llevar a cabo la experimentación se disponga de conectividad directa a Internet (Wi-Fi, 4G, etc.).

En la Figura 2.16 se puede observar el diseño interno del recolector. Se trata de un dispositivo basado en una placa Raspbery Pi 3 B [105] (aunque dada su naturaleza, podría ser portado de forma relativamente sencilla a otras plataformas de similares características) que contiene un sistema operativo Raspbian.

Como se puede ver en la figura, el dispositivo tiene varias interfaces de entrada y salida: Una de ellas se define para la comunicación con los dispositivos físicos (juguetes) a través del sistema de radiofrecuencia o Bluetooth que utilicen. Por otro lado, el manejo de este dispositivo se ha preparado para ser lo más flexible posible, cumpliendo los requisitos definidos durante el diseño de la arquitectura. En este sentido, se ha definido tanto una API REST HTTP (“HyperText Transfer Protocol”) para el manejo a través de aplicaciones externas que utilicen comunicaciones del tipo peti-

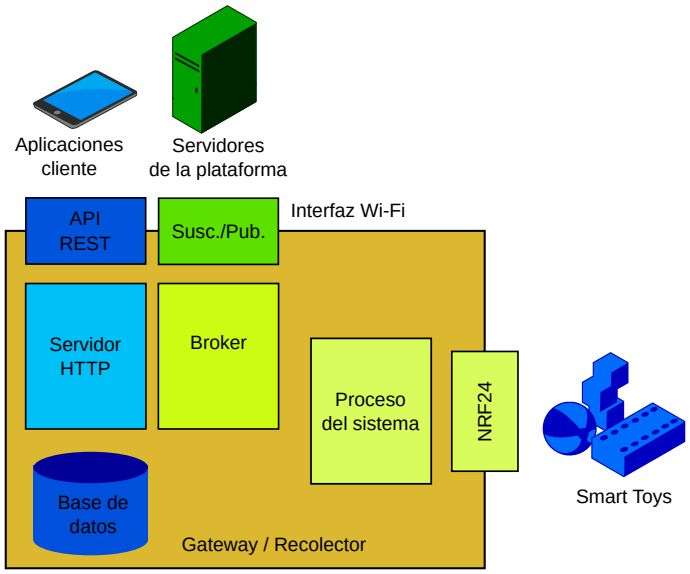

Figura 2.16: Diagrama del diseño del gateway y recolector de datos prototipo.

ción/respuesta. Por otro lado, y en base a ofrecer la posibilidad de utilizar patrones de comunicación de subscripción/publicación y notificación/publicación, se ha utilizado un servidor RabbitMQ [106] que ofrece una interfaz basada en broker AMQP. Alternativamente, también se ha propuesto la utilización de un broker Mosquitto [107] basado en MQTT, que será finalmente el utiliado en la implementación del sistema de control de acceso descrito en el capítulo 3. Este servicio se utiliza igualmente para la comunicación interna en el recolector, como se verá más adelante. Finalmente, se ofrece una interfaz Web integrada en el dispositivo, para poder ser utilizado sin necesidad de aplicaciones externas, a través de un navegador Web.

Los componentes a nivel estructural de este prototipo son:

- Servidor HTTP: Se trata de un servidor Web implementado en el lenguaje de programación Python a través del framework de desarrollo Django [108]. En este servidor se ha incluido tanto la definición de las APIs de manejo externo como la aplicación Web integrada (que cuenta con menos funcionalidades en el prototipo).

  Las APIs se han definido siguiendo la filosofía REST, por lo que se han determinado una serie de puntos de acceso o recursos sobre los que se permite la realización de operaciones. Estos recursos son:

  - Punto de acceso de experimentos: Ofrece todas aquellas operaciones que permiten la gestión de los experimentos a realizar con la plataforma de forma local, incluyendo el inicio, fin, configuración, etc.
  - Punto de acceso de datos: En este recurso se ofrecen las operaciones relacionadas con la gestión de datos generados durante los experimentos. Se permite el envío de datos a la plataforma, su eliminación, etc.
  - Punto de acceso de usuarios: Se trata principalmente de un punto de acceso para la gestión y configuración de los datos de los niños que utilicen los dispositivos. En este punto es conveniente recordar que nunca datos personales de los niños salen de la plataforma Cloud, asignándose identificadores numéricos para la realización de los experimentos.
  - Punto de acceso de actividades: En este punto de acceso se pueden llevar a cabo las operaciones de configuración de actividades. A través de él se pueden añadir nuevas actividades (incluyendo nuevas representaciones de datos dependiendo de los dispositivos a utilizar y el tipo de datos a almacenar en cada caso).

- Broker: En este servidor se ofrece un entorno para la suscripción de dispositivos a partir de un mecanismo de intercambio de mensajes centralizado a través de un Broker y el uso del protocolo AMQP o MQTT. Este sistema tiene una doble función. Por un lado, sirve para la comunicación de los distintos componentes del sistema dentro del gateway. Por otro, ofrece una interfaz de comunicaciones para clientes que utilicen el esquema de publicación/suscripción de datos. Se ofrecen colas (en AMQP) o "topics" (en MQTT) para realizar operaciones similares a las ofrecidas a través de la API REST.

- Proceso del sistema: Es un programa funcionando en modo demonio, que se compone de varios hilos de ejecución. En uno de los hilos, se encarga de la monitorización de la interfaz de comunicaciones de radiofrecuencia (NRF24), y del envío y recepción de mensajes a través de ella. Además, otro hilo se encarga del almacenamiento de los datos recibidos y un tercer hilo se ocupa de recibir mensajes desde los servidores de comunicaciones (HTTP, AMQP o MQTT) para gestionar el dispositivo.

- Base de datos: Utilizada para almacenar tanto los datos de configuración y gestión

del propio dispositivo como los datos que se retienen temporalmente recibidos de los sensores. Se basa en un modelo SQL en el que se diferencian datos de niños, datos de expertos y datos de actividades.

2.4.2.4. Clientes e Interfaces de usuario

Durante el desarrollo de los prototipos, se han utilizado múltiples sistemas para la gestión de los dispositivos. Inicialmente, se ha utilizado un cliente genérico REST para la generación de peticiones y respuestas que permitieran una rápida evaluación de las implementaciones llevadas a cabo en los Smart Toys. Sin embargo, posteriormente, en colaboración con el resto del equipo investigador del proyecto, se han definido interfaces de usuario que pudieran permitir acceder a las interfaces del gateway y el servidor de almacenamiento, con una mayor facilidad de uso por parte de los expertos en desarrollo.

Así, se han desarrollado dos interfaces cliente prototipo para la plataforma. Por un lado, se ha desarrollado una aplicación cliente para dispositivos Android utilizando el sistema Cordova [109], lo cual permitirá su sencilla exportación a otros entornos móviles. En la Figura 2.17 se puede ver esta interfaz funcionando sobre una tablet.

Figura 2.17: Interfaz de usuario de la plataforma sobre tablet Android.

Por otro lado, para poder utilizar los prototipos desde cualquier otro dispositivo, se ha diseñado una interfaz Web sencilla con la que manejar los aspectos básicos del sistema. Esta segunda interfaz se puede ver en la Figura 2.18.

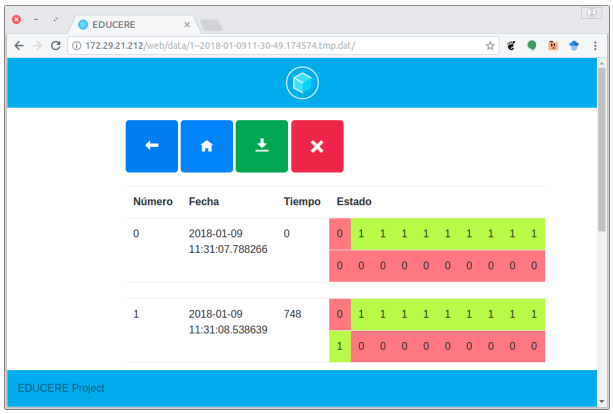

Figura 2.18: Interfaz Web de la plataforma.

2.4.2.5. Servidor de almacenamiento y análisis

En la versión prototipo de la plataforma se ha diseñado y desarrollado una versión más simple y reducida del sistema de servidores planteado en el diseño, basado en un entorno Web accesible desde Internet y desde los dispositivos cliente. La implementación específica del prototipo de este servidor, en cualquier caso, queda fuera del trabajo realizado dentro de esta Tesis, al haberse realizado por parte de otros investigadores del proyecto EDUCERE.

2.4.3. Preprocesado de datos en los Smart Toys

Para la obtención de datos relevantes de los sensores embebidos en los juguetes se ha optado por dos vertientes distintas: Por un lado, se ha propuesto el envío de los datos obtenidos 'en crudo', para ser posteriormente analizados *offline* en la plataforma. Este sistema permite una alta flexibilidad en el uso de los datos, ya que se almacenan todas las mediciones originales. Sin embargo, se ha explorado paralelamente la posibilidad de preprocesar los datos obtenidos en el propio dispositivo. De esta forma, es posible obtener directamente datos de alto nivel directamente del dispositivo, lo cual simplifica el diseño de los servicios de análisis y procesado posterior de datos, y permite determinar la capacidad de los sensores para dar información relevante.

A partir de los cubos diseñados, se ha determinado ofrecer desde cada uno de

ellos datos acumulados de movimientos completos (es decir, desde que se recoge de la superficie donde reposa hasta que se vuelve a parar). Concretamente, se determina para cada movimiento:

- Aceleración media.
- Aceleración máxima.
- Velocidad media.
- Velocidad máxima.
- Valores de agitación.

Para el cálculo de los valores de aceleración, se pueden utilizar directamente las lecturas ofrecidas por el MPU-9150. Etas mediciones se obtienen para cada uno de los ejes X, Y y Z, por lo que es necesario obtener un único valor que determine el valor medio o máximo durante un periodo de tiempo. El primer obstáculo a salvar es el valor de la gravedad (G, es decir, 9,8 m/s^2). Este valor está siempre sumado a alguno o algunos de los ejes, dependiendo del posicionamiento del cubo. En el caso más sencillo, si por ejemplo el eje Z es perpendicular a la superficie y el cubo está en un estado de reposo, se tiene G como un vector perpendicular al centro terrestre. En ese caso, simplemente se debe restar el valor de G del eje Z (teniendo en cuenta que el valor puede ser positivo o negativo dependiendo de la posición del eje). Sin embargo, cuando el cubo se gira, el vector G se convierte en una combinación del resto de ejes, tal y como se puede ver en la Figura 2.19.

Para poder eliminar este vector, en cualquier caso, es necesario conocer la posición del acelerómetro y cómo se orientan los ejes X, Y y Z. Estos valores se pueden obtener a través del magnetómetro incluido en el sensor. Esto se debe a que el magnetómetro mantiene una referencia fija hacia el norte magnético en su eje Xm, y una referencia fija hacia el centro de la Tierra en el eje Zm. El eje Ym estará localizado entonces a $\pi/2$ radianes de los otros dos, tal y como se muestra en la Figura 2.20.

Por tanto, es posible determinar la posición de los ejes X, Y y Z en base a los valores obtenidos por el propio sensor. Para ello, se pueden utilizar los ángulos de giro

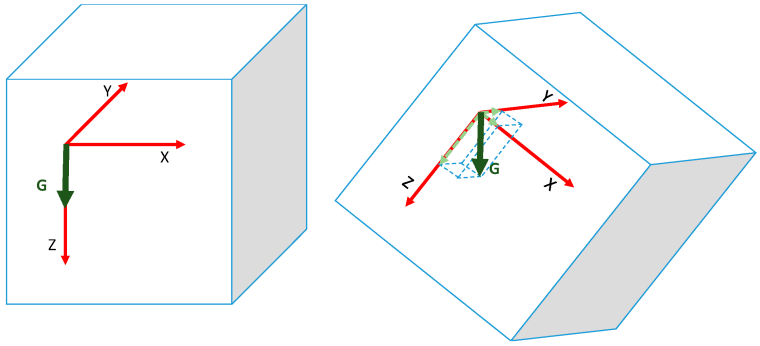

Figura 2.19: Diagrama que muestra el vector de la gravedad sobre los ejes de aceleración del sensor en un Smart Cube cuando un eje es perpendicular a la gravedad (a) y cuando no es perpendicular a la gravedad (b).

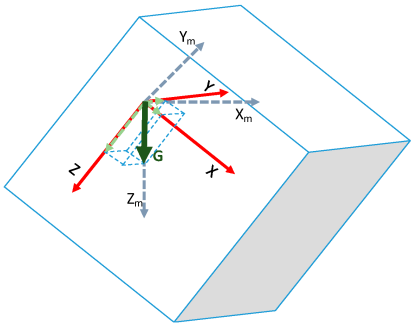

Figura 2.20: Localización del vector de gravedad en relación a los ejes X, Y y Z medidos por el sensor.

Eulerianos, concretamente con los llamados ángulos Tait-Bryan, una variante de estos ángulos utilizados principalmente en la navegación aeroespacial [110]. Estos ángulos habitualmente se conocen como *Yaw*, *Pitch* y *Roll* (traducidos a veces como Guiñada, Alabeo y Cabeceo).

Dado que el procesador incorporado en el sensor MPU-9150 permite obtener directamente estos ángulos (son preprocesados internamente a partir de los datos del giróscopo), es posible obtener la posición de los ejes eliminando el componente gravitatorio aplicando las siguientes ecuaciones:

$$nAX = AX + sin(\phi) \tag{2.1}$$

$$nAY = AY - cos(\theta) * sin(\phi) \tag{2.2}$$

$$nAZ = AZ - cos(\theta) * cos(\phi) \tag{2.3}$$

Donde nAX, nAY y nAZ son los valores para cada eje sin tener en cuenta la variación producida por G. AX, AY y AZ son los valores originales medidos en el acelerómetro para cada uno de los ejes. Finalmente, ψ, ϕ y θ son los ángulos de Tait-Bryan.

Además de lo anterior, hay que tener en cuenta el error propio del sensor a la hora de calcular valores precisos de aceleración y especialmente a la hora de calcular la velocidad a partir de estos valores, ya que, incluso para valores muy pequeños de aceleración, el error introducido en el cálculo hace que los valores de velocidad no sean correctos. Para evitar esto, se lleva a cabo una calibración inicial del dispositivo, mediante la medición de un número fijo de valores del sensor y la posterior obtención del valor medio, con el dispositivo en estado de reposo. Más adelante, se resta este valor medio a las mediciones tomadas con el dispositivo en movimiento para mitigar los efectos del error.

Dado que el valor de aceleración deseado en este preprocesado no es el valor por cada eje sino un único valor que identifique el movimiento, se obtiene un único valor de aceleración compuesto por la combinación de los tres ejes a partir de la siguiente ecuación, que obtiene el módulo de los vectores:

$$modAcc = \sqrt{nAX^2 + nAY^2 + nAZ^2} \tag{2.4}$$

Este valor combinado es interesante al mostrar la fuerza con la que se ha acometido el movimiento, independientemente de la dirección del mismo.

Aunque este valor de aceleración muestra la fuerza del movimiento, existe la necesidad de determinar por otra parte la forma en la que se ha realizado el movimiento en cuanto a la decisión con la que se ha llevado a cabo. Es decir, a partir de los datos de los sensores, se pretende determinar si el movimiento se ha hecho de forma decidida

o dubitativa. Para ello, se ha diseñado un método específico para la obtención de los valores de agitación durante el movimiento. Se ha definido primero una agitación como un movimiento oscilatorio realizado con la mano durante el movimiento.

Dado que el módulo del vector de aceleración calculado anteriormente es una función siempre positiva, se puede considerar cualquier máximo local de dicha señal como un valor de agitación. Para poder clasificar estas agitaciones según su intensidad, se han determinado cuatro niveles (etiquetados como 1, 2, 3 y 4+). Para categorizar un máximo local en uno de los niveles, se van contando las muestras obtenidas en cada medición entre un mínimo y un máximo local, teniendo siempre en cuenta que se está realizando un muestreo constante. Dependiendo del número de muestras contadas entre un mínimo y un máximo, se determina que una agitación está en un nivel u otro. Así, por ejemplo, en la Figura 2.21, se puede ver que en el caso del máximo local detectado en el módulo de la aceleración en a), estamos ante una agitación suave de nivel 1, ya que sólo hay una muestra entre el máximo y cada mínimo a su alrededor. En cambio, el valor de agitación de b) está rodeado de 3 muestras intermedias entre el máximo y el mínimo a su izquierda, pero por 5 muestras intermedias entre el máximo y el mínimo de su derecha. En este caso, se determina que se va a utilizar el menor de los dos valores para categorizar la agitación, por lo que sería incluido como una agitación de nivel 3.

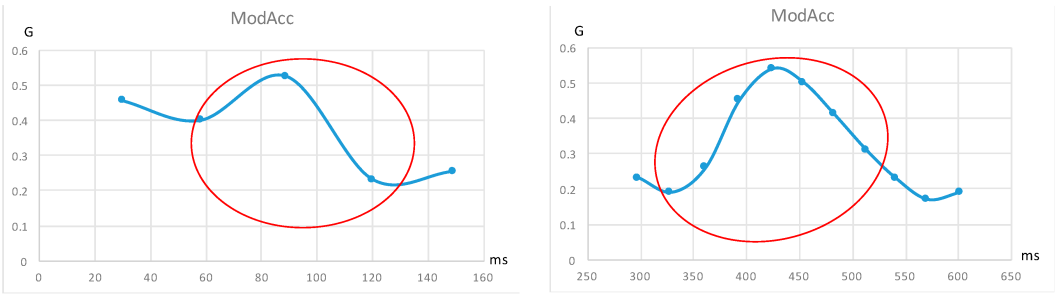

Figura 2.21: Gráficas que muestran ejemplos de máximos locales en la señal de aceleración. Un máximo rodeado de dos mínimos o agitación de nivel 1 (a) y un máximo rodeado de 3 y 5 muestras o agitación de nivel 3 (b).

El nivel 4+ incluye no sólo las agitaciones de categoría 4 (es decir, con 4 muestras entre el máximo y alguno de los mínimos a su alrededor) sino que incluye también aquellas de mayor órden. Esto es así debido a que, tras realizar varias pruebas ex-

perimentales, no se consiguen habitualmente valores mucho mayores. Finalmente, la cantidad de valores de agitación de cada tipo se incluye en un vector que identificará la agitación general durante el movimiento.

Hasta ahora, en esta sección sobre el preprocesado de datos se ha indicado que se enviarán desde los dispositivos valores obtenidos para cada movimiento completo. Para ello, hay que detectar cuándo se inicia y cuándo finaliza un movimiento en el propio dispositivo. Esta detección se realiza en base a los valores devueltos por el giróscopo del MPU-9150. Para ello se toma un valor (MGyro) que se obtiene de la fusión de los ángulos medidos en cada eje (GyroX, GyroY y GyroZ, en grados por segundo). A partir de este valor, se puede establecer un valor umbral a partir del cual se puede considerar si el cubo está en movimiento o no.

Además de determinar un movimiento a partir de este umbral, y teniendo en cuenta que ciertos "falsos" movimientos como golpes en la superficie donde se encuentra el dispositivo podrían ser tomados como movimientos reales, se ha considerado un umbral de tiempo que elimina todo movimiento detectado por debajo de él (el umbral se ha fijado en 0,2 segundos en el prototipo mediante experimentación).

Para el cálculo de los valores de velocidad, se realiza la operación de integración sobre los valores de aceleración. Al tener tres valores de aceleración, se deben integrar cada uno de ellos por separado para obtener tres componentes de velocidad VX, VY y VZ. Cualquier sesgo del sensor de aceleración puede hacer que la operación de integración vaya acumulando un error, y por tanto se fueran obteniendo valores indefinidamente incrementales de velocidad. Para evitar esto, se resta el sesgo estimado mediante la calibración en cada eje. De esta forma, se mitiga el efecto del sesgo y es posible determinar valores aceptablemente precisos de velocidad. Si el movimiento se mantuviera durante un largo tiempo, este ajuste no evitaría la aparición de errores, pero dado que en las actividades que se van a realizar los movimientos no tendrán duraciones mayores a unos pocos segundos, es un ajuste suficiente. En las siguientes ecuaciones se puede ver la operación del cálculo de la velocidad en cada eje a partir de la integración de la aceleración:

$$VX = \int_{t0}^{t} (nAX - e_x)dt \tag{2.5}$$

$$VY = \int_{t0}^{t} (nAY - e_y)dt \tag{2.6}$$

$$VZ = \int_{t0}^{t} (nAZ - e_z)dt \tag{2.7}$$

Donde t0 es el momento inicial del movimiento y t es el momento final. Cada valor e es el valor calculado para el sesgo inicialmente.

Finalmente, de la misma forma que se hace con la aceleración, se puede calcular el módulo de los tres ejes para obtener un único valor de velocidad medio y máximo en el movimiento.

$$modV = \sqrt{VX^2 + VY^2 + VZ^2} \tag{2.8}$$

2.4.4. Pruebas piloto

A partir de los prototipos construidos, se han realizado una serie de pruebas piloto con el objetivo de validar los diseños propuestos, tanto desde el punto de vista arquitectónico, verificando que se cumplen los requisitos definidos para la arquitectura, como desde el punto de vista de la validez de los datos obtenidos. Para ello, aprovechando el carácter multidisciplinar del proyecto EDUCERE, en el que han participado expertos en desarrollo infantil de varias universidades españolas, se han realizado pruebas en escuelas infantiles de la Comunidad de Madrid con la supervisión de estos expertos. En los siguientes apartados se describe cómo se han realizado dichas pruebas y los resultados obtenidos.

2.4.4.1. Descripción de los experimentos

Para la validación de los experimentos se han concertado con varias escuelas infantiles de la comunidad de Madrid una serie de sesiones en las que, niños de edades comprendidas entre aproximadamente 2 y 3 años pudieran realizar una actividad con

los dispositivos desarrollados.

Aunque el objetivo final de esta arquitectura de juguetes inteligentes es su implantación de forma autónoma en escuelas e incluso hogares, para estas pruebas se ha determinado que estuvieran presentes:

- Un educador de las propias escuelas para estimular la confianza y la tranquilidad de los niños durante las pruebas.
- Al menos un experto en desarrollo infantil que pudiera validar que estas pruebas se llevaban a cabo de forma adecuada desde el punto de vista de la experimentación en desarrollo (psicología, psicopedagogía, etc.).
- Al menos un ingeniero relacionado con el desarrollo y con la tecnología utilizada para validar el funcionamiento de los dispositivos y servir de apoyo técnico en caso de fallos.

La actividad a realizar seleccionada ha sido la de apilamiento de cubos. Esto se debe por un lado a que es una de las actividades más utilizadas por parte de los expertos, al estar estandarizada en varias escalas de medición sobre el desarrollo infantil, y por otro lado, a que permite validar el funcionamiento de los cubos prototipo desarrollados, que contienen la mayor innovación desde el punto de vista de diseño hardware y por tanto son un posible punto de fallo inicial.

Para asegurar una repetición lo más fiel posible en cada uno de los experimentos, se ha utilizado una plantilla sobre la que se han colocado en cada ocasión cinco cubos. Cerca de estos cubos se ha colocado el gateway, y se ha provisto al experto o expertos en desarrollo presentes de una tablet con la que poder gestionar las actividades a través de una de las aplicaciones cliente desarrolladas.

Se ha intenado en todo momento alterar lo menos posible el entorno habitual del niño, para lo cual se han llevado a cabo las pruebas en un entorno conocido por el niño (habitualmente una clase de la escuela), con un grupo de personas reducido y con la colaboración de su educador.

Finalmente, en el lugar de experimentación, y enfocando a la superficie donde se

va a realizar la actividad (normalmente una mesa baja, adaptada a la altura de los niños), se ha colocado en cada caso una cámara de vídeo con el objetivo de grabar cada uno de los experimentos. La grabación en vídeo no es parte del sistema de obtención de datos que se pretende conseguir con esta arquitectura, pero para las pruebas piloto se ha decidido que es necesario contar con una fuente de datos externa que permita realizar *a posteriori* un análisis comparativo de los resultados de los experimentos.

Se han llevado a cabo actividades de apilamiento de cubos con 65 niños (en total 32 niños y 33 niñas), con edades entre 23 y 37 meses (con una edad media de 29,02 meses, y una desviación típica de 3,81). En la Figura 2.22 se puede ver un ejemplo de una de las pruebas, extraído de uno de los videos grabados.

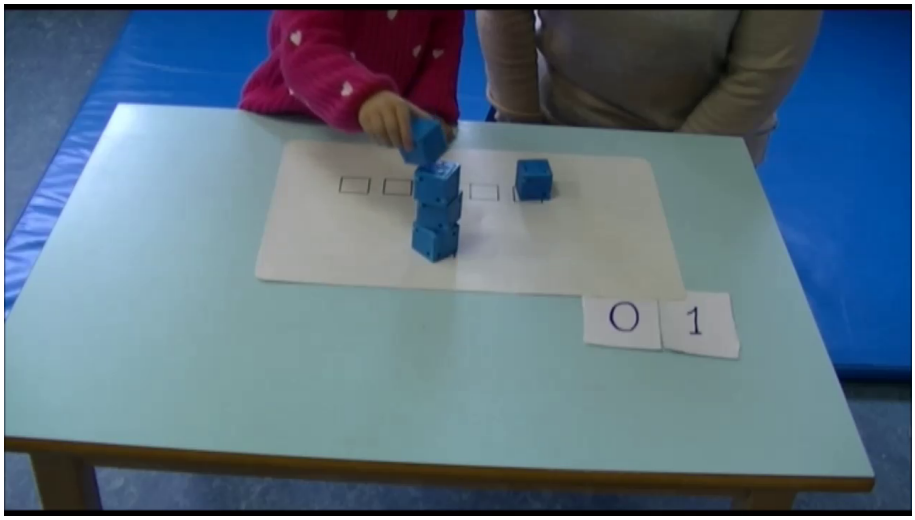

Figura 2.22: Captura de pantalla de uno de los videos grabados durante las pruebas realizadas.

Para poder realizar estas actividades, se ha contado con el permiso paterno de cada uno de los niños involucrados. Los padres de cada niño firmaron con anterioridad un documento de consentimiento informado de la actividad. En cualquier caso, se ha procurado en todas las actividades que las identidades de cada niño no fueran distinguibles ni en la grabación ni en el sistema, en el cual los datos sólo aparecen asociados a un identificador numérico y sólo las personas autorizadas pueden acceder a la información completa. Además, no se ha recabado ninguna información clínica de los niños.

El sistema se ha configurado para obtener datos en base al sistema de preprocesado explicado en la sección del diseño de los prototipos, por lo que al finalizar los experimentos se han obtenido una serie de vectores de datos para cada niño conteniendo, para cada movimiento realizado con cada cubo. A partir de esos datos, se ha establecido la tabla de variables resumen para cada experimento (Tabla 2.2)

Tabla 2.2: Tabla de resumen de datos por experimento y niño

Variable	Significado	Dimensión
Número de movimientos	Número total de movimientos realizados durante la actividad completa	$n \in \mathbb{N}$
Tiempo medio de movimientos	Media de las duraciones de cada uno de los movimientos registrados	ms
Velocidad media de movimientos	Media de las velocidades medias de cada uno de los movimientos registrados	m/s
Media de velocidades máximas	Media de los valores máximos de velocidad alcanzados en todos los movimientos	m/s
Mayor velocidad máxima	El mayor valor de entre todas las velocidades máximas registradas	m/s
Menor velocidad máxima	El menor valor de entre todas las velocidades máximas registradas	m/s
Media de las aceleraciones máximas	Media de todas las aceleraciones máximas registradas	m/s^2
Mayor aceleración máxima	Valor máximo de entre todas las aceleraciones máximas registradas	m/s^2
Menor aceleración máxima	Valor mínimo de entre todas las aceleraciones máximas registradas	m/s^2
Media de las agitaciones de nivel 1	Media del número de agitaciones de nivel 1 registradas en todos los movimientos	$n \in \mathbb{N}$
Media de las agitaciones de nivel 2	Media del número de agitaciones de nivel 2 registradas en todos los movimientos	$n \in \mathbb{N}$
Media de las agitaciones de nivel 3	Media del número de agitaciones de nivel 3 registradas en todos los movimientos	$n \in \mathbb{N}$
Media de las agitaciones de nivel 4+	Media del número de agitaciones de nivel 4+ registradas en todos los movimientos	$n \in \mathbb{N}$

Una vez pasadas todas las pruebas, se ha realizado un visionado detallado de cada video por parte de un grupo de cuatro expertos en desarrollo (un psicólogo del desarrollo, un fisioterapeuta y dos pedagogos) con el fin de obtener una puntuación por cada uno de ellos en base a su experiencia profesional. Estas puntuaciones, en una escala de 1 a 10, se han añadido a la matriz de datos obtenida para cada niño. Cada experto ha

visualizado la mitad de los videos grabados, de forma que al final cada video haya sido visto por dos expertos. Las puntuaciones se han establecido de manera individual por cada uno de ellos.

2.4.4.2. Resultados

Mediante el uso de la herramienta de análisis estadístico IBM SPSS [111], se ha realizado un análisis de la correlación intraclase utilizando un modelo ICC(1,k) (*Intraclass Correlation*) en el que que para cada participante se utilizan un número k de puntuaciones (en este caso 2, una por cada experto).

El objetivo es el cálculo de la fiabilidad de cada una de las 65 puntuaciones. El resultado para valores individuales del ICC es 0,961 (95 % CI 0.937-0.976; F64,64=50.39, P<.001) y la ICC para el valor medio de las puntuaciones es 0.980 (95 % CI 0.967-0.988; F64,64=50.39, P<.001). Estos valores parecen indicar que la fiabilidad de las puntuaciones es alta y está altamente correlacionada. Dados estos resultados, se toma la media de las puntuaciones como valor de puntuación en los siguientes análisis.

A continuación, se ha realizado un análisis factorial de los datos (ver Tabla 2.2). Este análisis pretende reducir las variables necesarias para la aplicación posterior de un modelo de regresión. Este tipo de modelo se podría utilizar para construir un sistema de detección en base a los datos posteriormente. En cualquier caso, los resultados de este análisis permiten determinar la validez de cada una de las variables procesadas.

El análisis factorial ha determinado que el 76,78 % de la varianza corresponde a los tres primeros factores. En la Tabla 2.3 se muestran los valores de correlación entre las variables y el factor.

En esta tabla, se muestran las correlaciones más importantes entre las variables y los factores. Estos factores pueden verse como agrupaciones de variables relacionadas entre sí. Por ello, se podrían definir de la siguiente forma:

- Componente 1 (Agitaciones): Altas correlaciones con las variables “tiempo medio de los movimientos” y “media de agitaciones” (para los niveles 1 a 4).

Tabla 2.3: Tabla de valores de correlación para cada variable y los tres componentes (factores) que más explican la varianza. En negrita, los valores máximos de correlación de cada variable con un componente.

	Componente 1	**Componente 2**	**Componente 3**
Número de movimientos	-0,049	0,294	**-0,782**
Tiempo medio de movimientos	**0,983**	-0,048	0,003
Velocidad media de movimientos	0,015	**0,840**	0,199
Media de las velocidades máximas	-0.024	**0,943**	-0,009
Mayor velocidad máxima	-0,078	**0,723**	-0,572
Menor velocidad máxima	0,035	0,264	**0,800**
Media de las aceleraciones máximas	-0,083	**0,809**	0,139
Mayor aceleración máxima	-0,090	**0,642**	-0,597
Menor aceleración máxima	-0,044	0,447	**0,784**
Media de agitaciones de nivel 1	**0,747**	-0,021	-0,229
Media de agitaciones de nivel 2	**0,896**	-0,120	-0,047
Media de agitaciones de nivel 3	**0,892**	-0,024	0,173
Media de agitaciones de nivel 4	**0,728**	0,022	0,225

- Componente 2 (Velocidades): Altos valores de correlación para las variables "Velocidad media de movimientos", "media de las velocidades máximas", "mayor velocidad máxima", "media de las aceleraciones máximas" y "mayor aceleración máxima".
- Componente 3 (Precisión): Altos valores de correlación para las variables "número de movimientos", "menor velocidad máxima" y "menor aceleración máxima". La correlación en el primer caso es negativa, por lo que a mayor número de movimientos, menor "precisión" en el desarrollo de la actividad.

A partir de estas agrupaciones, se han realizado análisis de regresión con el objetivo de obtener una "fórmula" que indique en qué medida cada variable contribuye con información sobre la realización de las actividades.

En la Tabla 2.4, se puede observar un resumen de los análisis llevados a cabo. En el primero (Modelo 1), se ha relacionado cada uno de los componentes anteriores con el valor de evaluación de la actividad presentado por los expertos, mientras que en el otro (Modelo 2) se han relacionado los componentes con la edad de los niños.

Tabla 2.4: Resultados de los modelos de regresión para la evaluación y la edad.

	Variable	**Predictores**	R	R^2	R^2 **ajustado**	**Desv. est.**
Modelo 1	Evaluación	Agit., vel., prec.	0,517	0,267	0,231	1,556
Modelo 2	Edad	Agit., vel., prec.	0,362	0,131	0,089	3,637

En la Tabla 2.5 se pueden ver los coeficientes calculados para cada componente en cada uno de los modelos mediante el análisis de regresión múltiple. Basándonos en los resultados del modelo vistos en la tabla, la ecuación de regresión para el primer modelo sería:

$$Evaluacion = 7{,}662 + 0{,}05(A) - 0{,}630(V) + 0{,}665(P) \tag{2.9}$$

A, V y P respresentan los conjuntos de variables de agitaciones, velocidades y precisión, respectivametne. El coeficiente del componente de agitaciones es muy bajo, por lo que su influencia en el resultado es limitada. Además, al ser positivo, se podría deducir que, a un mayor número de agitaciones, se obtiene una mejor evaluación en la actividad, lo cual es al menos contraintuitivo. El coeficiente para las variables de precisión es positivo y bastante por encima del cero, por lo que este tipo de variables tiene una relación directa con la puntuación de evaluación. Sin embargo, el coeficiente de las variables de velocidad es negativo y grande, por lo que existe una relación inversa sobre estas variables y la puntuación obtenida.

Tabla 2.5: Coeficientes obtenidos para cada uno de los componentes de análisis en cada uno de los dos modelos.

Modelo	**Componente**	**Coeficiente**	**Error estándar**	**P**
1	Constante	7,662	0,193	<0,001
	Agitaciones	0,050	0,194	0,80
	Velocidades	-0,630	0,194	0,002
	Precisión	0,665	0,194	0,001
2	Constante	29,015	0,451	<0,001
	Agitaciones	-0,152	0,455	0,74
	Velocidades	0,003	0,455	0,99
	Precisión	1,372	0,455	0,004

Para el segundo modelo tendríamos la ecuación:

$$Edad = 29,015 - 0,152(A) + 0,03(V) + 1,372(P) \tag{2.10}$$

En este caso, sólo el coeficiente de las variables de precisión es significativo, ya que en los otros dos componentes tenemos valores muy cercanos al cero. Este resultado es lógico dado que significaría que la precisión con la que se realizan las actividades está relacionada con la edad del participante.

2.5. Resumen y consideraciones finales

En este capítulo se ha presentado el diseño de la plataforma de Smart Toys que sirve de base al trabajo de investigación lleavado a cabo a lo largo de esta Tesis. La inexistencia de arquitecturas estándar y genéricas en entornos IoT nos ha llevado a plantead la definición de una arquitectura específica para esta plataforma. Sin embargo, para asegurar la extensibilidad e interoperabilidad de la plataforma con futuros desarrollos y otros sistemas, se ha realizado un diseño basado en un modelo de referencia que facilite esas futuras tareas mientras asegura que las funcionalidades requeridas en la plataforma se cumplen.

El diseño de la plataforma se basa en una serie de vistas de diseño arquitectónico que definen la plataforma desde distintos puntos de vista. Desde el punto de vista funcional, se han establecido una serie de grupos de funcionalidades con características comunes como son la gestión de procesos, la definición de entidades virtuales y servicios IoT, así como las consideraciones en cuanto a comunicaciones y seguridad. Por otro lado, desde el punto de vista de la información, se ha definido cómo se describe ésta en la plataforma, cómo se relaciona con los componentes funcionales de la misma y el ciclo de vida que sigue. Desde el punto de vista del contexto, se definen las relaciones entre la plataforma y su entorno, especialmente entre los distintos usuarios y beneficiarios de su uso.

Desde un punto de vista físico, se definen las entidades físicas que componen la plataforma: Smart Toys, gateways, servidores y clientes o interfaces de usuario. A

partir de estas entidades, se han construido una serie de prototipos, haciendo hincapié en el diseño de Smart Toys que puedan ser útiles para los expertos en desarrollo infantil. Estos juguetes se han basado en las escalas ya utilizadas por los expertos, para servir de punto de partida en las pruebas piloto de evaluación.

Se han desarrollado en el marco de la Tesis dos Smart Toys completos: Un conjunto de Smart Cubes apilables y una tabla de espigas. El diseño de ambos dispositivos, así como los mecanismos de procesamiento de los datos recogidos por sus sensores se ha incluido también en este capítulo.

Los prototipos han sido utilizados en unas pruebas piloto preliminares en escuelas infantiles reales, y el análisis de los datos recogidos ha permitido definir la importancia de cada tipo de información obtenida por los Smart Toys en relación a la evaluación realizada por expertos. Estas pruebas han servido también para verificar el funcionamiento de los prototipos y la plataforma y su viabilidad.

Capítulo 3

Seguridad y privacidad en un entorno de Smart Toys

En este capítulo se abordan los requisitos de seguridad de la plataforma diseñada y se describen los dos principales mecanismos propuestos para asegurarla. Para ello, tras una sección introductoria (sección 3.1), se ha realizado un estudio del estado del arte y los trabajos de seguridad relacionados (sección 3.2). A continuación se presenta un análisis de las principales amenazas al sistema en la sección 3.3, y a partir de él, se describen los dos mecanismos de seguridad propuestos: Un sistema de autenticación y cifrado para garantizar la confidencialidad y autenticidad de los mensajes enviados utilizando las características específicas de comunicaciones de los Smart Toys (sección 3.4) y un sistema de control de acceso diseñado para ser utilizado de forma unificada en toda la plataforma (sección 3.5). Finalmente, en la sección 3.6 se describen las implementaciones llevadas a cabo y se muestran los resultados obtenidos en las pruebas realizadas, y en la sección 3.7 se incluye un resumen del capítulo.

3.1. Introducción

Una de las características más importantes de todo sistema es la seguridad, especialmente en aquellos que se basan en el despliegue de múltiples dispositivos heterogéneos

comunicados entre sí y a través de Internet, como son las plataformas basadas en IoT.

Con el incremento de este tipo de sistemas, muchos delincuentes están variando la forma en la que realizan sus ataques, buscando las vulnerabilidades de estos dispositivos finales y las pasarelas que los conectan a Internet. La motivación de estos ataques viene por una parte de una menor preocupación por parte de los fabricantes en la seguridad y por otra en la gran cantidad de información relativa a usuarios que es posible obtener de estos dispositivos (incluyendo su localización, movimientos, salud, etc.). Un estudio llevado a cabo en 2014 por la empresa Hewlett-Packard [112] determinó que aproximadamente el 70 % de los dispositivos IoT analizados eran vulnerables a ataques de uno u otro tipo. Es más, tras un importante ataque a algunos de los servicios más importantes de Internet llevado a cabo a través de dispositivos IoT vulnerables en 2016 [113, 114], el FBI declaró que este tipo de ataques ser harían cada vez más comunes debido a la cantidad de *malware* disponible y la cantidad de posibles objetivos que no cuentan con la suficiente seguridad [115].

En este contexto, es imprescindible definir un modelo de seguridad, privacidad y confianza para los entornos IoT, que eviten el uso no autorizado del sistema, ya sea por usuarios o por otros dispositivos. Tal y como se indica en trabajos como [116], un modelo IoT debe garantizar anonimización de los datos, confidencialidad e integridad. Debe además ofrecer mecanismos para la autenticación y la autorización para el acceso.

En las plataformas IoT conformadas por Smart Toys como la que se propone en este trabajo, existen preocupaciones adicionales específicas acerca de los riesgos que pueden correr los niños que usen estas tecnologías [117], haciéndose hincapié en los riesgos para la privacidad que puede suponer su uso por parte de usuarios menores de edad. De hecho, algunos estudios indican que algunos de los sistemas comerciales disponibles actualmente podrían no cumplir los requisitos de seguridad que se deben exigir a estos sistemas [82].

Por otro lado, ya en el capítulo 2 se ha definido la seguridad como uno de los grupos funcionales fundamentales de la plataforma, basándonos en los marcos de referencia disponibles para el diseño de estas arquitecturas, por lo que se puede considerar un grupo de funcionalidades fundamental para el sistema.

Teniendo en cuenta las preocupaciones y riesgos que existen debido al uso de plataformas IoT, en la plataforma que se ha diseñado a lo largo de este trabajo se ha llevado a cabo un análisis de las amenazas específicas que podrían afectar a las comunicaciones y los dispositivos diseñados, y a partir de ellas, se han definido los sistemas de seguridad a implementar. En general, se ha intentado que estos sistemas se hayan basado en algoritmos, funciones y mecanismos ya conocidos de seguridad, que han sido ampliamente probados y validados anteriormente, dado el riesgo que un fallo en estos esquemas puede suponer para la plataforma y sus usuarios.

Ya que la confidencialidad y la privacidad son algunas de las preocupaciones más importantes en las plataformas de Smart Toys, se ha diseñado un sistema específico para proteger las comunicaciones de los Smart Toys, ofreciendo confidencialidad y autenticación en el punto más frágil de la plataforma, como son los dispositivos IoT. Por otro lado, se ha propuesto unificar el control de acceso en toda la plataforma, de forma que sea posible proteger de la misma manera todas las comunicaciones que se lleven a cabo en ésta, independientemente del mecanismo de comunicación utilizado en cada caso, facilitando así esta tarea a los usuarios.

3.2. Estado del arte y trabajos relacionados

La seguridad es una de las principales preocupaciones en cualquier sistema tecnológico, y las plataformas basadas en IoT no son una excepción. De hecho, casi desde las primeras propuestas relacionadas con este paradigma, han surgido estudios que describen los principales desafíos en cuanto a seguridad a los que se enfrentan este tipo de arquitecturas. Así, por ejemplo, en [118], se ofrece un estudio que determina los condicionantes legales que deben cumplirse en este tipo de sistemas, así como los hitos que se deben cumplir.

En estudios recientes como [119] o [120], se identifican algunos de los principales desafíos y se revisan estrategias y métodos para afrontarlos, basándose en los últimos avances en hardware y su utilización para garantizar la seguridad, especialmente en los dispositivos autónomos que componen estas arquitecturas. En [120] se clasifican los problemas de seguridad según afecten a cada una de las capas de una arquitectura

basada en niveles. Así, se divide la problemática en aquella derivada de la capa de aplicación (problemas que puedan surgir del almacenamiento de datos, sobre la privacidad de los mismos, la autenticación y el control de acceso, etc.), de la capa de red (medidas que se apliquen sobre las comunicaciones para asegurar la autenticación, integridad, confidencialidad, etc., así como la mitigación de posibles ataques de denegación de servicio) o de la capa de percepción (es decir, las medidas de seguridad físicas a implementar para evitar robos, vandalismo, etc.). Esta clasificación se puede extrapolar a las arquitecturas basadas en cinco capas vistas en la sección 2.2.

En trabajos como [121], se clasifican los requisitos de seguridad según precisamente la tarea de seguridad a la que se aplican. Así, en una plataforma de este tipo se identifican requisitos relativos a la autenticación y la confidencialidad de los datos, el control de acceso a los mismos, y los mecanismos para asegurar la privacidad y la confianza en el sistema.

En general, los mecanismos de seguridad sistemas IoT dependen en una gran medida de las características tecnológicas de los dispositivos implicados en el sistema, así como de los protocolos de comunicación utilizados por éstos. Esto es debido a que no todos los dispositivos tienen la misma capacidad de procesamiento, y no todos los protocolos utilizan el mismo esquema de intercambio de datos.

En [122] se incluye por ejemplo una clasificación de protocolos de comunicaciones utilizados en entornos IoT, donde se diferencian en función de la capa o nivel en el que se utilizan. Se mencionan protocolos del nivel de infraestructura (por ejemplo, IEEE 82.15.4 [123] , "IPv6 over Low power Wireless Personal Area Networks" (6LoWPAN) [124] o el "IPv6 Routing Protocol for low-power and Lossy networks" (RPL) [125]), nivel de descubrimiento de servicios (como "multicast Domain Name System (mDNS) o Domain Name System Service Discovery (DNS-SD) [126]) y finalmente, protocolos de nivel de aplicación.

Son estos protocolos de nivel de aplicación los que mayor relación tienen con los sistemas de control de acceso, ya que la protección de la información se realiza habitualmente en función de las estructuras de datos compartidas a través de ellos.

Estos protocolos se pueden clasificar de acuerdo a cómo organizan las comunicacio-

nes: En general se basan en el modelo de publicación/suscripción ya que es un modelo altamente demandado por las necesidades de las plataformas IoT, incluyendo la definida en este trabajo, tal y como se puede ver en la sección 2.3.2.2. Los protocolos más utilizados de este tipo son "Message Queue Telemetry Transport" (MQTT), "Advanced Message Queueing Protocol" (AMQP), y "Data Distribution Service" (DDS).

- MQTT es un protocolo diseñado por IBM para ofrecer comunicaciones ligeras en entornos M2M. Se basa en un servidor central (broker) que gestiona las suscripciones. Es un procolo usado ampliamente en entornos IoT por su bajo consumo y su fiabilidad [127].
- AMQP es una alternativa a MQTT que sigue el mismo esquema de comunicaciones, basándose en intercambio de mensajes a través de colas gestionadas por un broker [128].
- DDS es también una alternativa de publicación/suscripción desarrollada por el Object Management Group (OMG). Se diferencia de los anteriores en la no utilización de un servidor centralizado, basándose en lugar en comunicaciones multicast [129].

Además de estos protocolos anteriores, existen protocolos de aplicación clásicos como HTTP se utilizan también ampliamente en escenarios IoT, habitualmente a través de servicios REST ("Representational State Transfer") [130]. En una posición intermedia entre los protocolos basados en preguntas y respuestas como HTTP y los protocolos anteriores, se pueden situar otros protocolos muy extendidos como "Constrained Application Protocol" (CoAP) y "Extensible Messaging and Presence Protocol" (XMPP).

- CoAP se ha desarrollado por el "Internet Engineering Task Force" (IETF) a través del grupo "Constrained RESTful Environments" (CoRE), y define un sistema de intercambio de mensajes basado en REST [131] que permite una alta interoperabilidad entre los modelos de publicación/suscripción y petición/respuesta basados en HTTP.
- XMPP es también un estándar del IETF diseñado para mensajería instantánea pero utilizado en multitud de escenarios IoT [132].

En las siguientes dos secciones se realiza un análisis de los principales trabajos relacionados con la autenticación, integridad y confidencialidad de los datos en plataformas IoT, y sobre mecanismos existentes de control de acceso que se puedan utilizar, extender o adaptar a entornos IoT, dependiendo de las tecnologías y protocolos utilizados en cada caso.

3.2.1. Autenticación, integridad y confidencialidad de los datos

Sobre la autenticación, integridad y confidencialidad, existen un gran número de propuestas, algunas de ellas derivadas de mecanismos existentes en escenarios basados en Internet [133], pero en general, tal y como se concluye en [121], no existen aún mecanismos generales que permitan asegurar todos estos elementos en cualquier escenario IoT, debido a la gran heterogeneidad de los dispositivos y las comunicaciones entre ellos, que hace que los mecanismos no sean siempre aplicables en todos los casos.

La dependencia de estos mecanismos de seguridad de los protocolos hace que, en cada caso, y dependiendo de las especificaciones de éstos, se puedan utilizar unos u otros algoritmos. En [134] se pueden observar un gran número de ejemplos de mecanismos de autenticación, de integridad y de confidencialidad, clasificados según el protocolo utilizado.

Siguiendo la clasificación de protocolos incluida en la sección anterior, se pueden diferenciar en mecanismos a nivel de infraestructura o a nivel de aplicación. A nivel de infraestructura se pueden destacar:

- Mecanismos para IEEE 802.11.4: El estándar ofrece una serie de servicios de seguridad en la capa MAC, que apoyan otros mecanismos que se pueden utilizar en capas más altas de comunicación para el cifrado y la autenticación de los datos. Estas tareas se basan en los algoritmos AES en distintos modos (CBC, CTR, CCM). Existen dispositivos que implementan ya en su propio hardware estos sistemas, como [135].
- Mecanismos para 6LowPAN: A diferencia de 802.1.14, 6LowPAN no define mecanismos específicos de seguridad, pero sí se ofrecen una serie de requisitos y guías

de seguridad en su documentación, que dependerán de las restricciones de los dispositivos donde se implemente. Existen propuestas y estudios sobre la seguridad en este tipo de redes que se basan en el uso del protocolo DTLS ("Datagram Transport Layer Security") y otros mecanismos específicos [136].

- Mecanismos para RPL: Este protocolo ofrece un mecanismo específico para dotar de autenticación y, opcionalmente, confidencialidad a los mensajes que enruta, basado en AES en modo CCM e incluye campos específicos en su cabecera para especificar el nivel de seguridad [137].

A nivel de aplicación, se tienen entre otros los siguientes mecanismos:

- Mecanismos para CoAP: El mecanismo que se recomienda utilizar para ofrecer seguridad en CoAP es DTLS. Este sistema sin embargo no es aplicable a todos los dispositivos debido a un alto coste computacional [134].
- Mecanismos para MQTT: En MQTT no se especifican tampoco mecanismos de seguridad para confidencialidad, autenticación, etc., pero la especificación recomienda el uso de TLS ("Transport Layer Security") [138], con las mismas restricciones que se tienen en CoAP.
- Mecanismos para XMPP: Según su especificación [139], los mecanismos de seguridad se basan también en TLS para el cifrado y en SASL ("Simple Authentication and Security Layer") para la autenticación.

Además de estos mecanismos estandarizados para cada uno de los protocolos más utilizados, hay varias propuestas relacionadas con estas tareas de seguridad, sobre todo centradas en los mecanismos para la gestión de claves [140], que pueden ser utilizadas en este tipo de entornos [141, 142]. Aunque no es habitual, existen propuestas basadas en infraestructuras de clave pública como [143, 144]. Los sistemas de autenticación por su parte se suelen basar en mecanismos lo más ligeros posibles, ya que se deben diseñar con el objetivo de ser implementados y utilizados en dispositivos con pocos recursos [145]. En los últimos tiempos se están explorando técnicas basadas en mecanismos hardware. Entre ellas, se puede destacar los sistemas basados en PUF ("Physical Unclonable

Function"), referenciados en [119]. Este tipo de mecanismos permite tener información específica a partir de las características hardware de un dispositivo concreto, lo cual es muy útil a la hora de autenticar dispositivos en un entorno con un gran número de ellos, así como para la generación de material criptográfico para el cifrado en estos mismos entornos.

Como se puede observar en el estudio anterior, la mayoría de los mecanismos se centran en un protocolo específico, y no es posible determinar un mecanismo genérico que tenga en cuenta los requisitos de cualquier dispositivo o plataforma. A nivel de aplicación, muchos de estos protocolos se basan en TLS o sistemas similares para dotar a las comunicaciones de seguridad, sin embargo, esto no es extrapolable a todos los escenarios, donde dispositivos muy restrictivos en cuanto a recursos no pueden implementar esos sistemas. Por otro lado, a nivel de infraestructura se usan mecanismos basados habitualmente en algoritmos conocidos de cifrado como AES en sus distintos modos de funcionamiento.

3.2.2. Privacidad y control de acceso a los recursos

De entre todos los aspectos de seguridad citados anteriormente, aquellos que más afectan a esta plataforma tienen que ver con la privacidad y el control del acceso a la información generada por ella. De hecho, la privacidad es una de las mayores preocupaciones de este tipo de plataformas, como se puede ver en análisis de casos prácticos (basados en sistemas comerciales) tales como [82], o [146] y también en estudios anteriores en los que se ya se advertía de la posible pérdida de privacidad de los niños [147]. En [148], se analizan estas preocupaciones por la seguridad de los datos en juguetes con capacidad de conectividad.

Para evitar este problema, se han realizado varias propuestas en los últimos tiempos, como por ejemplo las mostradas en [149] y [150]. En el primero de estos trabajos se define primero la problemática en cuanto a privacidad derivada del uso de juguetes interconectados, y propone un marco de actuación tanto para padres como para fabricantes. El segundo define reglas conceptuales para asegurar la privacidad y propone un modelo de control de acceso basado en ellas.

En otros estudios como [151] se comparan varios Smart Toys comerciales analizando la seguridad que ofrecen. A partir de ello, realizan un análisis de las posibles amenazas y definen los requisitos para afrontarlas, para lo que utilizan el Security Development Lifecycle (SDL) de Microsoft. En [152] mantienen un enfoque similar, analizando las vulnerabilidades de varios juguetes comerciales, aunque en este caso no proponen soluciones para mitigarlas. Algo parecido se puede encontrar en [153] donde analizan varios Smart Toys y realizan un análisis sobre la seguridad y las posibles amenazas encontradas.

Mientras que la confidencialidad, la integridad y la autenticación de los datos se obtienen mediante la aplicación de esquemas de cifrado y compartición de claves, una tarea más "compleja" desde el punto de vista de la infraestructura necesaria para su despliegue es el control del acceso a los datos en la plataforma. Hay que tener en cuenta que esta tarea se complica en este tipo de escenarios, ya que existe una alta heterogeneidad en cuanto a puntos de acceso, dispositivos y mecanismos de comunicaciones utilizados.

Existen numerosas propuestas para modelar los sistemas de control de acceso en plataformas IoT, tal y como se puede ver en [154]. En este trabajo se presenta una clasificación de sistemas de control de acceso, protocolos y marcos de trabajo en estos escenarios.

Los modelos de control de acceso se pueden clasificar dependiendo de en qué basan este control. Así, por ejemplo, se pueden hablar de modelos basados en roles (RBAC), modelos basados en atributos (ABAC), modelos basados en control del uso (UCON), modelos basados en capacidades (CapBAC) y modelos organizacionales (OrBAC), entre otros.

- Los modelos de tipo RBAC [155] se basan en la clasificación de usuarios en el sistema de acuerdo a roles y los permisos dados a esos roles. Es un modelo más apropiado para escenarios con una sencilla identificación de los usuarios y servicios, pero hay propuestas para su aplicación en IoT como las que se muestran en [156] o en [157].

- Los modelos ABAC [158] se basan en políticas comprobables en base a ciertos

atributos de los sujetos. Se ha probado en sistemas IoT por ejemplo en [159] y [160].

- Los modelos UCON [161] se pueden ver como una evolución de los modelos RBAC y ABAC, en los que se tiene un sistema más flexible.

- Los modelos CapBAC [162, 163] se basan en el concepto de capacidad (normalmente un ticket o token que garantiza ciertos permisos), se suele basar en matrices de control de acceso (ACM), variantes de las clásicas listas de control de acceso (ACLs).

- Los modelos OrBAC [164] son una extensión de RBAC con nuevos niveles de abstracción para flexibilizar el modelo.

Más allá de los modelos de control de acceso, hay una serie de propuestas que se basan en protocolos de control de acceso conocidos en otros escenarios, normalmente utilizados para plataformas de Internet. Algunos de estos sistemas conocidos son Extensible Access Control Markup Language (XACML) [165], OAuth [166], y más recientemente, UMA [167].

- XACML es un lenguaje basado en eXtensible Markup Language (XML) diseñado para modelar políticas de control de acceso, e incluye un esquema para su utilización. Se ha utilizado en plataformas como [168] o [169]. Su uso está muy extendido para ofrecer esquemas de autorización, pero no ofrece algunas de las ventajas de OAuth como el control de acceso delegado, clave en plataformas como la presentada en este trabajo.

- OAuth por su parte es un marco de trabajo diseñado para ofrecer un esquema de control de acceso a servicios Web y aplicaciones. Es, probablemente, el sistema más utilizado en Internet hoy en día (en sus versiones 1.0 y 2.0 [170]), y por tanto, existen bastantes aproximaciones para su aplicación en IoT. Por ejemplo, [171], o [172], donde se propone aplicarlo sobre el protocolo CoAP. En [173] se utiliza sobre MQTT. Sin embargo, no es una aplicación que se pueda hacer directamente, ya que requiere adaptación, y aun así, presenta ciertas restricciones [174].

- UMA es una especificación que pretende evolucionar OAuth extendiéndolo el caso de uso de éste a escenarios más amplios. No se ha definido específicamente para usarse en plataformas IoT, pero su flexibilidad hace que sea interesante para su adaptación. En la sección 3.5.1 se realizará una definición más precisa de este sistema.

Otros trabajos se basan en otras aproximaciones menos comunes. Por ejemplo, en [175] se propone un servicio de autenticación para CoAP en entornos con restricciones energéticas, y en [176] utilizan "Security Assertion Markup Language" (SAML) sobre XMPP. Finalmente, algunas tendencias actuales proponen el uso de sistemas distribuidos como blockchain para el control de acceso [177].

3.3. Análisis de posibles amenazas en la plataforma IoT

Para cada una de las entidades físicas que componen la plataforma, existen riesgos y potenciales amenazas que pueden comprometer el sistema. Por tanto, es necesario identificar estas amenazas y a continuación diseñar soluciones y mecanismos que las eviten en la medida de lo posible.

En los siguientes apartados se define un análisis de las principales amenazas que se han identificado en la plataforma, clasificadas por el lugar donde se pueden producir. Este estudio no pretende ser exhaustivo de todos los posibles riesgos de seguridad que pueden producirse en este tipo de plataformas, si no que se centra en aquellas amenazas más probables y que requieren la adopción de estrategias de diseño previas a la construcción de los elementos que componen la plataforma.

Se han clasificado estas amenazas en tres grandes grupos, dependiendo de la parte de la plataforma a la que afectan. Así, se han identificado amenazas en las comunicaciones entre Smart Toys y gateways, en las comunicaciones entre gateways y clientes, y entre los servidores y los dispositivos que se conectan a ellos.

3.3.1. Amenazas en las comunicaciones entre Smart Toys y gateways

Las comunicaciones entre los Smart Toys y los gateways, o de forma más general las comunicaciones ente los Smart Toys y la plataforma son uno de los principales puntos a tener en cuenta, ya que el diseño de los juguetes puede requerir el uso de sistemas de comunicación no estandarizados y el uso de tecnologías limitadas en cuanto a recursos para implementar mecanismos de seguridad. Por otro lado, la naturaleza de los datos generados y el hecho de provenir de actividades infantiles, hacen necesario tener en cuenta especialmente posibles ataques sobre la confidencialidad y la integridad de los datos.

Se han identificado específicamente las siguientes amenazas importantes sobre estos dispositivos y sus comunicaciones:

- Amenaza 1: Un gateway falso podría enviar mensajes a los Smart Toys, que podrían provacar a su vez una Denegación de Servicio (DoS) o un mal funcionamiento de los dispositivos.
- Amenaza 2: Un gateway falso suplantando a uno auténtico podría capturar la información generada por los Smart Toys, lo cual sería una brecha en la confidencialidad y la privacidad de los datos.
- Amenaza 3: Los datos generados por los Smart Toys podrían ser capturados y modificados en su transmisión, por lo que se comprometería la confiabilidad en ellos.

3.3.2. Amenazas entre gateways e interfaces de usuario cliente

Los gateways, tal y como se ha visto en el capítulo 2, son el punto de acceso habitual a la plataforma por parte de los usuarios, en cuanto a la gestión de los Smart Toys y las actividades que se pueden realizar. Por tanto, son un punto de paso tanto de la información de que reciben de los Smart Toys como de la información que reciben a partir de los interfaces de usuario, y la información que envían a estos y a los servidores a través de Internet.

Las principales amenazas relacionadas con esta parte de la plataforma son:

- Amenaza 1: Un atacante podría usar la API REST de un gateway a partir de un cliente falso para ganar acceso a los datos o modificar la configuración del gateway o los Smart Toys. Esta misma amenaza se puede aplicar también a accesos a través de clientes que utilicen protocolos de publicación/suscripción.
- Amenaza 2: Un atacante podría obtener los ficheros de datos generados a partir de los datos de Smart Toys a través de un cliente conectado a la API o suscribiéndose a un canal, o incluso mediante el acceso físico al dispositivo gateway.
- Amenaza 3: Un atacante podría capturar los mensajes intercambiados entre el gateway y un cliente, obteniendo así datos confidenciales.

3.3.3. Amenazas en las comunicaciones ente gateways, clientes y servidores

Además de los puntos de acceso a los dispositivos (Smart Toys, gateways, clientes de usuario) que se han visto en las secciones anteriores, existen en la plataforma otras comunicaciones que se realizan a través de Internet, tal y como se vio en la descripción de las entidades físicas de la plataforma. Así, la transmisión de los datos desde gateways o clientes a los servidores, se realiza mediante protocolos estandarizados, lo cual proporciona por sí mismo algunos mecanismos de seguridad (comunicaciones cifradas mediante el uso de TLS/SSL, por ejemplo). Aun así, es necesario definir las principales amenazas que pueden surgir en esta parte del sistema.

- Amenaza 1: La utilización de comunicaciones HTTP no cifradas podría provocar la captura de datos restringidos, tanto datos obtenidos a partir de los Smart Toys como datos que a su vez puedan ofrecer información para el acceso a zonas restringidas en el servidor.
- Amenaza 2: El acceso no autorizado al servidor (incluyendo las bases de datos) puede comprometer el sistema al completo, incluyendo datos personales de los usuarios y de los participantes en las actividades.

- Amenaza 3: El uso no autorizado de los servicios proporcionados por la plataforma podría derivar en el acceso malicioso a datos sensibles.

3.4. Protección de las comunicaciones: Cifrado y transmisiones seguras

Uno de los aspectos que se puede considerar a la vez más importante y más delicado en la plataforma de Smart Toys es la tarea de asegurar las comunicaciones entre los propios Smart Toys y los dispositivos que hacen de gateway o de recolectores de datos durante las actividades. A partir de las amenazas que se han identificado en la sección 3.3, y teniendo siempre en cuenta las limitaciones que presentan o pueden presentar los dispositivos sobre los que se debe aplicar, se ha definido un mecanismo de seguridad que permita proveer a estas comunicaciones de integridad, autenticación y confidencialidad.

Como se ha podido ver en el estudio de la sección 3.2, los sistemas para asegurar la confidencialidad y la autenticación de los mensajes son muy variados y dependen fundamentalmente de los dispositivos y de los protocolos utilizados. En el caso de la plataforma de Smart Toys, las comunicaciones entre éstos y el resto de la plataforma no se realizan utilizando ningún protocolo de aplicación determinado (sí pueden utilizarse entre los gateways y los clientes), si no que se utiliza un sistema de aplicación simple y directo que permite manejar los juguetes desde los gateways. A nivel de infraestructura, por otro lado, se permite el uso de distintas tecnologías que a su vez implican el uso de distintos protocolos.

El mecanismo propuesto se basa en las características específicas de las comunicaciones de los Smart Cubes descritos en el capítulo 2, aunque es posible adaptarlo a los mecanismos de comunicación de otros Smart Toys con distintas capacidades de comunicación o que utilicen protocolos distintos para la misma. Las comunicaciones en estos dispositivos se basan en el sistema NRF24, que no dispone de mecanismos de seguridad propios específicos, por lo que se ha diseñado un mecanismo que aprovecha la división de las comunicaciones en distintos canales para simplificar los mecanismos de autenticación y cifrado. Esto permite tener un sistema que, manteniendo un nivel de seguridad suficiente, puede simplificar estas tareas en los dispositivos, que sufren las

mayores restricciones en cuanto a recursos computacionales y tiempo de respuesta. Las comunicaciones que realizan estos dispositivos se pueden ver resumidas en la Figura 3.1.

Figura 3.1: Mensajes intercambiados durante la actividad.

Como se puede observar, se trata de un mecanismo de comunicaciones simple, en el que existen tres fases principales:

1. Para iniciar la actividad, el gateway envía una señal indicando el inicio de la actividad (START). Este mensaje se envía en forma de difusión a todos los juguetes activos que formen parte de la actividad.

2. Al recibir esta señal, los Smart Cubes comienzan a enviar los datos de sus sensores hacia el gateway. Estos mensajes son confirmados por el gateway mediante mensajes de reconocimiento (ACK).

3. Finalmente, la actividad finaliza por parte del gateway cuando envía un mensaje de finalización (STOP) a todos los Smart Cubes, utilizando una señal de difusión nuevamente.

Aunque este es el mecanismo básico de comunicación entre los Smart Toys y los gateways correspondientes, es posible que en algunos casos existan otros mensajes intercambiados para el control y la gestión del sistema (por ejemplo, mensajes de error, advertencias, como por ejemplo un mensaje indicando baja carga de batería, o actualizaciones en los Smart Cubes).

Los mensajes de inicio y fin de las actividades se envían mediante mensajes de difusión. Esto implica que son recibidos por todos los Smart Toys activos en ese momento. Sería posible en dichos mensajes indicar qué juguetes o grupos de juguetes deben considerarse parte de la actividad, utilizando para ello la información intercambiada durante el emparejado previo que se explicará más adelante, sin embargo, por simplicidad, en este trabajo se ha supuesto que todos los juguetes activos en el momento recibirán la señal y serán parte de la actividad.

El mecanismo propuesto se basa en ciertas características del sistema de comunicaciones utilizado por los Smart Cubes, sin embargo, es extrapolable a otros Smart Toys que utilicen este u otro hardware de comunicaciones, realizando las adaptaciones correspondientes al dispositivo. En este caso, se utilizan los módulos NRF24L01+ de Nordic Semiconductor (véase la descripción de los prototipos en la sección 2.4.2). Este tipo de módulo, que presenta varias ventajas con respecto a otros sistemas de comunicación, tiene una restricción importante: Permite únicamente enviar mensajes de un máximo de 32 Bytes. Por otro lado, estos dispositivos permiten utilizar hasta seis diferentes "canales" virtuales sobre un único canal físico, mediante la utilización de seis direcciones distintas. Estos canales, denominados "pipes", permiten realizar varias conexiones simultaneas con distintos dispositivos, pudiéndose configurar en cada extremo como canales de entrada o de salida.

En el caso que nos ocupa, se fija una de las "pipes" para servir de canal de entrada de datos en todos los Smart Toys y de salida en el gateway (es decir, se fija el canal para el envío de los mensajes de difusión). Las otras 5 "pipes" se asignan de forma aleatoria a cada uno de los dispositivos que conforman la actividad, de forma que exista el mayor equilibrio posible entre el número de dispositivos y el número de canales (por ejemplo, si se utilizan 15 Smart Cubes, se distribuirán uniformemente las "pipes" para que cada una sea utilizada por 3 cubos). Esto implica que es posible que existan más dispositivos en una comunicación que canales unicast durante una actividad. Por tanto, se utiliza una dirección única extra para identificar cada uno de los Smart Toys en la actividad. De esta forma, aunque a nivel del canal, varios dispositivos reciban la misma señal, cada uno sólo tendrá en cuenta aquellas dirigidas a su dirección.

Sobre este mecanismo de comunicaciones, hemos propuesto un método de cifrado y autenticación específico que tiene en cuenta las restricciones de los dispositivos, y

aprovecha las ventajas ofrecidas por los mismos. Dadas sus características, se ha optado por implantar un mecanismo de clave simétrica que, al necesitar menos recursos, es más apropiado para este tipo de sistemas.

3.4.1. Emparejado de dispositivos

En los sistemas basados en clave simétrica, es crucial determinar cómo se va a producir el intercambio previo de claves entre los dispositivos que se comunican, ya que es necesario asegurar que las claves utilizadas son suficientemente seguras para el correcto funcionamiento del sistema criptográfico.

En el caso de los Smart Toys, se propone un sistema de emparejado de los dispositivos con un gateway previo a su utilización conjunta. Este emparejamiento se hará utilizando un canal seguro, idealmente, un sistema cableado o un sistema inalámbrico que requiera una cercanía entre los dispositivos (por ejemplo, mediante NFC). En el caso de los Smart Cubes, se puede utilizar el interfaz SPI (Serial Peripheral Interface) para llevar a cabo este proceso.

En este proceso, el gateway envía los siguientes datos de forma segura a cada Smart Toy:

- Número de Serie (NS): Un número único de 128 bits en toda la plataforma (se pueden determinar estos números de forma centralizada desde los servidores en Internet) que se asigna a cada juguete.
- "Master Secret" (ms): Un número aleatorio de 256 bits que permitirá luego la generación del material criptográfico. El valor será distinto para cada juguete, y permitirá la generación de claves específicas entre gateway y dispositivo.
- Una dirección para el Smart Toy (add): Un valor único para cada Smart Toy que será utilizado durante la comunicación. El valor sólo necesita ser único dentro del grupo de juguetes involucrados en estas comunicaciones, por lo que se pueden utilizar valores numéricos entre 1 y 15 por ejemplo para un conjunto de 15 Smart Cubes. Esto evita enviar los mensajes unicast utilizando el número de serie, enviando en su lugar un campo mucho más pequeño en la cabecera del mensaje.

- Un número de registro del juguete en el gateway. Es un número de serie local que permite al gateway llevar un control sobre el número de juguetes emparejados en cada momento. Se utilizan 8 bits para este número, por lo que un mismo gateway puede tener registrados hasta 256 Smart Toys en un momento dado.

Los datos recibidos en el Smart Cube se almacenan en la memoria no volátil del mismo, y no es posible realizar otro proceso de emparejamiento hasta que se desempareje del gateway actual. Estos datos son almacenados de forma segura en dicha memoria, utilizando para ello un mecanismo de bloqueo de su lectura. En el microcontrolador utilizado por los Smart Cubes, por ejemplo, este mecanismo viene incorporado en el propio integrado y se puede activar reseteando dos bits específicos [90]. Si alguien trata de leer esa memoria de forma externa al dispositivo, ésta se borrará automáticamente, por lo que no será posible comunicarse con el resto de la plataforma.

De igual forma, los datos enviados por el gateway son calculados en éste (excepto el número de serie, recibido de los servidores) y almacenados de forma segura en una base de datos cifrada.

3.4.2. Generación de material criptográfico

Una vez que el gateway y los Smart Toys están emparejados, es posible iniciar una actividad mediante el envío de un mensaje de inicio. Para proveer esta comunicación con un mecanismo que asegure la autenticación y la confidencialidad, es necesario generar las claves y otros elementos criptográficos que permitan cifrar y autenticar el contenido.

Concretamente proponemos la generación de los siguientes elementos criptográficos:

- Clave de sesión para autenticación (K_{as}, 128 bits).
- Clave de sesión para cifrado (K_{cs}, 128 bits).
- Vector de inicialización (IV, 128 bits).

La generación de este material se basa en una función derivada de la utilización de

una función *hash* (h) (para ofrecer un grado de seguridad mayor, esta función h puede llevarse a cabo mediante la aplicación de una función de derivación de clave estándar (KDF) en lugar de una función *hash*), sobre los valores intercambiados durante el proceso de emparejado. Las funciones concretas se pueden ver en las ecuaciones 3.1 y 3.2, que determinan la generación de las claves para la autenticación de mensajes de difusión (K_{asB_p}) y unicast (K_{asU_p}), respectivamente:

$$K_{asB_p} = h((seed \oplus \text{expand}_{128}(p)) \parallel ms1), 1 \leq p \leq 5 \tag{3.1}$$

$$K_{asU_p} = h((seed \oplus \text{expand}_{128}(p)) \parallel ms1 \oplus ms2), 1 \leq p \leq 5 \tag{3.2}$$

Donde:

- *seed* es un valor aleatorio de 128 bits que se genera en el gateway y se envía en el primer mensaje (inicio del experimento, START). Esto permite tener una semilla (*seed*) aleatoria distinta para la generación de las claves de sesión en cada caso.
- p es una de las posibles "pipes" o canales de comunicación entre el gateway y los Smart Cubes, tal y como se vio en la introducción a este apartado. Es un valor entre 1 y 5, y, tal y como se muestra en las ecuaciones, esto implica que habrá hasta 5 claves de cada tipo, una por cada canal.
- *ms1* y *ms2* son las dos mitades del valor *ms* compartido durante el emparejamiento, estando cada uno compuesto de los primeros 128 bits y los últimos 128 bits respectivamente.
- $expand_{128}$ es una función que devuelve el valor representado como un número de 128 bits, para poder llevar la operación.

Cada operación de XOR ($\oplus$) entre el *seed* y el valor de p se pasa a la función *hash* h y al resultado de la misma se aplica una operación de concatenación ($\parallel$) con los valores obtenidos a partir del "Master Secret" ($ms1$). La función *hash* a aplicar puede ser cualquiera que admita el número de bits de entrada, por ejemplo, el algoritmo de SHA3-256 [178].

Para el cálculo de las claves de cifrado (K_{c_p}), se utiliza un mecanismo similar, pero utilizando la segundaparte del "Master Secret" ($ms2$). En las ecuaciones 3.3 y 3.4 se muestra este proceso:

$$K_{c_p} = h((seed \oplus expand_{128}(p)) \parallel ms2), 1 \leq p \leq 5 \tag{3.3}$$

Este valor K_{c_p}, de 256 bits, se obtiene para cada p, y a partir de él se obtiene tanto la clave de cifrado correspondiente (K_{cs_p}) como el vector de inicialización (IV_p) correspondiente a cada canal, partiendo en dos el resultado. En la ecuación 3.4 se puede ver cómo el valor K_{c_p} es la concatenación de estos dos valores:

$$K_{c_p} = K_{cs_p} \parallel IV_p \tag{3.4}$$

3.4.3. Protección de mensajes de difusión

Los mensajes de difusión enviados desde el gateway permiten recibir una orden por parte de todos los Smart Toys activos (por ejemplo, de inicio o fin de una actividad) utilizando un único mensaje. Autenticar el origen de estos mensajes permite asegurar que el emisor es quien dice ser y por tanto, que no se pueda modificar el comportamiento de los Smart Toys por una tercera parte. Esta protección se basa en la generación de la clave de autenticación correspondiente ($KasB_p$), dependiente del canal de comunicación p seleccionado, tal y como se ha descrito anteriormente y se ha mostrado en la ecuación 3.1. Esta clave es generada en el Smart Cube al recibir el mensaje de inicio de la actividad, que incluye el campo *seed* utilizado en su generación.

A continuación, se utiliza un código de autenticación de mensajes ("Message Authentication Code", MAC) para asegurar la integridad y la autenticidad de los mensajes. Concretamente, proponemos el uso de una función de tipo KMAC [179] que permite la generación de estos mensajes de forma segura, basándose en el algoritmo SHA-3. El uso de SHA-3 tanto en esta operación como en el cálculo de las claves de autenticación y cifrado puede ofrecer ventajas en cuanto a la reutilización del mismo sistema en dispositivos con hardware limitado. Sin embargo, el sistema puede adaptarse a otros

algoritmos tanto para una tarea como para la otra.

Como se ha comentado anteriormente, se genera una clave de autenticación para mensajes de difusión por cada canal de comunicaciones, por lo que, a la hora de transmitir estos mensajes de difusión, o bien se transmiten todas las posibles claves de sesión en el mismo mensaje, o es posible que no se puedan autenticar los mensajes en todos los dispositivos destino.

Para evitar este problema, se propone un método basado en lo que llamamos "sliced MAC". Es decir, utilizar parte del código MAC calculado a partir de cada una de las posibles claves $KasB_p$, concretamente dos bytes del valor completo. El tamaño del MAC completo será por tanto de 10 Bytes. Este tamaño es el máximo que se puede enviar en un mensaje a través de un dispositivo como el que utiliza el Smart Cube, con 32 Bytes de tamaño máximo. Para calcular cada uno de los trozos de un "sliced MAC" (MAC_p), se utiliza la ecuación 3.5.

$$MAC_p = \text{reduce}_{16}(\text{KMAC}(K_{asB_p} \parallel m)), 1 \leq p \leq 5 \tag{3.5}$$

Donde p es el identificador de "pipe", m es el mensaje a enviar incluyendo todos sus bytes (el campo reservado para el MAC se rellena previamente con ceros), KMAC es la función de cálculo del MAC, y reduce_{16} es una función que obtiene los últimos 16 bits del resultado anterior.

Cada valor MAC_p se puede considerar una parte del MAC completo, por lo que el valor del "sliced MAC" sería la concatenación de los valores anteriores, tal y como se muestra en la ecuación 3.6.

$$MAC = \|_{1 \leq p \leq 5}\ MAC_p \tag{3.6}$$

En la Figura 3.2 se puede ver el formato de los mensajes de difusión y cómo se utiliza en ellos el valor anterior. Tal y como se ve en la figura, los 10 bytes del MAC se envían en el mensaje inicial junto con el campo *seed*, de 16 Bytes. La cabecera se completa con la dirección de destino (2 bytes, en este caso indicando que es un

mensaje de difusión), 2 bytes indicando el tipo de mensaje enviado, y otros dos bytes que identifican el tipo de Smart Toy al que se dirigen.

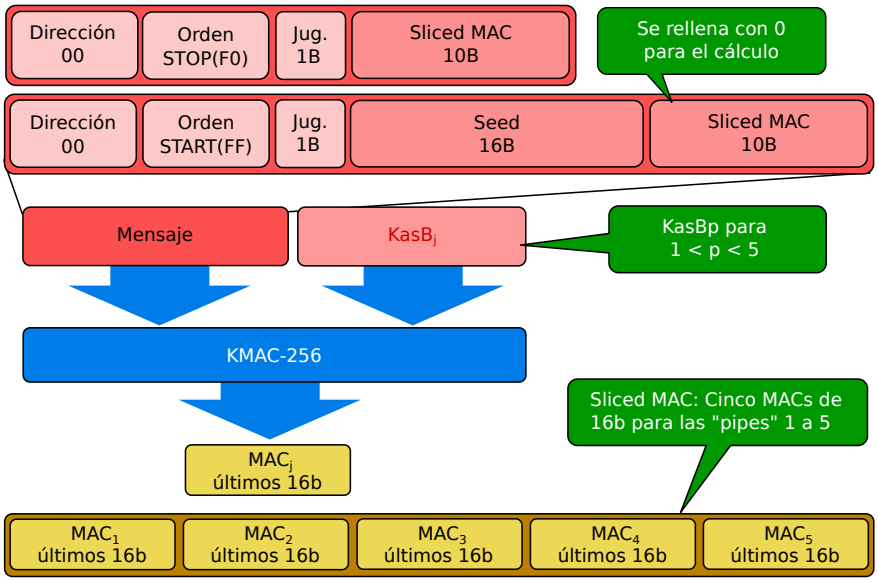

Figura 3.2: Esquema del uso del "sliced MAC" en mensajes de difusión

Cuando el mensaje es recibido en cada uno de los destinos, se calcula en los dispositivos el valor de la clave de autenticación, y, junto con el contenido del mensaje, se vuelve a realizar el cálculo de la función KMAC (ecuación 3.5). El resultado (concretamente los últimos 16 bits del mismo), se compara con el trozo de "sliced MAC" correspondiente para autenticar el mensaje.

3.4.4. Protección de mensajes de control unicast

De manera similar a como se han protegido los mensajes de difusión, es necesario autenticar los mensajes unicast enviados hacia los Smart Toys (por ejemplo los mensajes de reconocimiento (ACKs) enviados por cada mensaje de datos recibido).

Al igual que en el caso anterior, el mecanismo se basa en la generación de un código MAC a partir de la función KMAC presentada anteriormente. Por tanto, se puede calcular cada MAC_p en este caso utilizando la misma ecuación 3.5 definida para los mensajes de difusión. Sin embargo, en este caso, se puede utilizar el MAC completo tal y como se calcula en dicha ecuación. Es decir, tal y como se ve en la ecuación 3.7,

el código MAC utilizado en este caso es igual al código MAC_p para la "pipe" concreta utilizada en la comunicación unicast.

$$MAC = MAC_p \tag{3.7}$$

El esquema utilizado para la autenticación de los mensajes unicast se puede observar en la Figura 3.3. Se puede ver que el mecanismo es similar al caso anterior. Se genera el MAC correspondiente y se envía junto al mensaje. Posteriormente en el dispositivo de destino, se calcula la clave K_{asU_p}, y a partir de ella se ejecuta la función KMAC. El resultado se compara con el MAC para asegurar el origen del mensaje.

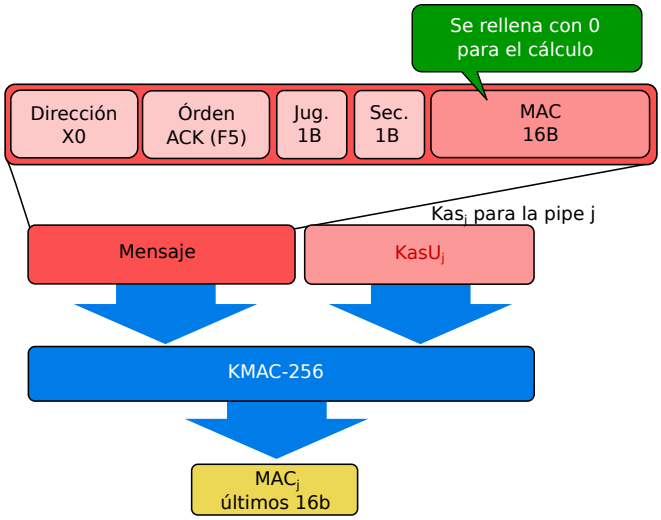

Figura 3.3: Esquema de autenticación KMAC para mensajes unicast.

3.4.5. Protección de mensajes de datos

En las secciones anteriores se ha abordado la problemática relacionada con la autenticación de los mensajes recibidos por los Smart Toys, con el objetivo de evitar que un atacante pueda suplantar a un gateway y gestionarlos de forma maliciosa.

Sin embargo, es necesaria una operación más de protección de las comunicaciones entre estos dos dispositivos, para impedir que un atacante pueda interceptar y obtener datos de las actividades. Recordemos que estos datos pueden ser muy sensibles ya

que, mediante un análisis posterior, podrían potencialmente dar información acerca de los niños que están realizando los juegos. Como se ha indicado anteriormente, para evitar esto se ha utilizado un sistema de cifrado basado en clave simétrica. El algoritmo concreto a utilizar debe depender de las restricciones de los Smart Toys. En las pruebas realizadas con los Smart Cubes se ha utilizado AES en modo CBC, que permite una operación de cifrado sin comprometer el rendimiento del microprocesador.

En la Figura 3.4 se puede ver el mecanismo de cifrado y descifrado de los mensajes de datos. Antes de proceder al cifrado del mensaje, se calcula un *checksum* que permita una comprobación de integridad de los datos al descifrarse. Este valor se obtiene de realizar una operación XOR sobre todos los bytes del mensaje (excepto, obviamente, el propio campo de *checksum*).

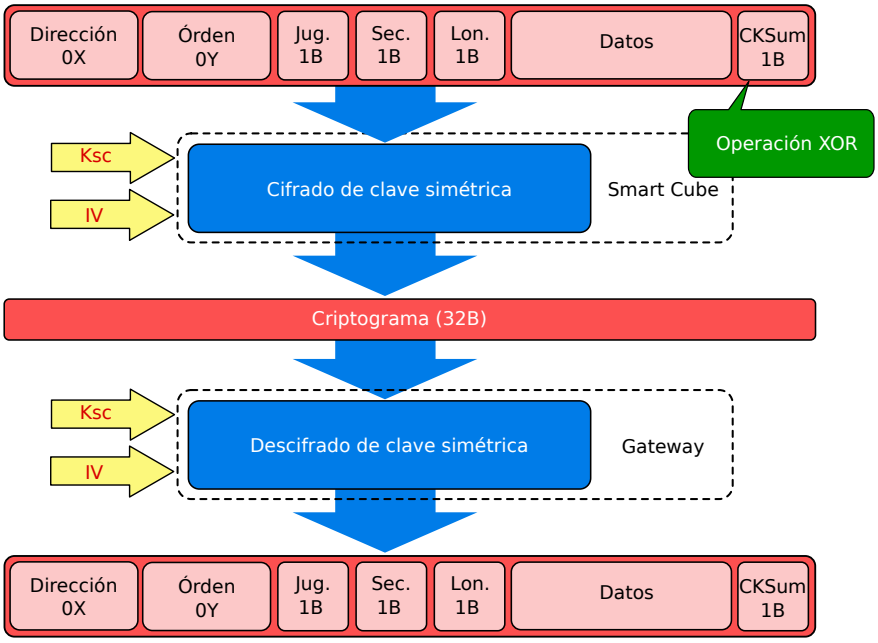

Figura 3.4: Esquema del cifrado y descifrado de los mensajes.

Por otro lado, se utilizan tanto la clave de cifrado como el vector de inicialización Kcs e IV, calculados tal y como se describió en las ecuaciones 3.3 y 3.4 como valores de entrada de la función de cifrado E. Estos valores pueden ser calculados en ambos extremos de la comunicación a partir del campo *seed* compartido al inicio de la actividad.

En la ecuación 3.8 se define el proceso de cifrado, y la función inversa utilizada para descifrar los mensajes se puede ver en la ecuación 3.9.

$$C_p = E(M_p, Kcs_p, IV_p) \tag{3.8}$$

$$M_p = D(C_p, Kcs_p, IV_p) \tag{3.9}$$

3.4.6. Seguridad en los dispositivos gateway

Los dispositivos gateway se encargan fundamentalmente de tareas de gestión de los Smart Toys y las actividades relacionadas, y funcionan como pasarela para el envío de los datos a los servidores en Internet. Es por tanto fundamental que los datos que pasan a través de estas pasarelas estén igualmente protegidos, tal y como lo están las comunicaciones entre los Smart Toys y los gateways.

Para proteger los datos almacenados en los gateways, han modelado ficheros estructurados que contendrán los datos recogidos en cada actividad. Estos ficheros se han cifrado en su creación con un sistema basado en una clave simétrica de un solo uso (K_s). Esta clave de 128 bits se genera por cada fichero de forma aleatoria, y se utiliza para cifrarlos mediante el algoritmo AES en modo CBC.

Para poder descifrar los ficheros en su destino, es necesario compartir la clave. Esto se hace mediante el cifrado de la misma a partir de un esquema de clave asimétrica RSS. Se utiliza la clave pública del servidor destino de 2048 bits (K_{p_s}) para cifrar la clave de un solo uso, así como el vector de inicialización (IV) utilizado en el cifrado, tal y como se puede ver en la ecuación 3.10.

$$C_{K_s} = E((K_s||IV), K_{p_s}) \tag{3.10}$$

A continuación, esta clave cifrada se incluye como parte del fichero de datos, concatenada con los datos cifrados, generándose el mensaje cifrado completo (C). En la

ecuación 3.11 se puede ver esta operación, donde M son los datos originales a cifrar.

$$C = E(M, K_s, IV)||C_{K_s} \tag{3.11}$$

El proceso de descifrado se hace según las ecuaciones 3.12 y 3.13, que muestran cómo se descifra primero la clave de un solo uso y luego se utiliza para descifar el archivo.

$$K_s, IV = D(C_{K_s}, K_{p_s}) \tag{3.12}$$

$$M = D((C - C_{K_s}), K_s, IV) \tag{3.13}$$

Además de los mecanismos de cifrado de los datos, los gateways incorporan otros mecanismos de seguridad imprescindibles para asegurar completamente el sistema de intercambio de datos. Así, por ejemplo, todas las comunicaciones realizadas a través de APIs REST o de peticiones Web se llevan a cabo mediante HTTPS (es decir, cifradas mediante SSL/TLS). También se cifran todas las peticiones realizadas a través de los protocolos de intercambio de mensajes (MQTT, AMQP) mediante la utilización de TLS. Finalmente, y dado que el gateway puede funcionar como un punto de acceso Wi-Fi, se protege esa conexión mediante el protocolo WPA2.

3.5. Protección de los datos de usuarios: Privacidad y control de acceso

En las anteriores secciones se han detallado los mecanismos diseñados para ofrecer integridad, confidencialidad y autenticación en las comunicaciones, especialmente en los puntos "más débiles" de la plataforma, como pueden ser las comunicaciones entre los Smart Toys y los dispositivos que reciben sus datos.

Existe sin embargo una cuestión fundamental en la plataforma, y es el control del

acceso a los datos generados por los dispositivos y a la información ofrecida en ella. En este tipo de plataformas se generan una gran cantidad de datos heterogéneos y no todos ellos se comparten utilizando los mismos métodos de comunicación. Además, es importante que no todos los usuarios tengan acceso a todos los datos, por lo que se hace necesario implementar algún sistema de autorización y control de acceso.

Existen muchos esquemas de control de acceso disponibles en la literatura, tal y como se muestra en la sección 3.2. pero no todos estos sistemas son aplicables directamente a un entorno IoT, donde existe esta heterogeneidad en el manejo de la información. En esta sección se propone un sistema de control de acceso único para toda la plataforma, que permita extender los esquemas utilizados en entornos Web y de servicios a entornos IoT donde se tienen otros tipos de comunicación. Para ello, se modelan los canales de comunicación IoT como recursos y se protegen de forma similar a como se protegen los recursos de una API HTTP.

3.5.1. Tecnologías utilizadas

Antes de proceder a definir y explicar la propuesta que se hace en este trabajo sobre un sistema unificado de control de acceso, se explican los fundamentos de las tecnologías utilizadas para su diseño. Por un lado, se hace una breve descripción de los elementos fundamentales que componen un protocolo de publicación/suscripción de mensajes, ya que la propuesta se basa en la extensión de un mecanismo orientado a plataformas Web a este tipo de sistemas. La descripción se hace en base al protocolo MQTT, uno de los más populares y extendidos actualmente en este tipo de plataformas, pero se puede hacer extensivo a otros protocolos que usen el mismo esquema basado en un Broker de gestión de mensajes (por ejemplo, AMQP).

Además, se presenta una descripción del mecanismo de control de acceso utilizado como base para la propuesta, User-Managed Access (UMA), que extiende las funcionalidades de OAuth para aproximarlo a otros escenarios más complejos de control de acceso.

3.5.1.1. Protocolos de publicación/suscripción: MQTT

En la sección dedicada al estado del arte relacionado con los aspectos de seguridad, se han listado ya algunos de los principales protocolos utilizados actualmente en plataformas IoT para modelar comunicaciones basadas en los mecanismos de suscripción y publicación. Este tipo de comunicaciones son muy utilizadas en este tipo de sistemas para ofrecer una mayor flexibilidad y escalabilidad en comunicaciones formadas por una gran cantidad de datos y muchos mensajes. Se basan en la creación de canales de comunicación en los que los dispositivos, servicios o usuarios pueden acceder para enviar información (publicar) o recibirla (suscribir) sin necesidad de un esquema de solicitudes y respuestas.

La mayoría de estos protocolos se basan en un elemento centralizado que permite la gestión de las colas de mensajes entre los distintos extremos de la comunicación.

Uno de los protocolos más populares que siguen este esquema es MQTT (que ha sido estandarizado recientemente [180]). Diseñado originalmente por IBM para el intercambio de información de telemetría, ha sido ampliamente adoptado por redes de dispositivos basados en sensores y plataformas IoT, entre otras cosas por su diseño orientado a la minimización del uso del ancho de banda y consumo energético. En la descripción de la propuesta se ha utilizado este protocolo como ejemplo de sistema de suscripción y publicación, por su amplio uso y por su parecido con otros protocolos similares como AMQP o XMPP.

En MQTT cada canal de comunicaciones utilizado se denomina “topic”. Un “topic” es un canal etiquetado que permite a los servidores enrutar los mensajes de su origen a sus destinatarios.

Pese a las ventajas de usar este protocolo, MQTT no se diseñó originalmente teniendo en cuenta la seguridad. El protocolo ofrece algunas características de seguridad como la posibilidad de utilizar autenticación basada en parejas de usuario y contraseña como credenciales, pero no especifica ningún modelo específico de autorización o control de acceso. En el estándar se recomienda el uso de otras técnicas como VPN o TLS para asegurar las comunicaciones a través de MQTT.

En la Figura 3.5 se muestran las principales entidades funcionales que conforman un sistema que utiliza el protocolo MQTT. Estas entidades son:

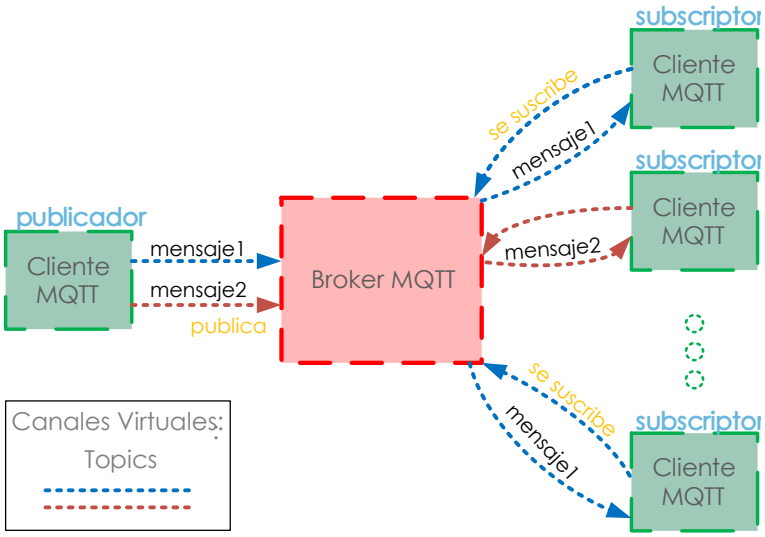

Figura 3.5: Principales entidades que forman las comunicaciones en MQTT.

- "Broker": Es un servidor central que se encarga del enrutado y la gestión de los mensajes intercambiados a través de MQTT.
- Publicadores: Son entidades cliente que utilizan uno o varios "topics' para enviar datos a otros clientes que se han suscrito al canal.
- Suscriptores: Son otras entidades cliente que se suscriben a uno o varios "topics" para recibir información a través de ellos. Cada uno de los clientes puede funcionar a la vez como publicador y suscriptor.
- "Topics": Como se ha adelantado ya, cada "topic" se puede considerar un canal de comunicación que identifica la temática de una serie de mensajes intercambiados. Estos canales se etiquetan siguiendo una estructura jerárquica, donde se tienen varios niveles temáticos separados por barras (/). Por ejemplo, un "topic" en MQTT podría ser: */casa/cocina/temperatura*).

Un aspecto importante del protocolo es que, al no especificar ningún esquema de autorización o control de acceso a la información, no incluye campos en sus cabeceras

para estas tareas. Aun así, en el mensaje de tipo CONNECT (mensaje utilizado por el protocolo para establecer una nueva conexión con un "broker"), se puede incluir una cabecera opcional de autenticación (compuesta de dos campos: usuario y contraseña). Esta cabecera se envía sin cifrar salvo que se utilice algún sistema de cifrado por encima del protocolo (por ejemplo, TLS como se ha indicado anteriormente). Esta cabecera se puede aprovechar para enviar otro tipo de información (por ejemplo, tokens o tickets) que permita incluir el sistema de control de acceso sin necesidad de realizar una modificación del protocolo o utilizar otro protocolo extra sobre él. Esta ventaja se usa en nuestra propuesta como se verá más adelante.

3.5.1.2. El perfil de control de acceso UMA

En nuestra propuesta para el control de acceso hemos optado por la utilización del perfil User-Managed Access (UMA) para determinar los procesos y flujos de mensajes que garanticen el registro adecuado de los recursos y su acceso controlado mediante el otorgamiento de los permisos correspondientes. UMA se puede ver como una evolución del esquema propuesto por OAuth 2.0, para su utilización en escenarios no cubiertos por el estándar. OAuth es un sistema ampliamente utilizado en servicios basados en Internet que permite a un usuario que posee cierta aplicación o servicio dar permiso a otra aplicación o servicio de su propiedad para compartir cierta información.

Sin embargo, en OAuth no se define cómo se puede establecer el control del acceso a los recursos. Por ejemplo, cómo un usuario podría dar permiso a una tercera parte (de la que él no tiene control) para la utilización de los recursos a su nombre (definiendo recurso como cualquier contenido susceptible de ser compartido a través de Internet: datos personales, servicios, etc.).

UMA se ha definido específicamente para ofrecer una solución a estos escenarios no cubiertos inicialmente por OAuth, utilizando el estándar para ello. UMA está en desarrollo actualmente, por parte de un grupo de trabajo integrado en "Kantara Initiative" [181], una organización sin ánimo de lucro compuesta por varias empresas y comunidades de desarrolladores. La especificación de UMA está en proceso de estandarización por la "Internet Engineering Task Force" (IETF) [182].

Las entidades funcionales que componen UMA se pueden ver en la Figura 3.6. Estas entidades son:

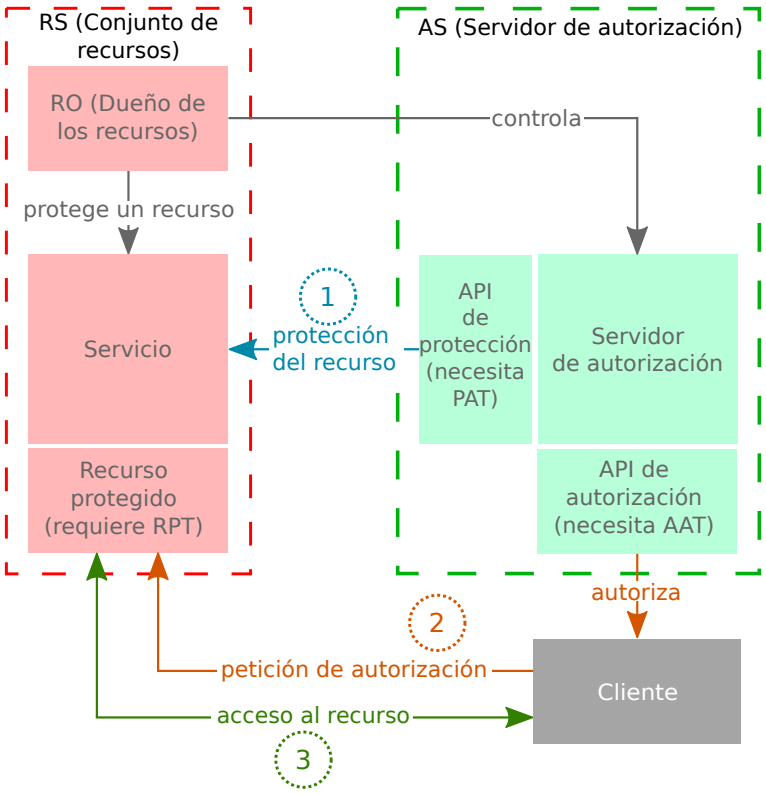

Figura 3.6: Entidades funcionales y fases en el funcionamiento del perfil User-Managed Access (UMA).

- "Resource Owner" (RO), dueño del recurso: Es la entidad que representa al dueño del recurso a proteger. Se encarga de garantizar o denegar el acceso al recurso. Normalmente se trata de un usuario.
- "Requesting Party" (RP), parte solicitante: Es la entidad que representa ala parte que solicita el acceso al recurso.
- "Client", cliente: Representa cualquier aplicación utilizada por los RPs para realizar solicitudes de acceso a recursos protegidos por un RO.
- 'Resource Set" (RS), conjunto de recursos: Se define como un grupo compuesto por uno o más recursos a proteger de forma conjunta (es decir, las políticas de acceso serán las mismas para todos ellos).

- "Resource Server", servidor de recursos: Es la entidad encargada de implementar las tareas de gestión derivadas de la protección de los RS por parte de sus respectivos ROs.
- 'Authorization Server" (AS), servidor de autorización: Es la entidad encargada de emitir los tokens de autorización y los permisos correspondientes a los Clientes. Se encarga de asegurar la protección de los RSs localizados en los Servidores de Recursos. Se compone de dos APIs protegidas a su vez por OAuth, y un interfaz de gestión utilizado por los ROs.

El proceso de protección de recursos planteado por UMA se basa en la utilización de tres tokens distintos. En la Tabla 3.1 se pueden ver cada uno de ellos y la forma en la que se utilizan.

Tabla 3.1: Tokens utilizados en el perfil User-Managed Access (UMA)

Token	**Usado por**	**Para acceder a**	**Objetivo/s**
Authorization API Token (AAT)	Cliente	API de autorización (AS)	Solicitar un RPT
Protection API Token (PAT)	Servidor de recursos	API de protección (AS)	Registrar recursos o comprobar permisos
Requesting Party Token (RPT)	Cliente	Un recurso en un servidor de recusos	Acceder a un recurso protegido

A partir de la figura y la tabla anterior, las tres principales fases en el proceso de protección y acceso a los recursos son:

1. Protección de un recurso: El dueño (RO) de un grupo de recursos (RS) registra el RS en el servidor de autorización (AS) utilizando para ello la API de protección que ofrece el AS (y utilizando por tanto un PAT válido, obtenido a través de un proceso de OAuth).
2. Solicitud de autorización de acceso: La parte solicitante (RP) utiliza un Cliente para solicitar el acceso al recurso a través del servidor de recursos. Para ello, debe utilizar un AAT válido para acceder a la API de autorización del servidor de autorización (AS), que a su vez le permita conseguir un RPT.
3. Acceso al recurso: Utilizando el RPT válido y los permisos asociados, la parte solicitante (RP) es capaz de acceder al recurso a través del cliente.

3.5.2. Propuesta para el sistema de control de acceso

Dado que el funcionamiento de UMA está definido para servicios y plataformas basados en Internet, proponemos utilizar este mecanismo de manera híbrida, para poder gestionar de forma conjunta los recursos de la plataforma de Smart Toys basados en Internet (por ejemplo, servicios web, datos de los usuarios, etc.) y los recursos provenientes de los sensores (servicios alojados en gateways, datos de los Smart Toys, etc.).

Proponemos modelar el servidor de recursos definido en UMA de forma conjunta con el "Broker" del protocolo de intercambio de mensajes (MQTT en este caso). De esta forma, aseguramos una compatibilidad completa con los estándares de cada protocolo, añadiendo las funcionalidades de autorización y control de acceso necesarias.

Por otro lado, proponemos definir cada "topic" de MQTT (y en general, cada canal de comunicación en este tipo de comunicaciones) como recursos a ser protegidos en el esquema de control de acceso, tal y como se protegerían los puntos de acceso de una API REST en un entorno Web.

Siguiendo el esquema de UMA, si cada "topic" es un recurso, debe estar asociado a un dueño de recurso (RO). Por tanto, para todo "topic" se identificará un dueño encargado y responsable de determinar y administrar las políticas de control de acceso asociadas a éste.

En la Figura 3.7, se puede ver el sistema propuesto, incluyendo las entidades que lo componen y los principales flujos de comunicación entre ellos. Los cuatro principales flujos de comunicación que se muestran en la figura son:

- Interacción de los usuarios (A): Son las interacciones entre los usuarios y cada componente que dispone de una interfaz de usuario: fundamentalmente, el cliente usado para solicitar accesos y acceder a los recursos, el interfaz de gestión del servidor de recursos, y el interfaz de gestión del servidor de autorización.
- Flujo de comunicación de UMA (B): Abarca todas las solicitudes y respuestas intercambiadas entre los componentes que forman parte del mecanismo definido en UMA.

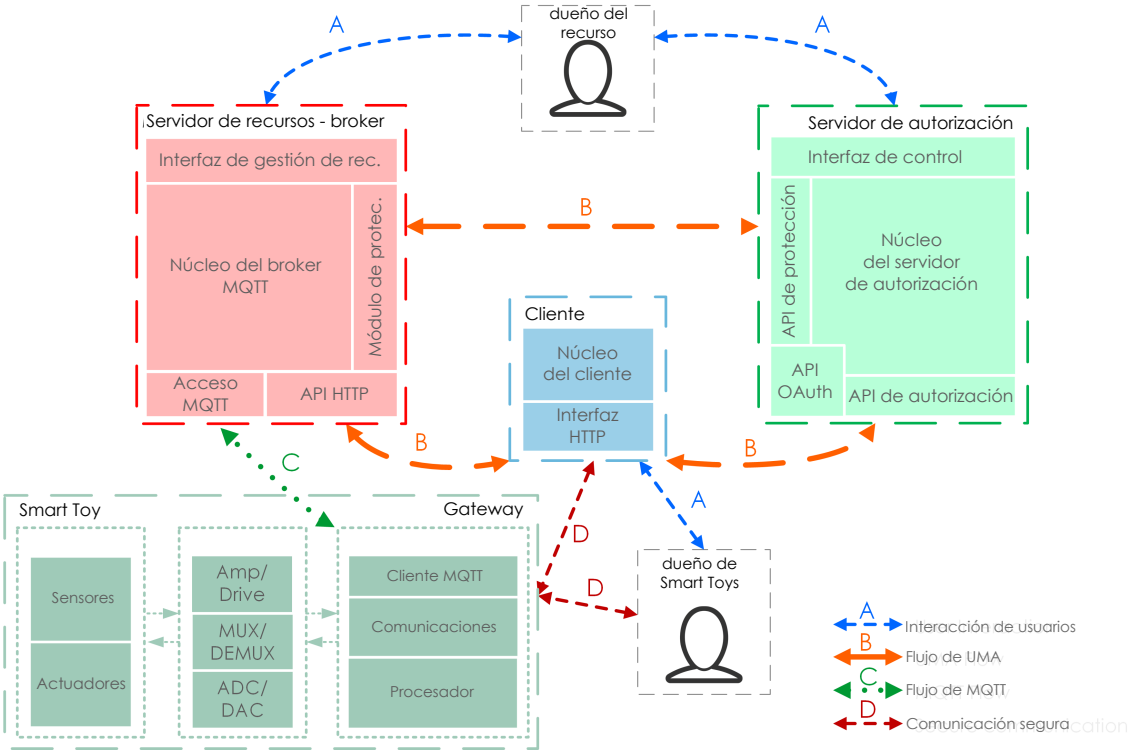

Figura 3.7: Diagrama que muestra las entidades que componen el sistema de control de acceso propuesto y sus flujos de comunicación.

- Flujo de comunicación MQTT (C): Conjunto de mensajes enviados entre los clientes MQTT y el "broker" definido en el protocolo.

- Flujo de comunicación segura (D): Un canal específico seguro para la comunicación entre los Smart Toys (o los gateways, ya que normalmente serán los encargados de ofrecer los datos generados por los juguetes hacia la plataforma) y la persona responsable de su configuración.

3.5.2.1. Entidades funcionales

En esta sección se presenta una explicación de cada una de las entidades funcionales mostradas en la Figura 3.7, especificando los cambios conceptuales sobre las entidades que componen el mecanismo propuesto en UMA (definidos en la sección 3.5.1).

- Recurso: Cada "topic" de MQTT se modela como un recurso en el sentido definido por UMA. Por tanto, un grupo de "topics" con los mismos requisitos en cuanto a autorización serán vistos por el sistema como un "Resource Set" (RS). En este sentido, existirá un conjunto de políticas de acceso asociado a cada conjunto de "topics". Estas políticas se definen en el servidor de autorización (AS) por parte del dueño del "topic", al que se denominará "Topic Owner". Las políticas serán evaluadas para determinar si un solicitante podrá tener permisos para publicar o suscribirse al "topic".

- Servidor de recursos Broker: Como se ha indicado anteriormente, en nuestra propuesta se unen en una única entidad los conceptos de "Broker" MQTT y de servidor de recursos UMA. Esta entidad se encarga tanto de la gestión de los mensajes enviados a través del sistema como de proteger los recursos a través de UMA, implementando las interfaces y APIs necesarias.

- Servidor de autorización: Este componente se define de la misma forma que en las especificaciones de UMA. Es en esta entidad donde se definen las políticas para cada recurso, teniendo en cuenta que las éstas podrán referirse tanto a recursos Web como a recursos definidos como canales de comunicaciones IoT.

- Cliente: Aplicación que permite a los dueños de los Smart Toys o Gateways obtener un RPT válido a través del flujo de autorización de UMA. El dueño de los Smart Toys debe transmitir el RPT al dispositivo mediante un canal seguro, para que éstos puedan actuar de forma independiente en el envío o recepción de los mensajes.

- Smart Toy/Gateway: Se trata de cualquier dispositivo en la plataforma que pueda ser generador o consumidor de datos a través del protocolo de intercambio de mensajes (es decir publicador o suscriptor a un "topic"). En esta plataforma, serán habitualmente los gateways asociados a un conjunto de Smart Toys, o los propios Smart Toys si tienen esta capacidad.

3.5.2.2. Secuencia de mensajes en el sistema

Como se ha descrito anteriormente en la sección 3.5.1.2, y tal y como se puede ver en la especificación de UMA (véase [183]), el proceso de control de acceso se divide en tres fases. En esta sección se describe cada una de ellas de acuerdo a la propuesta de este trabajo.

1. Fase 1: Protección de un "topic": Cuando se define un nuevo "topic" en el sistema, se debe registrar para ser protegido mediante el sistema de control de acceso. El creador del "topic" será considerado como su dueño, por lo que será el encargado de su registro y protección. Desde el punto de vista de UMA, se puede definir al creador del "topic" como el dueño del recurso ("Resource Owner" (RO)).

   A la hora de definir un "topic" es necesario tener en cuenta una serie de consideraciones:

   a) En MQTT los topics siguen un sistema jerárquico en forma de árbol, por lo que un "topic" a proteger debe ser definido desde la raíz del árbol hasta la hoja.
   b) Un "topic" ya definido y protegido no puede volverse a proteger.
   c) Un "topic" hijo no puede ser protegido si el usuario no tiene permisos sobre el "topic" padre.
   d) La protección aplicada a un "topic" específico será aplicada a éste y a todos sus "topics" descendientes.

   Una vez que el topic está definido como un recurso de UMA, el proceso para su protección es el definido por la especificación de UMA.

2. Fase 2: Obtención de autorización: Después de haberse registrado el "topic", cualquier acción de suscripción o publicación tendrá que seguir el flujo de autorización definido en UMA. Esta fase debe comenzarse por parte de un usuario (habitualmente el dueño de un conjunto de Smart Toys y gateway).

   Al finalizar este flujo, el dueño de los dispositivos obtendrá un token RPT que permitirá a los Smart Toys comunicarse con la plataforma a partir de los permisos definidos. Si el dueño de los dispositivos necesita añadir o modificar los permisos

asociados al token, se puede repetir este flujo de mensajes, saltando los pasos innecesarios (por ejemplo, los tokens AAT y RPT estarán ya almacenados en el cliente utilizado, al menos mientras se mantenga abierta la sesión).

3. Fase 3: Envío y recepción de mensajes MQTT: En esta fase se utiliza el RPT adquirido en la fase anterior para acceder al recurso, es decir, en este caso, para obtener la posibilidad de publicar y/o suscribirse a un "topic" determinado.

   El dueño del Smart Toy o gateway debe configurar los dispositivos para que almacenen de forma segura el RPT obtenido, y sea utilizado en las comunicaciones de forma autónoma, sin necesidad de que el usuario realice ninguna acción más.

   En la Figura 3.8 se puede ver la secuencia de mensajes de nuestra propuesta.

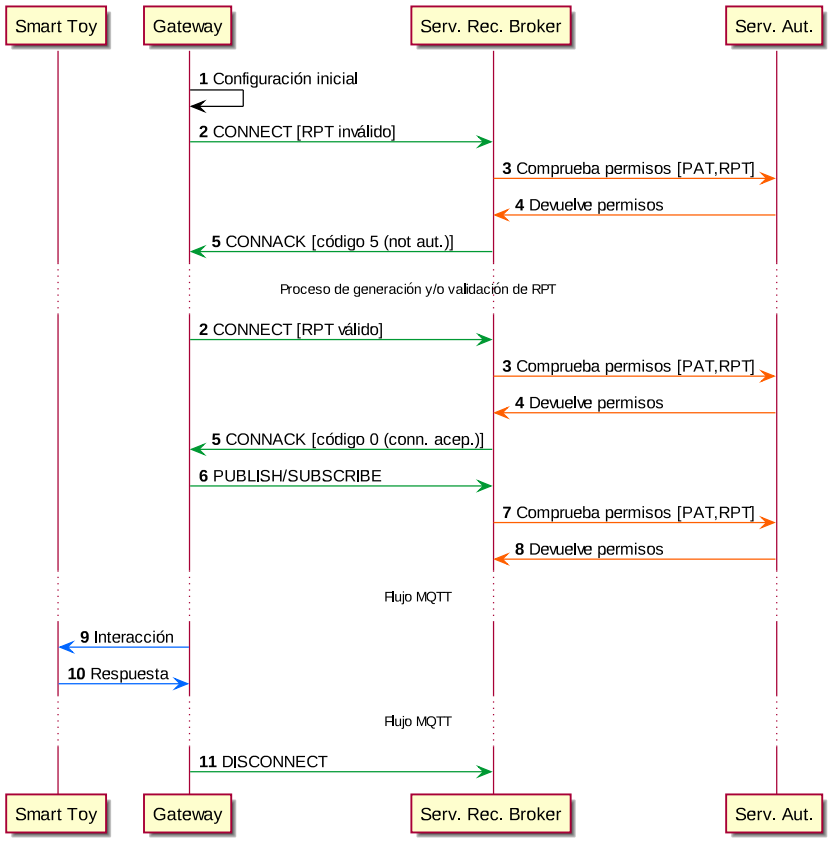

Figura 3.8: Diagrama de secuencia del acceso a un canal MQTT protegido.

1. Configuración inicial: El Smart Toy o gateway debe contener la información definida por su dueño (por ejemplo, localización del servidor de recursos, localización

del broker, nombre del "topic" a utilizar, etc.). Esta información puede ser fijada durante la fabricación del dispositivo y luego modificada por el dueño. Todos los Smart Toys y gateways tendrán que tener al menos un identificador único de su fabricante, un identificador único del dispositivo en la plataforma y un número de serie (tal y como se vio también en la sección 3.4.1).

2. Envío del mensaje CONNECT: El dispositivo inicia el flujo mediante el envío de este mensaje al servidor de recursos broker. Este mensaje debe incluir en su cabecera el token RPT. Para ello se aprovechará la cabecera de autenticación de MQTT.

3. Comprobación de permisos: El servidor de recursos utiliza el RPT en su solicitud a la API de protección del servidor de autorización (incluyendo el token de acceso PAT). A su vez el servidor realiza un proceso de introspección tal y como lo define OAuth [184], incluyendo la lista de scopes asociados al RPT.

4. Retorno de permisos: El servidor de autorización evalúa las políticas asociadas, para lo cual debe tener en cuenta la identidad del conjunto de recursos para los que se evalúan (que identifica mediante el PAT) y la identidad del solicitante (que se identifica mediante el RPT), así como la lista de scopes incluida en la petición. El servidor de autorización generará una lista de permisos que se devolverán al servidor de recursos broker.

5. Envío de mensaje CONNACK: El servidor de recursos broker asocia estos permisos a la sesión abierta entre el dispositivo que funciona como cliente MQTT y él mismo. Si el RPT que se envió en la solicitud no se ha validado, la respuesta del servidor de autorización en el paso anterior habrá sido una lista vacía de permisos. En ese caso, el servidor de recursos broker responderá con un código 5 en la cabecera del mensaje CONNACK. Este código indica que el dispositivo no está autorizado para realizar la operación y que debe iniciar el proceso de validación del RPT antes de continuar. En la sección 3.5.2.3 se explica con detalle este proceso.

6. Acciones de publicación o suscripción: Una vez que el dispositivo tiene garantizados los permisos adecuados, puede realizar las operaciones de suscripción o publicación de acuerdo a su configuración.

7. Comprobación de permisos: Tal y como se hizo en el paso 3, el servidor de recursos utiliza la petición de introspección de OAuth junto con una lista de scopes relativa a las acciones de publicación y suscripción determinadas.

8. Retorno de permisos: Igual que en l paso 4, el servidor de autorización generará la lista de permisos después de evaluar las políticas correspondientes.

9. Interacción MQTT: A partir de este paso el dispositivo es capaz de publicar y suscribirse al "topic" MQTT correspondiente, dependiendo de las acciones que se deseen realizar.

10. Respuesta: De la misma forma, el "topic" se utilizará para las respuestas que se puedan obtener por parte de los dispositivos.

11. Envío de mensaje DISCONNECT: Cuando se hayan realizado todas las acciones determinadas por la actividad o juego correspondiente, se puede enviar un mensaje de desconexión, con lo que se cerrará la sesión MQTT activa.

3.5.2.3. Validación de tokens de acceso

Los tokens utilizados para garantizar los permisos de publicación o suscripción deben ser validados antes de poder ser utilizados. Cómo generar este tipo de token y cómo se debe almacenar para su uso seguro puede ser solucionado de distintas formas. En el escenario más simple, se pueden generar estos tokens durante la fabricación de los dispositivos (Smart Toys, gateways), e incluidos en su memoria. Sin embargo, este sistema no es siempre viable debido a las características de los dispositivos. En esta sección se propone un mecanismo alternativo para la generación de estos tokens y su validación. Utilizando este sistema, es posible gestionar estos tokens independientemente del despliegue de los dispositivos.

El proceso se inicia a partir de un Smart Toy o gateway que desea incluirse en la plataforma y no ha obtenido un token RPT validado durante su fabricación. También se iniciará este proceso si el RPT utilizado en el establecimiento de una comunicación es rechazado (véase el envío del mensaje CONNACK descrito en la tercera fase explicada en la sección 3.5.2.2).

En la Figura 3.9 se muestran los pasos que componen este proceso. A continuación, se incluye una breve explicación de cada uno de los pasos.

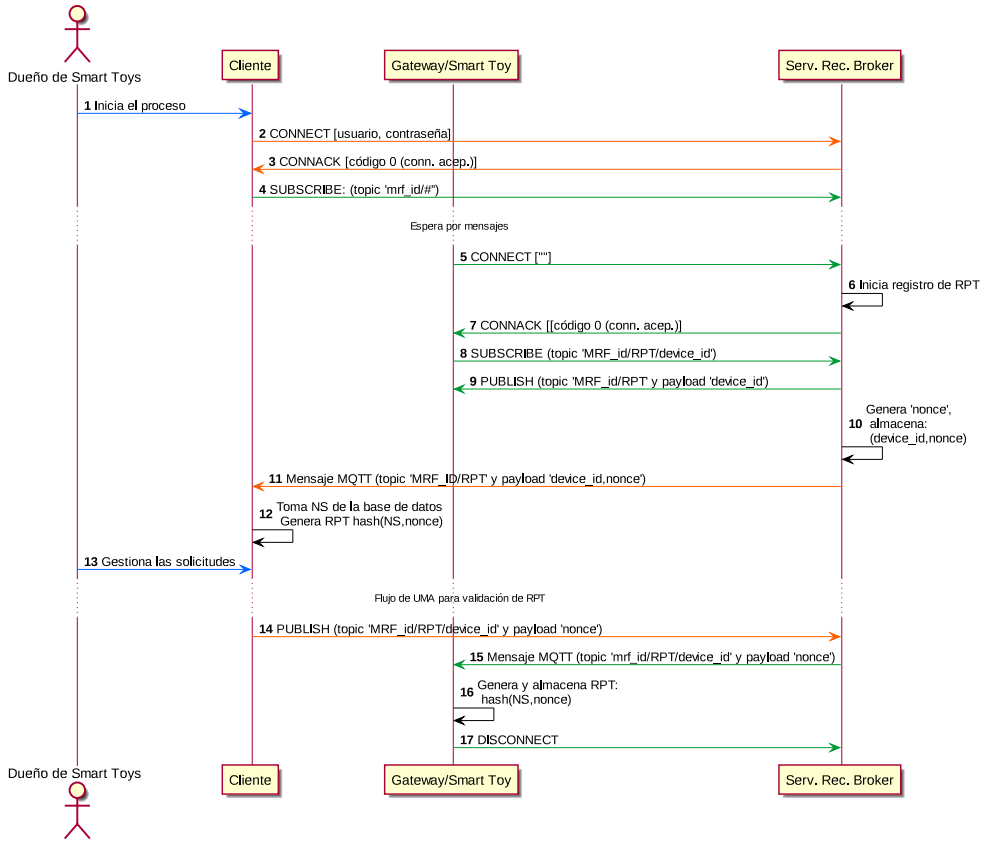

Figura 3.9: Diagrama de secuencia del método propuesto para la validación de tokens RPT.

1. Inicio del proceso: Siguiendo la metodología definida en OAuth 2.0, el dueño del dispositivo es el encargado de iniciar el proceso de validación. Para ello, se debe conectar al cliente a través del cual realizará las gestiones necesarias.

2. Envío de CONNECT desde el cliente: El cliente utiliza un mensaje CONNECT de MQTT hacia el servidor de recursos broker utilizando sus credenciales en el servidor.

3. Envío de CONNACK hacia el cliente: El servidor de recursos comprueba las credenciales del usuario enviados a través del cliente y responde con un mensaje CONNACK y el código 0 si estos son correctos.

4. Suscripción desde el cliente: Una vez que el cliente completa el proceso de conexión con el servidor de recursos broker, se suscribe a un "topic" especifico fijado por el fabricante (etiquetado con el identificador del fabricante, que estará almacenado en el dispositivo. Esta será la única suscripción disponible en el servidor sin necesidad de utilizar un token válido, sólo dependerá de la autenticidad de las credenciales del dueño del dispositivo (que se habrá registrado previamente en la plataforma).

5. Envío de mensaje CONNECT desde el gateway o Smart Toy: Después de que el dispositivo haya intentado realizar una nueva conexión y haya recibido un código de no autorización (código 5), se intenta de nuevo la conexión, pero utilizando una cadena vacía como RPT.

6. Inicio del registro del RPT: Cuando el servidor de recursos broker recibe una solicitud de conexión con el campo RPT vacío, almacena el identificador del cliente y le permite una suscripción y publicación temporal sobre ciertos "topics" específicos.

7. Envío de CONNACK al gateway o Smart Toy: El servidor de recursos broker responde a la solicitud de conexión con un CONNACK y el código 0.

8. Suscripción por parte del gateway o Smart Toy: El dispositivo a continuación se suscribe a un "topic" etiquetado mediante el identificador del fabricante y su propio identificador (es decir un topic identificado como "mrf_id/RPT/device_id". Después espera a que termine el proceso de validación del RPT.

9. Publicación por parte del gateway o Smart Toy: El dispositivo para el que se desea realizar la validación del RPT publica su propio identificador en el topic etiquetado como "mrf_id/RPT".

10. Generación de un "nonce": El módulo de autorización dentro del servidor de recursos broker genera, a partir del mensaje recibido en el "topic" "mrf_id/RPT" un número aleatorio de un solo uso ("nonce") y lo almacena. Este número será incluido en el siguiente mensaje enviado.

11. Envío de mensaje MQTT hacia el cliente: Se envía la petición para la validación del RPT hacia el cliente utilizando el "topic" "mrf_id/#", que fue creado en el paso 4 de este proceso.

12. Generación de un nuevo RPT en el cliente: A partir del identificador del dispositivo, el ciente comprueba el número de serie de éste y, a partir del número de serie y el valor aleatorio "nonce" recibido en el anterior paso, se genera un nuevo token RPT utilizando una función *hash*.

13. Gestión de peticiones: El RPT generado en el paso 12 se almacenará en el cliente hasta que el dueño del dispositivo confirme su validez. Para evitar la tarea de validar cada RPT de forma individual, este proceso se permite hacer para un conjunto de tokens al mismo tiempo. Además, los permisos asociados a un conjunto de tokens pueden haber sido previamente establecidos en el servidor de autorización, facilitando esta tarea al usuario.

14. Publicación desde el cliente: Una vez que se obtiene el RPT y sus permisos asociados, y éste ha sido validado, el cliente comunica la validación correcta mediante la publicación del valor "nonce" en el "topic" correspondiente ("mrf_id/RPT/Device_id").

15. Mensaje MQTT al gateway o Smart Toy: El dispositivo recibe un mensaje a través del "topic" al que se suscribió en el paso 8 de este proceso.

16. Generación del RPT en el gateway o Smart Toy: En este paso el dispositivo sabe que ya se ha validado el RPT, por lo que utiliza la misma función *hash* que en el paso 12 para generar el RPT utilizando su propio número de serie y el "nonce" recibido.

17. Envío de mensaje DISCONNECT: Una vez realizados todos los pasos, desde el dispositivo se cierra la conexión MQTT enviando un mensaje DISCONNECT al servidor de recursos broker. Todas las comunicaciones que se hagan desde el dispositivo en adelante se acompañarán del RPT validado.

3.6. Implementaciones, pruebas y resultados

Los dos mecanismos propuestos en las secciones 3.4 y 3.5 han sido implementados para poder validar y probar su viabilidad tanto de forma independiente como formando parte de la plataforma de Smart Toys. En las siguientes secciones se describen las implementaciones, pruebas y resultados llevados a cabo en cada uno de los casos.

3.6.1. Implementación de los mecanismos de autenticación y cifrado

Para poder probar los mecanismos de autenticación y cifrado basados en clave simétrica, se ha llevado a cabo una implementación sobre los prototipos definidos en el capítulo anterior de este trabajo. Para evitar posibles fallos de seguridad en las implementaciones, se han utilizado, siempre que ha sido posible, librerías estándar que ya implementan los algoritmos utilizados. Así, se han utilizado las librerías PyCrypto [185] de Python para la seguridad implementada en los gateways y la librería Crypto [186] de Arduino para la parte de seguridad que corresponde a los Smart Toys.

Una de las principales preocupaciones al utilizar un sistema de cifrado en un sistema como este donde los recursos son limitados, es el tiempo y recursos de procesado que hay que añadir al sistema debido al cifrado. En este caso el objetivo es determinar si esta sobrecarga temporal podría interferir con el funcionamiento normal del dispositivo.

Primero se ha medido el tiempo medio consumido por los mecanismos de cifrado y autenticación propuestos, mediante la ejecución repetitiva del programa que los implementa en Arduino. El resultado obtenido es que, por cada mensaje enviado, se utilizan $9687\mu s$ para incluir la autenticación y el cifrado. Si se comparara este tiempo con el tiempo necesario para enviar un mensaje sin cifrar (aproximadamente $800\mu s$ según las mediciones), se tiene que la sobrecarga derivada del cifrado es muy alta.

Sin embargo, hay que tener en cuenta que los Smart Toys no envían habitualmente mensajes de forma continua. Así, en los experimentos realizados en las pruebas piloto, que utilizan el preprocesado de los datos de los sensores para enviar un mensaje por movimiento, se ha podido medir una media de $3,56$ segundos entre cada movimiento, y por tanto entre cada mensaje. Así, si se tiene en cuenta el tiempo entre mensajes, el tiempo dedicado a los procesos de seguridad alcanza únicamente un $0,29\,\%$ aproximadamente del tiempo total. Además, los mensajes enviados normalmente no se realizan de forma consecutiva por parte del mismo Smart Cube, por lo que el tiempo medio entre dos mensajes se podría considerar aún mayor.

Si suponemos, por el contrario, que los datos enviados por parte de los Smart Toys se hacen de forma continuada para obtener los datos tal y como salen de los sensores (se verá más acerca de este modo de funcionamiento en el capítulo 4 de este mismo

libro), se puede comprobar que el tiempo consumido en el cifrado sigue permitiendo el correcto funcionamiento del sistema. Suponiendo una frecuencia de transmisión de $50Hz$ (que es un valor suficiente para obtener valores válidos de los sensores) se tendría un mensaje cada $0,2s$, o lo que es lo mismo, cada $200000\mu s$, por lo que el cifrado de los mensajes podría seguir siendo una posibilidad.

Por otro lado, además de determinar la viabilidad del sistema en cuanto a los recursos necesarios para su implementación, se han hecho los cálculos necesarios para evaluar si el sistema es suficientemente seguro ante posibles ataques. Así, en el caso de la autenticación, si por ejemplo un atacante pudiera potencialmente capturar un mensaje de inicio de actividad para intentar enviar otros mensajes suplantando al emisor, tendría que probar 2^{128} posibles combinaciones que al final resultarían $2^{128}/2^{16}$ posibles claves reales. Además, podría comparar los resultados con cada una de las 5 "pipes" de comunicación para reducir la cantidad de posibles claves a probar, pero, aun así, el ataque es muy difícil por fuerza bruta. Hay que tener en cuenta, además, que una actividad dura solo unos pocos minutos, por lo que el ataque debería estar limitado a ese tiempo, ya que la clave se cambia en cada actividad.

Además, el sistema propuesto está fuertemente basado en mecanismos conocidos de seguridad, tales como TLS, AES, o HMAC, lo cual ya ofrece un gran nivel de seguridad en el sistema. Por ejemplo, el algoritmo AES de 256 bits utilizado para el cifrado de los mensajes de los Smart Toys garantiza hasta 2^{256} combinaciones diferentes para la clave de sesión, lo cual lo hace computacionalmente seguro según los estándares actuales. Especialmente en este sistema donde cada clave sólo se usa para cifrar un número muy pequeño de mensajes y durante un período de tiempo también muy reducido. Además, cada "pipe" utiliza una clave distinta lo que hace que en una misma actividad se utilice aún menos veces.

3.6.2. Implementación y pruebas del sistema de control de acceso

Al igual que el sistema propuesto para el cifrado y la autenticación de los mensajes, el sistema de control de acceso ha sido objeto de una implementación y pruebas asociadas que han permitido validarlo. En esta sección se define cómo se ha llevado a cabo la implementación, las pruebas realizadas y los resultados obtenidos.

3.6.2.1. Implementación del sistema

Para poder obtener un prototipo funcional del sistema, se han implementado y/o desplegado cada una de las entidades del sistema propuesto. Para ello, se han utilizado servicios Web alojados en la Nube (Cloud Amazon Web Services (AWS)), prototipos de Smart Toy y gateway (basados arduino y placas nodeMCU v1.0 [187] respectivamente).

La implementación de cada entidad se ha llevado a cabo de la siguiente manera:

- Servidor de recursos broker: Este servidor se ha desarrollado mediante la utilización de dos servidores conectados internamente entre sí. Las funcionalidades MQTT se han basado en el broker Mosquitto [188], ya que permite la generación y añadido de plug-ins de autorización personalizados. Para añadir la implementación necesaria en este caso, hemos utilizado un plug-in llamado mosquitto_pyauth [189], que permite definir estos esquemas de autorización en base a scripts escritos en el lenguaje Python. Por otro lado, la API HTTP necesaria para conectar el servidor de recursos con el cliente utilizado para su gestión, el módulo de protección usado para conectar el servidor de recursos con el servidor de autoriación, y la gestión de los "topics" se han implementado en un único servidor HTTP.

- Servidor de Autorización: Para este servidor, se ha programado de cero un servidor HTTP escrito en Python que implementa la especificación de UMA, incluyendo las APIs de protección y autorización, y una interfaz de usuario para el control y definición de políticas de acceso. Además, incorpora OAuth para la generación de tokens de acceso a las APIs.

- Cliente: Se utiliza una aplicación Web simple desarrollada para el prototipo, que permite la configuración de los Smart Toys y el gateway y lleva a cabo las comunicaciones con el servidor de recursos a través de la API HTTP de este último.

- Dispositivos IoT: Se utiliza un grupo de Smart Toys conectados directamente a un gateway que sirve de punto de acceso al sistema de control de acceso, por lo que los Smart Toys sólo se comunican con él, y a efectos de este esquema, forman

un único dispositivo. El gateway está desplegado en una placa NodeMCU v1.0 (basada en ESP8266), y utiliza una librería que implementa un cliente MQTT disponible para dispositivos compatibles con Arduino.

- Dispositivo de medición: Se añade al resto de entidades un dispositivo extra que permite la medición del consumo energético derivado de la utilización del esquema de control de acceso. Para ello se utiliza una placa Adafruit INA219 [190] de medición de corriente continua, así como una placa Arduino Uno R3 con un Arduino Ethernet Shield incorporado para recoger los datos de la medición.

Cada una de estas entidades se ha desplegado además en las siguientes localizaciones:

- Tanto el servidor de recursos broker como el servidor de autorización se han desplegado en un entorno Cloud para simular los servidores localizados en Internet que corresponden a la plataforma de Smart Toys. Específicamente, se utilizan dos instancias EC2 t2.micro de AWS (que cuentan con 1 vCPU a 2,5 GHz de la familia Intel Xeon, y 1GB de memoria). Estas máquinas se han desplegado en el área EU-West y se han configurado en una subred común. Cada una de las máquinas ejecuta Ubuntu Server 14.04 como sistema operativo.
- Cliente: El cliente se ha desplegado en un servidor local con acceso a Internet.
- Dispositivos: Los Smart Toys se comunican con el gateway a través de un intefaz de radiofrecuencia y éste a su vez cuenta con una interfaz Wi-Fi y acceso a Internet, todo ello en un entorno local.
- Dispositvo de medición: Recoge y almacena los datos de consumo medidos y los devuelve a un servidor local al que está conectado.

3.6.2.2. Configuración del sistema para la medición del consumo energético

Las pruebas de consumo energético se han clasificado en función de su objetivo: Se ha llevado primero a cabo una serie de tests para validar la elección de un sistema

basado en el intercambio de mensajes sobre modelos de comunicación clásicos de Internet, y la segunda evalúa la sobrecarga energética producida por el uso del sistema de control de acceso sobre uno de estos protocolos.

- Las pruebas de validación del sistema de comunicaciones consisten en:

    1. Comparación de dos sistemas de comunicación populares en servicios de Intenet: API REST y un sistema basado en suscripción y publicación (MQTT).
    2. Configuración del gateway como un servidor web capaz de servir los datos de Smart Toys a partir de una API REST, y otro gateway idéntico para servir los datos en un "topic" MQTT.
    3. Definición de un conjunto común de intercambio de mensajes para permitir medir el consumo energético en cada caso.
    4. Inicio de la medición cuando se intercambia el primer mensaje y fin cuando todos los mensajes se han enviado de forma satisfactoria.

- Las pruebas de sobrecarga en cuanto a la energía por su parte son:

    1. Configruación de dos gateways como clientes MQTT. Cada dispositivo tendrá las mismas características que el otro.
    2. Se despliega un broker estándar MQTT (utilizando el software Mosquitto), sin incluir el esquema de control de acceso.
    3. Se define un conjunto de mensajes para permitir la medición de la energía en cada dispositivo.
    4. Se inician las mediciones con el envío del primer mensaje y se finalizan cuando todos los mensajes se han enviado de forma satisfactoria.

3.6.2.3. Resultados de la implementación

Los resultados de las pruebas definidas en la sección anterior se describen en este apartado. Se ha establecido una configuración para las todas las pruebas realizadas con el sistema:

- Todos los dispositivos se encuentran al inicio encendidos y en un estado estable.
- El gateway cuenta al inicio con un token RPT validado, por lo que no es necesario utilizar el flujo de validación visto en la sección 3.5.2.3.
- El gateway, funcionando como cliente MQTT se ha suscrito al "topic" usado para el intercambio de mensajes en las pruebas.
- El servidor de recursos broker ha validado esa suscripción al verificar los permisos.

El test consiste en completar un ciclo completo de publicación y recepción (con una suscripción previa) de un mensaje MQTT. El dispositivo publica un mensaje en el "topic" de prueba y espera a la recepción. Se han seleccionado dos puntos de medida:

- En el gateway se inicia la medición cuando realiza la acción de publicación y se para cuando se confirma la recepción del mensaje.
- En el servidor de recursos broker se realiza la medición dentro del propio módulo de autorización PyAuth, iniciándose cuando se envía la solicitud de validación de permisos al servidor de autorización (es decir, en el flujo de introspección de OAuth), y finaliza cuando se recibe la respuesta.

Para obtener una medición de referencia en cada uno de los dos escenarios, se ha ejecutado el mismo ciclo de mensajes en cada ocasión, pero deshabilitando el módulo de control de acceso. De esta manera, se ha podido determinar el tiempo necesario para procesar cada mensaje MQTT por parte del broker.

Las medidas obtenidas en cuanto a tiempo en el gateway se pueden expresar según la ecuación 3.14 en el escenario sin control de acceso, y según la ecuación 3.15 en el escenario con el control de acceso activado.

$$T_{T\_NoAuth} = T_{P\_IoTD} + T_{Tx\_Net} + T_{P\_RSB} \tag{3.14}$$

$$T_{T\_Auth} = T_{P\_IoTD} + T_{Tx\_Net} + T_{P\_RSB} + T_{P\_PyAuth} + 2 * T_{P\_Intros} \tag{3.15}$$

La ecuación 3.14 representa el tiempo del ciclo completo de prueba cuando el módulo de control de acceso no está activo. Consiste en la suma del tiempo de procesado en el dispositivo (T_{P_IoTD}), el tiempo de transmisión del mensaje a través de la red (T_{Tx_Net}) y el tiempo de procesado en el servidor de recursos broker (T_{P_RSB}).

De forma similar, en la ecuación 3.15, se puede ver el tiempo medido cuando el módulo de control dea cceso está activo. En este caso, se añaden el tiempo utilizado por el módulo de autorización (T_{P_PyAuth}) y el tiempo necesario para obtener la respuesta de validación por parte del servidor de autorización (T_{P_Intros}). Este último tiempo incluye tanto el tiempo de procesado en el servidor de autorización durante la validación como el tiempo de transmisión de la respuesta. Hay que tener en cuenta que se realizan dos peticiones de introspección en cada ciclo: una para la acción de publicación y otra para la acción de suscripción, por eso este último valor aparece multiplicado por dos.

Aplicando las dos expresiones anteriores, es posible obtener el tiempo adicional necesario debido al módulo de autorización (T_{P_PyAuth}), tal y como se puede ver en las ecuaciones 3.16 y 3.17.

$$T_{T\_Auth} = T_{T\_NoAuth} + T_{P\_PyAuth} + 2 * T_{P\_Intros} \tag{3.16}$$

$$T_{P\_PyAuth} = T_{T\_Auth} - (T_{T\_NoAuth} + 2 * T_{P\_Intros}) \tag{3.17}$$

El mismo ciclo de prueba se ha repetido 1000 veces, para poder obtener un valor medio representativo de cada medición. En la Tabla 3.2 se pueden ver los resultados medios obtenidos, junto con el cálculo de la desviación estándar en cada caso.

Tabla 3.2: Resultados de las mediciones de retardos medios (en milisegundos)

Variable	**Media**	**Desviación estándar**
T_{T_NoAuth}	43.83	5.99
T_{T_Auth}	67.74	11.07
T_{P_Intros}	11.84	1.28
T_{P_PyAuth}	0.23	-

Finalmente, se han llevado a cabo las pruebas definidas para el cálculo del consumo energético. Durante estos experimentos, se ha monitorizado la placa que hacía las veces de dispositivo (tanto en valores de voltaje como en valores de corriente), a partir del sensor INA219.

Los resultados medidos durante estas pruebas se pueden ver en la Tabla 3.3. En esta tabla se muestran los valores medios de corriente medidos por el sensor en cada prueba. En la Tabla 3.4 por su parte, se muestran los consumos con cada configuración.

Tabla 3.3: Mediciones de corriente durante los experimentos (en miliamperios)

Estado del dispositivo	**Wi-fi**	**Media**	**Desv. estándar**
En reposo	✗	33.47	1.67
En reposo	✓	81.08	4.21
Cliente HTTP con peticiones	✓	96.37	7.38
Cliente MQTT con peticiones	✓	88.48	2.40
Cliente MQTT con peticiones y autorización	✓	88.51	2.27

En la Tabla 3.3, se puede observar el valor medio y la desviación estándar del consumo de corriente eléctrica por parte del dispositivo que funciona como gateway, cuando se realiza un envío y recepción de 1000 mensajes con las configuraciones:

- Estado de reposo sin Wi-Fi activa: En este caso se deshabilita el interfaz Wi-Fi de la placa NodeMCU, que además se mantiene en un estado de reposo (es decir, sin enviar ni recibir mensajes) durante el tiempo medio que se tardaría en recibir 1000 mensajes en el resto de experimentos.
- Estado de reposo con Wi-Fi activa: El módulo que proporciona la conectividad Wi-Fi produce un gasto de energía importante, por lo que se ha decidido hacer por separado ambas mediciones.
- Cliente HTTP con peticiones: En este caso se habilita la interfaz Wi-Fi y se configura para realizar una transmisión y recepción de 1000 mensajes a través de una API REST.
- Cliente MQTT con peticiones: Se habilita la interfaz Wi-FI y se configura para realizar una transmisión y recepción de 1000 mensajes a través del protocolo MQTT.

- Cliente MQTT con peticiones y autorización: Como en el caso anterior, se habilita la interfaz Wi-Fi y se realizan, a través de MQTT la serie de 1000 peticiones y respuestas a mensajes, pero en este caso habilitando el módulo de autorización del servidor de recursos broker, y por tanto utilizando el esquema de control de acceso.

Además de mostrar los valores de corriente que se pueden ver en la Tabla 3.3, hemos calculado el valor medio de consumo teniendo en cuenta el voltaje usado para alimentar el dispositivo. Este valor medio es de 5,08V con una desviación estándar de 0,07.

A partir de los valores resultantes, se puede obtener el consumo energético para la transmisión de cada mensaje aplicando la ecuación 3.18.

$$E_c = V_{cc} * I_c * t_m[J] \tag{3.18}$$

Donde V_{cc} es el voltaje medido en Voltios, I_c es la corriente en Amperios, y t_m es el tiempo necesario para la transmisión del mensaje (en segundos).

En la Tabla 3.4 se muestra el consumo de energía necesario para la transmisión de un único mensaje en cada una de las configuraciones utilizadas en las pruebas donde se ha producido intercambio de mensajes (no se tienen en cuenta las mediciones realizadas en reposo).

Tabla 3.4: Resultados de la medición de consumo energético (milijulios)

Prueba	**Media**	**Desviación estándar**
Petición REST	33.16	5.71
Petición MQTT	19.70	1.01
Petición MQTT con autorización	30.46	1.75

3.7. Resumen y consideraciones finales

En este capítulo se han determinado las características de seguridad que son necesarias para asegurar las comunicaciones de la plataforma de Smart Toys diseñada. Estas características se han obtenido en base a un estudio de las posibles amenazas que pueden surgir en este sistema, y en base a un estudio del estado del arte de las tecnologías y mecanismos disponibles para hacer frente a estos peligros. Las principales preocupaciones en cuanto a la seguridad en estos sistemas son las relacionadas con la privacidad y la confidencialidad de los datos, por lo que se han diseñado dos sistemas específicos para ofrecer un sistema de comunicaciones seguro dentro de la plataforma.

Para la confidencialidad y la autenticación de los mensajes intercambiados entre Smart Toys y gateways se ha utilizado un sistema basado en mecanismos estándar de cifrado y autenticación, pero que tiene en cuenta las características específicas de las comunicaciones mediante radiofrecuencia de los dispositivos. Para ello, se ha propuesto el método de "slicing MAC" que permite dividir el código en varias partes dependiendo del canal concreto de comunicación utilizado en cada caso. Esto permite obtener un nivel de seguridad suficiente minimizando el impacto en cuanto a procesamiento y tiempo en los dispositivos.

Por otro lado, se ha propuesto unificar el mecanismo de control de acceso en toda la plataforma, facilitando así la aplicación de políticas comunes de control de acceso independientemente de si deben aplicarse a las APIs de los servicios alojados en Internet o a los datos intercambiados desde los Smart Toys o sus gateways correspondientes. Para ello, se ha propuesto extender UMA, un perfil de OAuth 2.0 a este tipo de plataformas IoT, modelando las comunicaciones de los protocolos de publicación/suscripción como MQTT o AMQP. como recursos en el sentido de los servicios Web. De esta forma, es posible protegerlos tal y como se hace con estos servicios a través del mecanismo propuesto por UMA.

Ambas propuestas se han implementado y probado para determinar el tiempo y el consumo energético extra que requieren en los dispositivos, habiéndose validado en ambos casos su viabilidad, ya que el consumo y el tiempo necesario para la realización de estas tareas no impide el funcionamiento normal de los dispositivos. Mediante

estas propuestas, y otros mecanismos estandarizados de seguridad, se ha dotado a la plataforma de Smart Toys de seguridad para evitar las posibles amenazas que se han identificado en ella.

Capítulo 4

Detección y clasificación automática de movimientos con Smart Toys

En este capítulo se muestra el diseño y desarrollo de una metodología para la detección y clasificación automática de los movimientos realizados durante las actividades de juego con los Smart Toys diseñados. El capítulo se inicia con una introducción (sección 4.1) y luego presenta el estudio realizado sobre el estado del arte y trabajos relacionados en cuanto a la detección y clasificación de actividades mediante sensors (sección 4.2). A continuación se lleva a cabo una descripción del sistema propuesto para la detección y clasificación de los movimientos dividida en tres partes en la sección 4.3. En las secciones 4.4 y 4.5 se muestran los experimentos que se han llevado a cabo y sus resultados respectivamente, y se incluye un resumen final en la sección 4.6.

4.1. Introducción

En los capítulos 2 y 3 se ha proporcionado una base tecnológica a partir de la cual construir una plataforma de Smart Toys fiable, segura y funcional. El uso de esta plataforma permite a los expertos en desarrollo infantil realizar experimentos y acti-

vidades, así como obtener y almacenar los datos a partir de actividades concretas. Sin embargo, estos datos son generados por sensores que, por si mismos no dan suficiente información a los expertos para evaluar la actividad.

En el capítulo introductorio de este trabajo (capítulo 1), se ha mostrado que las actividades de juego son un mecanismo muy interesante para la evaluación, por parte de expertos, de su desarrollo psicomotor. Dadas las limitaciones en cuanto al desarrollo tecnológico de las herramientas actuales y la formación de los expertos, esta evaluación se realiza de forma eminentemente manual. Los datos se obtienen de la inspección visual de los niños realizando estas actividades.

En este capítulo, el trabajo de investigación se ha centrado en analizar las técnicas existentes de clasificación automática de movimientos para transformar los datos medidos en información útil para los expertos. Para ello, siguiendo el tercer objetivo general planteado en el capítulo 1, se van a proponer métodos y algoritmos que, a partir de las actividades desarrolladas con los dispositivos diseñados en este trabajo, permitan obtener de forma automática los movimientos que las componen, teniendo en cuenta las características específicas de estas actividades. Además, se automatizará el proceso de clasificación de esos movimientos, de forma que se pueda obtener información útil y más concreta con la que los expertos en desarrollo puedan trabajar sin necesidad de llevar a cabo una evaluación visual. Este sistema, además, puede colaborar en la generación de nuevas técnicas de evaluación basadas en la nueva información proporcionada.

La aparición en los últimos tiempos de numerosas técnicas basadas en la Inteligencia Artificial ha permitido el desarrollo de un gran número de sistemas autónomos que tomen decisiones de forma independiente. En el caso de la detección y clasfificación de movimientos, se han utilizado ampliamente técnicas de Machine Learning (aprendizaje automático), un subcampo de la Inteligencia Artificial. Estas técnicas han ofrecido muy buenos resultados aplicándose a la detección de múltiples actividades [191]. Se basan en el establecimiento de una serie de atributos comunes a cada clase de elementos a clasificar, y, a partir de un algoritmo, detectar si un valor de entrada corresponde a una u otra. En la sección 4.2, se ofrece una visión de los métodos y técnicas más habituales utilizadas actualmente. Esto permite determinar las características de estos sistemas y los desafíos específicos que supone aplicarlos a la plataforma de Smart Toys.

Las actividades de juego habituales en un entorno como el que nos ocupa suelen tener una duración corta (alrededor de pocos minutos) y los movimientos que las componen son reducidos, muy concretos y poco repetitivos. Estas características diferencian en gran medida los movimientos de estas actividades de los movimientos realizados en las actividades que suelen ser objeto de clasificación en la literatura. Como ejemplo de actividad sobre la que realizar la detección de movimientos, usaremos la construcción de una torre de cubos, ya utilizada en las pruebas piloto del sistema (véase la sección 2.4.4). Esta actividad se compone de una serie de movimientos (agarre de un cubo, movimientos verticales u horizontales y apilamiento de los cubos) que se realizan durante la actividad (habitualmente una única vez por cada cambio de posición de un cubo) y cuya duración no suele sobrepasar los 2 o 3 segundos.

Para cubrir las carencias de los métodos habituales aplicados a este tipo de movimientos, se ha diseñado una novedosa metodología para poder detectar y clasificar movimientos no repetitivos, de corta duración y con características espectrales que dificultan su clasificación. Este método se basa en el análisis de las señales proporcionadas por los sensores incluidos en los Smart Cubes. A partir de un conjunto amplio y variado de señales de referencia, se consigue una secuencia de tendencias en la aceleración (esto es, una estructura de datos que indique la secuencia de aceleraciones positivas, negativas o neutras que se producen a lo largo de la señal) que permita caracterizar un movimiento concreto a partir de una sucesión ordenada de aceleraciones (es decir, la obtención de un patrón para el movimiento).

Este subsistema de obtención de patrones de movimiento, una vez caracterizada la señal utilizando el método propuesto de análisis de tendencias de la señal, realizará un proceso de optimización basado en un algoritmo genético. Al finalizar este proceso de optimización se proporcionará el patrón de movimiento y el conjunto de variables del sistema más adecuado para un movimiento concreto.

Esta metodología propuesta se utilizará de igual forma para el análisis de una señal continua proporcionada por los Smart Cubes. El sistema aplicará las variables obtenidas para cada movimiento en el proceso de optimización de patrones y comparará los patrones temporales detectados con los definidos en cada caso, con el fin de detectar y clasificar cada movimiento.

Para validar la solución propuesta, se ha llevado a cabo su implementación, obteniendo un sistema completamente funcional. Por otro lado, se ha generado un conjunto de datos representativo de los movimientos para ser utilizado en el sistema propuesto y en otros dos métodos de detección y clasificación: Uno basado en la búsqueda de señales similares utilizando distancias Euclídeas, y otro basado en un modelo de aprendizaje supervisado (concretamente, una máquina de vectores de soporte). Esto permitirá no sólo validar de forma aislada el sistema propuesto, sino el compararlos frente a algunas de las soluciones existentes más populares.

Estos resultados muestran que el sistema propuesto presenta una alta precisión y grado de acierto, y además presenta algunas importantes ventajas: Primero, que el sistema es capaz de buscar patrones de movimientos óptimos a partir de un conjunto de señales de movimiento aisladas. Esto permite utilizar el sistema para analizar diferentes actividades de juego, algo fundamental en los entornos de Smart Toys. Además, al ser un método basado en el análisis muestra a muestra de las señales, puede utilizarse en entornos que requieran una respuesta en tiempo real.

4.2. Estado del arte y trabajos relacionados

Existe un gran número de estudios que abordan la tarea de la detección y clasificación automática de actividades realizadas por seres humanos. De hecho, el auge de los algoritmos de aprendizaje supervisado y no supervisado, junto con la aparición de multitud de dispositivos provistos de sensores y la aparición de paradigmas como el del Internet of Things [192–194], han propiciado que se hayan realizado multitud de trabajos con distintos enfoques y para la clasificación de actividades muy diversas.

El objetivo final de los métodos de reconocimiento de actividades es muy diverso, ya que este tipo de sistemas han probado su importante utilidad en campos tales como la salud [195–197], la monitorización de actividades físicas o deportivas [197–199], la generación de sistemas de vigilancia o de seguridad basados en mediciones biométricas [200] o la generación de interfaces de interacción entre humanos y máquinas [201, 202], entre otras.

En las siguientes secciones se presenta un estudio de las principales actividades detectadas por los sistemas automáticos, de los dispositivos utilizados y su colocación, y finalmente se clasifican los métodos más habituales utilizados en la detección y clasificación de actividades.

4.2.1. Tipos de actividades detectadas

En cuanto al tipo de actividades que se suelen clasificar en estos trabajos, existe también una alta heterogeneidad. En [191], se presenta una clasificación de algunas de las actividades más comunes. En este trabajo, las actividades se clasifican en:

- Actividades de movimiento o ambulatorias: Caminar, correr, sentarse, etc. [203, 204].
- Actividades de la vida diaria: Comer, beber, cocinar, etc. [205].
- Actividades relacionadas con el transporte: Conducir, montar en bicicleta, etc. [206].
- Actividades militares [207].
- Actividades relacionadas con el ejercicio: Remar, levantar pesos, etc. [208].
- Actividades que solo implican movimientos de la parte superior del cuerpo: Hablar, mover los ojos, la cabeza o la mano [209].

En [210] hay una clasificación similar. Aunque existen sistemas dedicados a la mayoría de los ejemplos citados anteriormente y muchos otros, lo cierto es que la mayoría de los trabajos se centran en las actividades del primer grupo, es decir, aquellas relacionadas con el movimiento como caminar, correr, etc.

4.2.2. Dispositivos y colocación de sensores

En esta sección se clasifican diversos estudios relacionados con estos sistemas dependiendo de cómo se disponen los sensores utilizados para obtener la información

que permita la detección de las actividades, y en función de los sistemas y modelos utilizados para su clasificación.

En [211], se puede encontrar un estudio bastante extenso sobre trabajos relacionados con la detección de actividades que adquieren los datos a partir de dispositivos "Wearables", es decir, de dispositivos con sensores ubicados en distintas partes del cuerpo humano. Así, se tienen dispositivos que se localizan en la cintura [212–214], en la muñeca [215, 216], en la parte baja de la espalda [199] o, en muchos de los trabajos, en una combinación de las anteriores disposiciones y otros lugares del cuerpo tales como la cadera, el tronco, el pecho, etc. [217–221]. En trabajos más recientes como [222] o [223], se utilizan datos provenientes de Smart Phones sujetados o llevados por parte de las personas que realizan las actividades, en lugar de situar sensores fijos en un lugar determinado del cuerpo. Evidentemente, la posición de los sensores determinará el tipo de actividad concreta a detectar, ya que los datos obtenidos en cada caso indicarán con mayor precisión el movimiento de cierta parte del cuerpo. En cuanto a los tipos de sensores utilizados, la mayoría de estos trabajos utilizan acelerómetros únicamente o, en muchos casos, en conjunción con otros sensores como por ejemplo sensores de electromiografía (EMG).

4.2.3. Técnicas de detección y clasificación

En general, estos trabajos tienen una cosa en común además de la utilización de sensores para la obtención de datos, y es la utilización de técnicas de Machine Learning para la clasificación de los movimientos realizados en las actividades. Estas técnicas consisten habitualmente en la extracción o cálculo de una serie de características a partir de los datos de los sensores en una cantidad suficiente para alimentar un modelo de aprendizaje supervisado o no supervisado (es decir, marcando los datos de entrenamiento como pertenecientes a una de las clases a clasificar o no. La precisión del modelo entrenado se valida mediante el uso de un subconjunto de los datos de entrenamiento (datos de validación), o bien mediante un sistema de validación cruzada, el subconjunto de validación se va rotando mediante un número k de combinaciones de subconjuntos sobre el total de datos.

Una cuestión importante en estos sistemas es el conjunto de atributos a extraer

o calcular a partir de los datos de los sensores en bruto. Habitualmente y dado que se obtienen a partir de muestras de señales, se suelen clasificar en atributos basados en el dominio del tiempo y atributos basados en el dominio de la frecuencia [211]. Algunos de los atributos más utilizados en el dominio del tiempo son, por ejemplo, el valor medio de las muestras, la varianza, el valor de asimetría (*skewness*), el valor de curtosis, etc. [224]. Otros posibles valores a calcular, específicos para acelerómetros, son la suma de la integración del módulo de la aceleración en cada eje [225], o el valor cuadrático medio (*Root Mean Value*, RMS) [226]. En cuanto al dominio de la frecuencia, se suelen utilizar medidas derivadas de la transformada discreta de Fourier (*Discrete Fourier Transform*, DFT) de la señal, la densidad espectral de potencia (*Power Spectral Density*, PSD) o la entropía [211, 227].

La selección de un método de clasificación u otro depende tanto de los datos utilizados como de la selección de atributos y del resultado esperado. Como ya se ha adelantado, habitualmente los métodos de clasificación se dividen en métodos de aprendizaje supervisado y métodos de aprendizaje no supervisado [228]. Habitualmente se utilizan los métodos de aprendizaje supervisado cuando el conjunto de datos de entrenamiento no se puede clasificar a priori como parte de una clase concreta. Otra posible clasificación de métodos sería entre aquellos que no consideran los datos históricos (es decir, se basan solo en una única toma de los datos) y los métodos secuenciales que sí tienen en consideración la historia de los datos.

Finalmente, es posible clasificar estos métodos dependiendo del tipo de técnica utilizada en ellos para la clasificación de los elementos. En este sentido, se pueden considerar las siguientes clases:

- Métodos probabilísticos: Son métodos basados en modelos de probabilidad que son capaces de determinar si un elemento está en una clase u otra con cierta probabilidad. Ejemplos de estos métodos son los clasificadores Bayesianos [229], los clasificadores basados en modelos mixtos Gaussianos (*Gaussian Mixture Models*, GMM) [230], o modelos de Markov ocultos (*Hidden Markov Models*, HMM) [231].
- Métodos geométricos: Son aquellos métodos que se basan en la división del espacio en regiones dependiendo de los atributos de cada clase, y clasifican los datos de entrada de acuerdo a la región en la que se deban incluir. Ejemplos de métodos

basados en este sistema son las redes neuronales artificiales (*Artificial Neural Networks*, ANN) [232–234], métodos basados en el algoritmo de los k vecinos más próximos (*k-Nearest Neighbors*, kNN) [235] o los basados en máquinas de vectores de soporte (*Support Vector Machines (SVM)* [227]. Existen también una serie de métodos basados en umbrales que se pueden clasificar como métodos geométricos [236–238].

- Métodos basados en plantillas: Se basan en la comparación entre los datos de entrada y una serie de plantillas predefinidas que se pueden obtener de forma manual o a través de un método de entrenamiento [239].
- Métodos binarios: Se basan en la creación de un árbol de decisión donde cada nodo es capaz de discriminar entre dos posibles estados, hasta determinar la clase de forma completa [213, 240].

En general, en todos los trabajos referenciados para cada tipo de método de clasificación, se consigue una alta precisión en la detección de las actividades que se desean clasificar. Hay que tener en cuenta, por otro lado, que la gran mayoría de estos trabajos se basan en la clasificación de actividades de una duración relativamente larga y formadas por movimientos repetitivos a lo largo del tiempo, específicamente, se basan la mayoría en actividades ambulatorias según la clasificación incluida al inicio de esta sección (correr, caminar, etc.).

Sin embargo, la mayoría de estos métodos no son tan efectivos al ser aplicados a actividades de corta duración formadas por movimientos muy breves y no repetitivos, tales y como los movimientos que se producen durante las actividades de juego en nuestro caso de uso: movimientos de unos pocos segundos de duración y que se repiten únicamente unas pocas veces durante toda la actividad. Existen trabajos que sí se centran en la clasificación de actividades con similares características. Por ejemplo, en [202], se propone un sistema para la clasificación de los movimientos oculares basado en un electrooculograma (*Electrooculography*, EOG). Este tipo de sensor es capaz de medir las diferencias de potencial entre los nervios de la retina y la córnea. Se trata de un clasificador binario simple que analiza la forma de la señal producida por el EOG para clasificar hasta tres posibles movimientos. Otros ejemplos de trabajos centrados en movimientos cortos son [241] y [242], que proponen sistemas para la clasificación

de gestos realizados con la mano, utilizando una combinación de acelerómetros y sensores para electromiogramas (*Electromyography*, EMG). El primero de estos trabajos utiliza un método basado en modelos ocultos de Markov (HMM) y obtiene los datos de sensores colocados en las muñecas y antebrazos de los participantes en las pruebas. El segundo se basa en un clasificador basado en un análisis discriminante lineal (*Linear Discriminant Analyisis*, LDA) y utiliza una combinación de sensores colocados alrededor del antebrazo de los individuos participantes. Ambos sistemas ofrecen una alta precisión en la detección y clasificación de una serie de gestos predefinidos, pero presentan como desventaja la necesidad de la instalación compleja de varios sensores y electrodos en los cuerpos de los participantes.

En [243] presentan un sistema capaz de detectar también una serie de movimientos realizados con la mano, pero tiene la misma desventaja que los trabajos anteriores: Se basa en la utilización de un conjunto de sensores multimodales y utiliza una aproximación basada en máquinas de vectores de soporte. Otra propuesta para el reconocimiento de gestos realizados con la mano se puede encontrar en [244]. En este caso, se utiliza un algoritmo conocido como *Dynamic Time Warping algorithm* (DTW) junto con una serie de plantillas preseleccionadas. En [245], sin embargo, se utiliza un método basado en modelos ocultos de Markov (HMM) y también una serie de plantillas preseleccionadas de gestos manuales. Todos estos trabajos también obtienen altas precisiones en la detección y clasificación de los gestos, pero presentan un problema en cuanto a la necesidad de reconfigurar y reentrenar los modelos utilizados para la inclusión de nuevos gestos.

Además de las aproximaciones basadas en sensores, en algunos trabajos como [246] y [247] se presentan propuestas donde la detección de estos gestos se lleva a cabo utilizando métodos para el reconocimiento de patrones en imágenes. La principal desventaja de este tipo de técnicas es la necesidad de instalación de cámaras para captar dichas imágenes en el lugar donde se vaya a proceder a aplicar la detección, lo cual limita mucho el uso del sistema.

4.3. Sistema basado en la generación de patrones para la clasificación de actividades de corta duración

La mayoría de los trabajos relacionados con la detección automática de actividades se centran en actividades que podríamos considerar de larga duración y/o repetitivas, tal y como se ha mostrado en la sección 4.2. Ejemplos de este tipo de actividades son caminar, correr, comer o cocinar, por ejemplo.

Tras la revisión del estado del arte, una conclusión que se puede extraer del estudio de dicha sección es que para este tipo de actividades las técnicas basadas en Machine Learning son en general bastante efectivas. Sin embargo, como se verá más adelante en los resultados experimentales de esta contribución, no es el caso del tipo de movimientos que se intentan clasificar en este trabajo. En este caso, se trata de actividades de corta duración, de no más de unos pocos minutos, y en las que los movimientos realizados no son especialmente repetitivos (cada tipo de movimiento puede aparecer como máximo unas pocas veces en cada actividad).

Para este tipo de movimientos ha sido necesario idear una metodología que, a partir de la búsqueda y extracción de un patrón de tendencias de aceleración específico para cada movimiento (basado en las señales de aceleración obtenidas por los sensores) sea capaz de determinar y clasificar la aparición de cada movimiento dentro de la actividad. Dicho sistema se describe en esta sección, y se basa en la obtención de dichos patrones en base a un conjunto de datos específico medido para cada movimiento.

En la Figura 4.1 se pueden ver las principales entidades que componen el sistema. A la entrada del sistema se tiene una señal de la aceleración generada por los sensores integrados en un Smart Cube. Para facilitar la comprensión del método propuesto, nos centraremos en el uso del valor de aceleración correspondiente al eje Z (paralelo a la gravedad). Esta señal ha sido medida de forma secuencial durante una actividad, y probablemente esté compuesta de uno o varios movimientos. Esta señal se preprocesa mediante un módulo genérico que permite normalizar las señales y evitar errores debidos a interferencias, y se pasa por una serie de N módulos, uno por cada movimiento a detectar. Estos módulos, compuestos por las mismas fases de análisis, un conjunto de atributos y patrones de entrada específicos de cada movimiento, tratan de extraer

detectar cada movimiento para el que han sido configurados. Finalmente, se comparan las N detecciones y se procede a una clasificación que depende de la confianza que se tenga en la detección de un movimiento determinado. Esta decisión es la que se obtiene a la salida del sistema en forma de vector conteniendo los movimientos detectados y sus instantes temporales de inicio y fin.

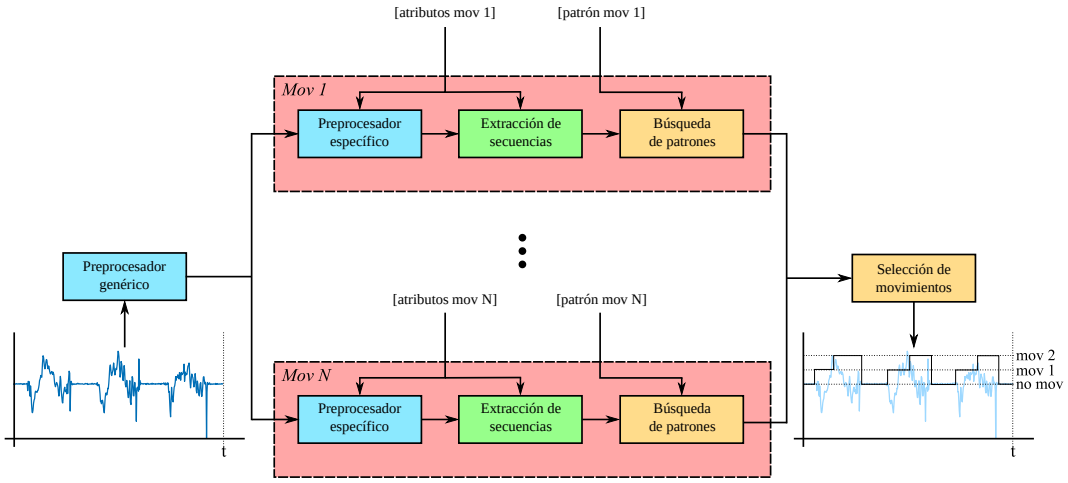

Figura 4.1: Diagrama indicando los principales módulos que componen el sistema propuesto para la detección y clasificación de movimientos.

En las siguientes secciones, se explicarán con detalle cada uno de los pasos y módulos del sistema, incluyendo el método para la optimización de las variables de entrada de cada módulo de detección de patrones de cada movimiento.

4.3.1. Análisis de señales de aceleración y generación de patrones

Para la obtención de los patrones de referencia de cada movimiento a analizar, es necesaria la utilización de datos de referencia que permitan obtener un patrón de aceleraciones común. Para esto, es necesario realizar un análisis de las señales de aceleración. El análisis espectral de la señal para la extracción de atributos a utilizar en los métodos habituales de clasificación no es útil en este caso debido a la corta duración de cada movimiento (de unos pocos segundos como máximo).

Sin embargo, es posible utilizar los datos del acelerómetro de 3 ejes incorporado en el juguete para componer un conjunto de muestras correspondientes a un vector de

aceleración de cada movimiento. Una actividad por tanto se puede descomponer en una secuencia temporal de movimientos ordenados, donde cada uno de esos movimientos será a su vez una secuencia temporal ordenada de aceleraciones. Estas aceleraciones indican la trayectoria del juguete durante el movimiento, por lo que pueden identificarlo dentro de la actividad.

Hay que tener en cuenta que, en este tipo de actividades de juego, puede existir una enorme diversidad de movimientos dependiendo de la actividad. Por tanto, el sistema debería ser capaz de incluir nuevos movimientos mediante la adquisición de estos conjuntos de datos que permitan extraer el patrón de referencia para cada nuevo movimiento, sin necesidad de modificar el resto del sistema y la detección de los movimientos ya incluidos en él.

Para obtener estos patrones en base a las señales de aceleración, proponemos estudiar las tendencias de crecimiento y decrecimiento de las aceleraciones en los vectores de aceleración. En las siguientes secciones, se definen los módulos del sistema de obtención de patrones: el procesador genérico, el preprocesador específico de cada movimiento, y el módulo de extracción de secuencias.

4.3.1.1. Preprocesador de señales genérico

Las señales utilizadas para su clasificación provienen de la utilización de Smart Toys diseñados en la plataforma. Estos dispositivos envían la información medida en sus sensores de forma inalámbrica a través de la plataforma. Una vez recopilados estos datos, es imprescindible realizar un tratamiento previo de las señales obtenidas para evitar que se utilicen en el sistema posibles datos erróneos o incompletos que pueden haber surgido durante la transmisión de la información debido a interferencias u otro tipo de errores.

El módulo preprocesador de señales genérico se encarga de llevar a cabo esta adaptación de las señales, para lo cual lleva a cabo una interpolación de las muestras de la señal utilizando la frecuencia original de muestreo del propio dispositivo. Esto hace que se consigan señales uniformes. Además, se realiza un análisis de cada eje para descartar posibles señales incompletas o erróneas, y se utilizan datos de calibración del Smart

Toy concreto donde se han producido las mediciones para eliminar posibles derivas en los sensores.

4.3.1.2. Preprocesador de señales específico

Además de la preparación de las señales llevada a cabo en el preprocesador genérico, es necesario realizar una serie de operaciones extra, cuya configuración dependerá del movimiento a detectar. Es por ello que se denomina preprocesador de señales específico a cada uno de los módulos funcionales que realizan estas operaciones para cada movimiento.

Este módulo lleva a cabo primero una operación de suavizado de la señal entrante. Este tipo de operación permite conformar las señales para hacerlas más similares, eliminando posibles mediciones erróneas que hayan escapado del anterior preprocesado, y rebajando los picos de señal más dispares. Para ello utiliza un filtro de Savitzky-Golay [248]. Este tipo de filtro utiliza dos parámetros variables para determinar el grado del suavizado, pero fijar estos valores no es una tarea trivial, ya que los valores óptimos dependen de cada movimiento a identificar. Por tanto, se ha diseñado el módulo de preprocesado para que estos valores se puedan modificar de forma dinámica en cada caso.

Posteriormente al proceso de suavizado, se lleva a cabo una nueva interpolación de la señal. Esta operación es interesante para la obtención posterior de las secuencias de tendencias de aceleración en el siguiente módulo, ya que estas tendencias dependerán de las diferencias entre muestras específicas. Nuevamente, este proceso de interpolación utilizará un valor de muestreo que será dependiente del tipo de movimiento a detectar, ya que movimientos que presenten una duración media mayor, pero con una magnitud acotada, se pueden representar con mayores períodos de muestreo (lo que equivale a una necesidad de almacenar menos muestras por movimiento), mientras que movimientos rápidos y con muchas variaciones pueden requerir tasas más grandes.

4.3.1.3. Extracción de secuencias

En este módulo se reciben las señales preprocesadas y, a través de un algoritmo de extracción, se determinan secuencias de tendencias de aceleración para cada señal de entrada. Las tendencias de aceleración indican si la aceleración crece o decrece en un instante determinado mediante un conjunto de etiquetas, como se verá más adelante. En el algoritmo se usan una serie de valores que se definen a continuación:

- Se definen los valores de aceleración de una señal determinada como $f(x)_{(1 \leqslant x \leqslant N)}$, donde N es el número total de muestras de la señal.
- A partir de los valores anteriores, se obtiene un nuevo conjunto de valores $d(x)_{(1 \leqslant x \leqslant N)}$ que indica las diferencias entre muestras consecutivas, y que viene dado por la ecuación 4.1:

$$d(x) = f(x) - f(x-1) \tag{4.1}$$

  Es decir, cada valor de $d(x)$ contiene la diferencia entre un valor determinado de aceleración y el valor de la muestra anterior.

- Definimos $t(x)_{(2 \leqslant x \leqslant N-1)}$ como el conjunto de tendencias de aceleración correspondientes a cada una de las muestras de $f(x)$. Este conjunto de datos contendrá valores de entre los incluidos en un conjunto de datos que determina las posibles etiquetas para las tendencias de aceleración. Aunque se podrían determinar más etiquetas en base por ejemplo a la inclinación de los valores de diferencia de aceleración, en nuestro caso hemos definido este conjunto tal y como se observa en la expresión 4.2:

$$accelValues = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} \tag{4.2}$$

  La ecuación indica que los valores del conjunto $t(x)$ pueden ser exclusivamente uno de los tres valores del conjunto anterior: 1 para aceleraciones positivas, -1 para indicar las aceleraciones negativas, y 0 para indicar la ausencia de aceleración.

Además de los conjuntos de datos anteriores, para la extracción de las secuencias será necesario determinar una serie de valores umbral que indiquen cómo etiquetar cada valor de aceleración en cada instante determinado. Estos umbrales deben especificarse en cada caso, y son por tanto variables dependiendo del movimiento concreto que se vaya a analizar.

- *zeroThreshold*: Es un valor que indica el límite a partir del cual se considera que la variación de aceleración entre dos muestras consecutivas se puede considerar aceleración o no. Es un umbral que se debe aplicar a aquellas muestras cercanas al valor medio de la señal.
- *meanThreshold*: Es un valor que se compara con el valor medio absoluto de un subconjunto de muestras consecutivas. Si éste es superior al valor umbral, consideraremos que todas las muestras seguirán la misma tendencia que la mayoría del subconjunto.
- *repeatThreshold*: Es un valor que se compara con la división entre el número de muestras que siguen la tendencia más popular dentro de un subconjunto de muestras consecutivas. Si el valor obtenido de la división es mayor que el valor del umbral, se considera que todas las muestras del subconjunto siguen la tendencia más popular en el subconjunto.

El algoritmo de extracción de secuencias sigue una serie de pasos que se explicarán a continuación. Cada iteración del algoritmo evalúa una muestra en orden creciente temporal (de $x = 1$ a $x = N$, siendo i el instante de la muestra analizada en cada iteración).

1. El algoritmo clasifica la tendencia de la muestra actual utilizando el valor de aceleración correspondiente ($f(x)$) y el valor de diferencia con el valor anterior ($d(x)$), a partir de los posibles valores del conjunto *accelValues* (en nuestro caso, como ya se ha indicado anteriormente, esto significa que los posibles valores son: 1 para aceleraciones positivas, -1 para aceleraciones negativas, y 0 cuando no hay aceleración. Para determinar qué valor se debe seleccionar se utiliza la expresión 4.3:

$$currentTrend = \begin{cases} 0, & \text{Si } |d(i)| < zeroThreshold \text{ y} \\ & |f(i)| < zeroThreshold/2 \\ 1, & \text{Si } d(i) > 0 \\ -1, & \text{Si } d(i) < 0 \end{cases} \tag{4.3}$$

2. Esta decisión sobre la tendencia de una muestra concreta puede ser considerada por el algoritmo como definitiva o condicional. En el segundo caso, el valor definitivo dependerá de los valores de las muestras inmediatamente anteriores o posteriores. Para ello, se consideran dos posibles estados del sistema: "estable" e "inestable". Consideramos el sistema "estable" cuando todos los valores anteriores al valor actual $t(i)$ se han marcado como definitivos. En este caso:

   - Si $currentTrend = t(i-1)$:
     - $t(i) = currentTrend$
     - El valor de $t(i)$ se marca como definitivo, y se mantiene el estado de estabilidad.
   - Si $currentTrend \neq t(i-1)$:
     - $t(i) = currentTrend$
     - El valor de $t(i)$ se marca como provisional, por lo que puede variar en las futuras iteraciones del algoritmo. Además, el sistema se pasa a estado "inestable". En este estado se utiliza un puntero p para indicar la posición i marcada como provisional.

   En estado "inestable", el algoritmo se comportará en este paso de la siguiente manera:

   - Se genera o utiliza un conjunto de muestras denominado *tempTrend*. En este conjunto se incluyen valores de las muestras entre $2p-i$ y i, donde p es el puntero a la muestra etiquetada provisionalmente, e i es la muestra analizándose en la iteración actual. Por ejemplo si nos encontramos analizando la siguiente muestra con respecto a la apuntada por p (es decir, $i = p+1$), el conjunto de datos incluido en *tempTrend* será el definido por la expresión 4.4:

$$tempTrend = [t(p-1), t(p), t(p+1)] \tag{4.4}$$

O puesto en función de la muestra actual i, se puede indicar como en la expresión 4.5:

$$tempTrend = [t(p-1), t(p), t(i)] \tag{4.5}$$

A continuación, se calcula la media absoluta de los valores de diferencia de aceleración para los datos incluidos en el conjunto anterior, tal y como se muestra en la ecuación 4.6:

$$tempMean = |\sum_{j=2p-i}^{i} \frac{d(j)}{2i-2p+1}| \tag{4.6}$$

A partir de estos valores se realizan las siguientes operaciones:

- Se comprueba si el valor medio está por encima o por debajo del umbral correspondiente, *meanThreshold*. Si $tempMean > meanThreshold$, se evalúa la tendencia global del conjunto de muestras siguiendo la expresión 4.3 del primer paso del algoritmo. Con los resultados obtenidos se actualizan los valores de $t(x)$ en las muestras correspondientes al conjunto de datos *tempTrend*, es decir, para todo x tal que $2p - i \leqslant x \leqslant i$.
- Se realiza la división entre el número de apariciones de la tendencia más popular en el conjunto de datos temporal y el tamaño de éste conjunto. El valor resultante se compara con el umbral correspondiente *repeatThreshold*. Si el valor es mayor que el del umbral, se actualizan los valores de $t(x)$ correspondientes al conjunto de datos temporal *tempTrend* (para todo x tal que $2p - i \leqslant x \leqslant i$) utilizando la tendencia más popular.
- En cualquiera de los dos casos anteriores, tras la actualización se elimina el puntero p y el sistema se vuelve a considerar en estado "estable", quedando las tendencias actualizadas como definitivas.
- Si ninguna de las dos comparaciones anteriores se cumple, suponemos que no es posible tomar una decisión definitiva sobre las muestras contenidas en el conjunto de muestras temporal, por lo que en esta iteración del algoritmo se mantiene el estado de "inestable" y se inicia la siguiente

iteración para la muestra $i + 1$.

- Se comprueba si el tamaño del conjunto *tempTrend* es mayor o menor de un tamaño máximo fijado al inicio (*bufferSize*). Si esto ocurre, se mueve el puntero p a la siguiente posición ($p = p+1$) y se establece que el primer valor de los contenidos en el conjunto temporal *tempTrend* ($t(2p - i)$) es definitivo.

3. Cuando se han analizado todas las muestras y se tiene un conjunto $t(x)$ completo con un valor por cada muestra original menos una ($N-1$), se procede a realizar un resumen del mismo, manteniendo un único valor por cada conjunto de tendencias similares e indicando para cada uno de estos conjuntos el instante temporal de inicio y de fin.

Al final de la ejecución de los pasos enumerados, se obtendrá, para la señal de entrada, una estructura de datos formada por las secuencias de etiquetas que la identifican. Esta secuencia se puede utilizar a continuación como entrada para el sistema de clasificación en base a las secuencias patrón de referencia.

4.3.2. Algoritmo para la optimización de las variables de entrada

En los módulos de preprocesado específico y de extracción de secuencias se ha podido observar que ciertos valores de entrada se han determinado de forma variable, y dependiente de los movimientos concretos a analizar en cada momento. La obtención de los valores óptimos para cada posible movimiento no es un problema trivial, ya que existen virtualmente infinitos posibles valores para cada una de las siete variables. Las variables son:

- *bufferSize*: Es el valor que indica el tamaño para el conjunto de muestras consecutivas que se pueden incluir en el conjunto temporal de muestras no definitivas *tempTrend*. Se utiliza en el algoritmo de extracción de secuencias para determinar si se puede ampliar dicho conjunto de datos o se debe mover el puntero de datos temporales (paso 3).

- *order* y *framelen*: Son los valores que utiliza el filtro de Savitzky-Golay para el suavizado de la señal en su configuración en el módulo de preprocesado específico. El primer valor indica el orden polinomial sobre el que se lleva a cabo la función de filtrado, mientras que el segundo valor indica el tamaño de la trama de entrada para el filtro. El valor de *order* debe ser siempre menor que el valor de *framelen* y además este último debe ser siempre un valor impar.
- *optSampleRate*: Este valor se utiliza durante el preprocesado específico y es el valor variable de interpolación utilizado durante este proceso, que tiene como objetivo la simplificación de las señales de entrada en los movimientos en los que sea posible.
- *zeroThreshold*, *meanThreshold* y *repeatThreshold*: Son los valores umbral definidos en la sección 4.3.1.3 y utilizados en el algoritmo de extracción de secuencias para determinar las tendencias de aceleración en el paso 2.

La diversidad de posibles valores que pueden tomar estas siete variables nos ha llevado a considerar la obtención de los mejores posibles valores para cada movimiento como un problema de optimización. Para resolver este problema, hemos planteado la utilización de un Algoritmo Genético (GA), un algoritmo evolutivo basado en la definición de una población de posibles soluciones al problema que se va variando en cada iteración a partir de unas funciones de mutación y cruce, y de una función de fitness que permite averiguar en cada paso la cercanía de la solución a la solución óptima del problema.

Un individuo de la población utilizada en el algoritmo es un conjunto de posibles valores de cada una de las siete variables anteriores, y como funciones de cruce se ha utilizado la selección aleatoria de genes de cada uno de los individuos padre y como función de mutación la obtención de un número aleatorio basado en una distribución de probabilidad Gaussiana.

La función de fitness por su parte, se ha definido tal y como se muestra en la Figura 4.2.

El objetivo de esta función es obtener los valores óptimos para las variables que permitan obtener los mejores patrones posibles en base a las secuencias del extractor.

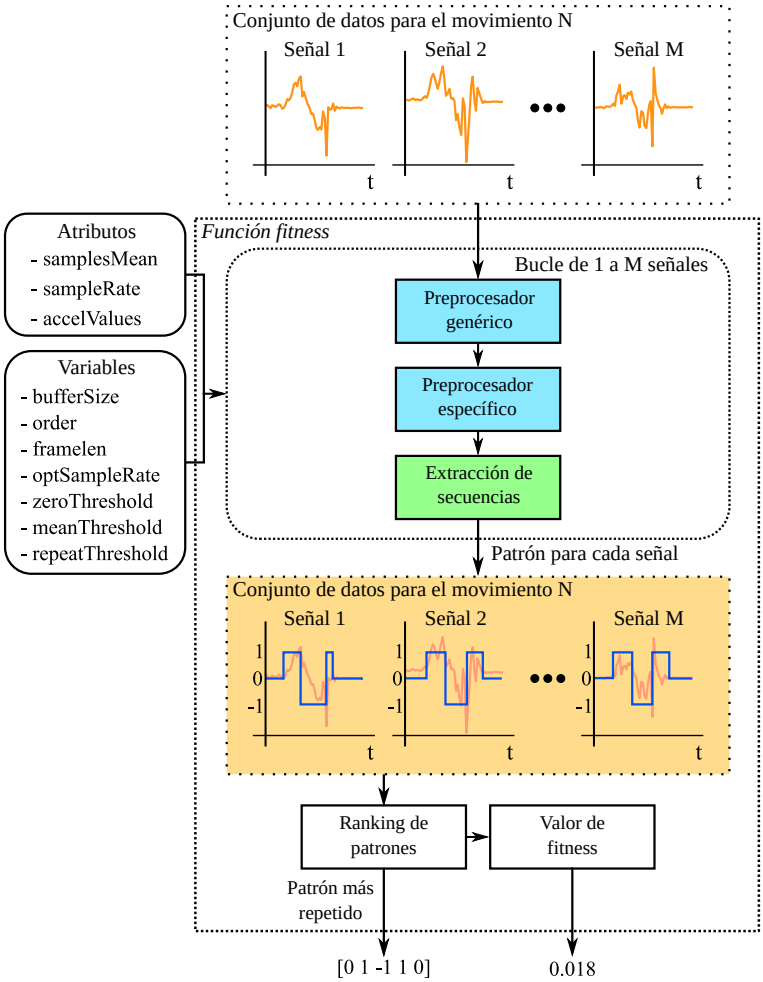

Figura 4.2: Diagrama representando la función de fitness utilizada para la optimización de variables de entrada mediante un Algoritmo Genético.

Esto implica que las variables serán mejores, cuantas más veces se repita la secuencia de salida si se aplica el algoritmo sobre distintas señales representando al mismo movimiento. Es por ello que la función se aplicará, en cada iteración del algoritmo genérico, sobre un conjunto de señales de muestra obtenidas al repetir el mismo movimiento un número M de veces.

Sobre cada una de estas señales, se ejecutarán los tres módulos (preprocesador genérico, preprocesador específico y extractor de secuencias) utilizando en cada iteración del GA una serie de valores posibles para las variables de entrada. Nótese que en la figura se diferencian las siete variables a optimizar de otros tres atributos de entrada que

se fijarán dependiendo de las señales de entrada en cada caso. Estos tres parámetros son:

- *samplesMean*: Es el valor medio de las aceleraciones en cada eje de la señal a analizar. Es, por tanto, un valor dependiente de la señal y del sensor del Smart Toy concreto. Se determina en base a un archivo con datos de calibración obtenido del propio dispositivo.
- *sampleRate*: Es la tasa de muestreo original del sensor del Smart Toy. Es utilizada, como ya se ha indicado anteriormente, por el preprocesador genérico para asegurar un muestreo uniforme de la señal.
- *accelValues*: Es el conjunto de posibles valores de tendencia de aceleración, tal y como se definió en la ecuación 4.2.

Hay que tener en cuenta que las variables a optimizar tienen una serie de restricciones, por lo que se debe configurar el algoritmo para que las tenga en cuenta:

- El valor mínimo para la variable *bufferSize* es de tres muestras. Esto es debido a que el sistema no sería capaz de determinar nunca la validez definitiva de una decisión temporal sobre una única muestra, ya que debe compararse al menos con la anterior y la posterior, según el funcionamiento del algoritmo.
- El valor máximo de esa misma variable (*bufferSize*) debe ser siempre menor que el número total de muestras. En cualquier caso, el valor debería limitarse a un número menor y acotado, ya que un conjunto demasiado grande de muestras temporales puede hacer que se termine recorriendo todas las muestras de la señal sin llegar a conseguir tomar una decisión definitiva sobre estas muestras. Por tanto, hemos limitado el tamaño máximo de este valor al 10 % del tamaño medio de muestras de las señales de entrada.
- Como ya se ha indicado anteriormente, las variables utilizadas para el filtro de suavizado tienen sus propias restricciones, concretamente el valor de *order* debe ser siempre menor al de *framelen* y que este último debe ser siempre un valor impar. Además, hay que tener en cuenta que si se seleccionan valores tales que $order = framelen - 1$, el filtro no produce ningún suavizado sobre la señal.

- El valor de muestreo variable *optSampleRate* debe ser siempre menor o igual a la tasa de muestreo original de la señal. Para evitar además tasas demasiado pequeñas, hemos limitado el mínimo a 10 Hz.
- Los valores umbral (*zeroThreshold*, *meanThreshold* y *repeatThreshold*) pueden tomar cualquier valor del conjunto de los números reales positivos ($\mathbb{R}$), pero se han limitado igualmente para favorecer una convergencia del algoritmo en un tiempo razonable. Los valores límites de los umbrales se han obtenido en base a los valores medidos durante la calibración de los juguetes.

En cada iteración de la función de fitness, se realizará, para cada señal del conjunto de datos de muestra, la ejecución de cada uno de los módulos, utilizando los tres parámetros fijos anteriores y las siete variables determinados por el Algoritmo Genético: Primero se realizará el conformado de la señal en base a la tasa de muestreo original (*sampleRate*), y se calculará el valor medio de las aceleraciones (*samplesMean)* para saber cuál es el valor medio de la aceleración. A continuación, se ejecuta el preprocesasdo específico. Para ello, se pasa la señal por el filtro de suavizado usando los valores *order* y *framelen*. Después se realiza un nuevo interpolado utilizando la tasa de muestreo variable (*optSampleRate*), y se pasa el resultado al módulo de extracción de secuencias, donde se utilizan los tres valores umbral (*zeroThreshold*, *meanThreshold* y *repeatThreshold*) junto con el tamaño máximo de muestras no definitivas (*bufferSize*) para obtener los valores correspondientes de tendencias de aceleración.

Al final de la iteración, se tendrán M conjuntos de valores de tendencias de aceleración, uno por cada señal del conjunto de señales de referencia. Estas señales se comparan entre sí, y se establece la secuencia más repetida. El valor de fitness será un valor dependiente del número de veces que se repite esa secuencia entre el total de secuencias obtenidas. Por tanto, el valor de fitness será menor cuanto mejores sean las variables de entrada en cuanto a capacidad de encontrar un patrón común a todas las señales. Idealmente, el valor de fitness de una solución óptima será 0. En la salida de esta función se devuelve, además del valor de fitness propiamente dicho, el conjunto de valores de la secuencia más repetido en cada caso.

Al finalizar la ejecución del Algoritmo Genético, se tendrá el valor de fitness más pequeño posible, y la secuencia que más veces se ha repetido durante la ejecución. Esta

será la secuencia utilizada como patrón de referencia para el movimiento específico que se esté optimizando.

4.3.3. Método de clasificación basado en el algoritmo de generación de patrones

Después de definir los procesos anteriores, es necesario definir el funcionamiento del sistema ante señales compuestas por varios movimientos. En esta sección se aborda la clasificación de este tipo de movimientos y se explica el proceso para la toma final de decisiones.

Una vez mostrados los métodos y algoritmos propuestos en la secciones 4.3.1 y 4.3.2, se describe en esta sección su aplicación en el sistema final, centrándonos en los últimos módulos de búsqueda de patrones y selección de movimientos. De forma general el sistema puede dividirse en tres fases: La primera de ellas se encargará de la generación de los patrones de referencia y la obtención de las variables óptimas para cada movimiento, a través del proceso explicado en las anteriores secciones. En la segunda fase, se tendrán tantos módulos de detección de movimientos (incluyendo en estos tanto el preprocesador específico del movimiento como el módulo de extracción de secuencias y un módulo de búsqueda de los patrones de referencia en las señales de entrada) como tipos de movimiento se deseen detectar. En una fase final, se tomarán las decisiones sobre las diferentes salidas de cada uno de los módulos anteriores y se clasificarán de esta manera los movimientos detectados. Este proceso se puede observar en la Figura 4.1.

4.3.3.1. Fase 1: Adquisición de patrón de referencia para un movimiento

Para poder obtener el patrón de referencia de un movimiento concreto, es necesario utilizar un conjunto de señales de entrada representativas del movimiento. Estas señales se obtienen mediante la repetición del movimiento en cuestión un número concreto de veces por varias personas. Esto aumenta la diversidad de las señales, lo cual es muy importante para que el patrón de referencia sea realmente común a la mayor parte de señales obtenidas para el movimiento.

A partir de este conjunto de datos, se ejecuta el sistema descrito en la sección 4.3.2, que permite optimizar los valores a utilizar posteriormente para el movimiento, así como obtener el patrón de referencia a partir de la función de fitness del Algoritmo Genético. El patrón obtenido se almacena para ser utilizado posteriormente por el módulo de búsqueda de patrones.

Se repite este proceso para los N movimientos a clasificar, obteniéndose así hasta N conjuntos de variables óptimas y patrones de referencia. En la Figura 4.3 se puede ver una representación de este proceso mediante un diagrama de bloques.

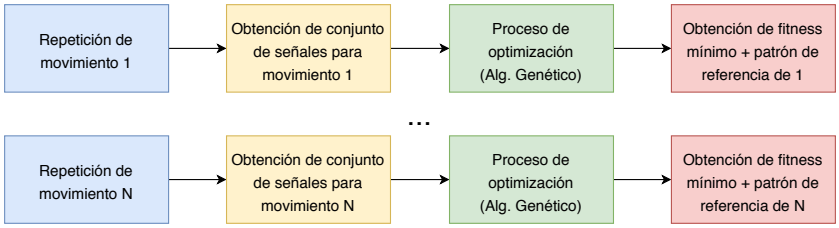

Figura 4.3: Diagrama de bloques representando la fase de adquisición de patrones de referencia para N movimientos.

4.3.3.2. Fase 2: Extracción de secuencias en paralelo

En esta fase, se tiene como entrada una señal compuesta por uno o varios movimientos que se deben detectar y clasificar. Esta señal, al igual que se hace con las señales del conjunto de datos de referencia, se debe pasar por el preprocesador genérico para obtener una señal que se pueda pasar a cada uno de los módulos definidos en la fase anterior.

En cada uno de estos módulos, se preprocesa la señal utilizando las variables optimizadas de cada movimiento, y se extraen en paralelo las secuencias detectadas en cada caso. Estas secuencias estarán formadas por series de etiquetas indicando las tendencias de aceleración en cada momento y serán la entrada del módulo de búsqueda de patrones.

Este módulo recorrerá la secuencia obtenida para la señal e irá comparando cada tendencia con las definidas como patrón de referencia del movimiento. Este módulo irá comparando las dos secuencias hasta encontrar una coincidencia total o parcial entre

la secuencia de entrada y la de referencia.

A la salida del módulo de búsqueda de patrones, se tendrá una estructura de datos que es dependiente del parámetro de interpolado *optSampleRate*, ya que la señal utilizada para extraer las secuencias había sido remuestreada con ese valor, pero no es algo relevante ya que los instantes temporales de salida corresponderán con los de la señal original. Esta estructura estará formada por una lista de ceros y unos, dependiendo de si cada muestra de la señal coincide con el patrón del movimiento o no. Además, se incluye en dicha estructura el grado de confianza con la que se ha detectado el movimiento, mediante la indicación de si se ha encontrado una coincidencia total (se indica una confianza de 1) o parcial (se indica una confianza basada en el número de muestras coincidentes dividida por el total de muestras que componen el patrón de referencia). El diagrama de bloques de la Figura 4.4 muestra el proceso en paralelo para la obtención de estas estructuras.

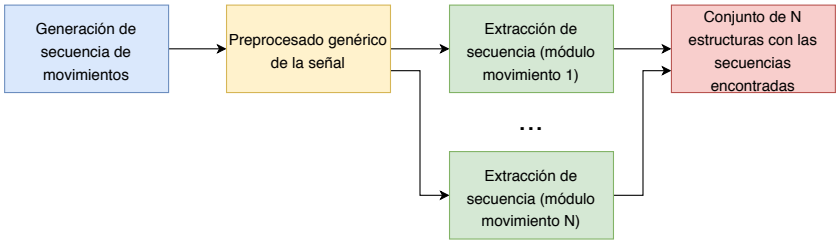

Figura 4.4: Diagrama de bloques representando la fase de extracción de secuencias en paralelo.

En la Figura 4.5 se puede ver un ejemplo de este proceso. En la Figura 4.5-a se tiene la señal de entrada y los valores de tendencia de aceleración correspondientes en cada muestra. En la Figura 4.5-b se pueden ver las detecciones realizadas por cada uno de los módulos de clasificación. Cada línea coloreada representa el grado de confianza en la detección de un tipo de movimiento en cada muestra. Se puede ver que en algunas muestras se detectan varios posibles movimientos, pero con diferentes grados de confianza. Finalmente, en la Figura 4.5-c se muestra el resultado de la decisión de qué movimiento se ha realizado en cada caso. Este último proceso se describirá en la siguiente sección.

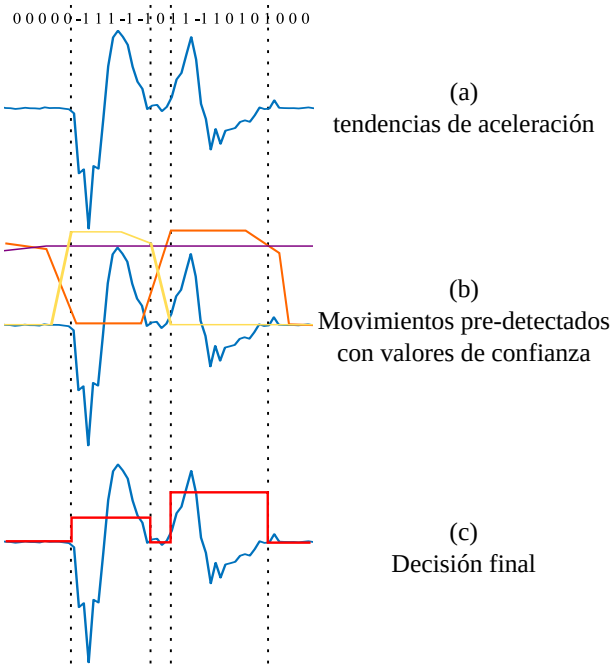

Figura 4.5: Ejemplo de detección y clasificación de movimientos.

4.3.3.3. Fase 3: Selección de movimientos

El proceso de decisión final sobre qué movimiento se debe detectar en cada caso se realiza a través de las características de la señal original junto con las decisiones parciales que se obtienen de cada módulo en la fase anterior.

Para esto, en esta fase se realiza una interpolación inversa de los N resultados anteriores, para poder obtener los valores correspondientes a las muestras originales de la señal. Para ello se utiliza el inverso de la variable *optSampleRate* como parámetro para la función de interpolación.

A partir de estos nuevos resultados, se compara cada estructura de datos para determinar cuál de ellas ofrece una mayor confianza en cada muestra. Si solo en una de las estructuras se ha detectado un movimiento, se elige ese como decisión final. Si por el contrario la misma muestra aparece en más de uno de los resultados, se elige aquel que ofrezca un valor más alto de confianza.

En el caso de tener varias decisiones con el mismo valor de confianza, se elige

aquel movimiento que derive de un patrón más largo, al ser más probable que si se ha detectado el patrón complejo, estemos ante ese movimiento. En la Figura 4.6 se puede ver este proceso mediante un diagrama de bloques.

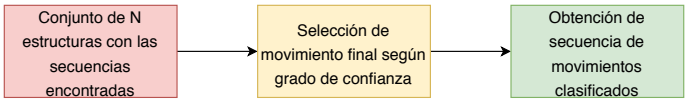

Figura 4.6: Diagrama de bloques representando la fase de selección final de movimientos.

Al finalizar esta fase, se tendrá una única estructura de datos conteniendo, para cada muestra de la señal original, una decisión sobre si contiene parte de un movimiento y qué movimiento es, tal y como se ha podido ver en el ejemplo de la Figura 4.5-c.

4.4. Configuración de experimentos: comparación con otros métodos

Una vez definida la propuesta, se ha realizado un estudio experimental comparativo para poder validarla y determinar su grado de acierto y precisión. Para ello, se ha implementado el sistema junto con otros dos sistemas basados en métodos comunes para la detección de actividades en base a señales, y se han diseñado varios experimentos que se han llevado a cabo con los tres métodos.

Los experimentos se han llevado a cabo con los prototipos descritos en el capítulo 2, configurados para enviar todos los datos leídos en los sensores sin preprocesar ni modificar.

Se ha configurado el dispositivo de gateway para ser capaz de recibir los datos de dos formas diferentes, dependiendo de si se van a utilizar los datos para obtener los conjuntos de datos de entrenamiento de cada movimiento o se van a tomar datos de actividades compuestas por varios movimientos para detectar y clasificar éstos. En el primer caso, las muestras de cada movimiento separado se almacenan, de forma automática, en ficheros separados, uno por cada conjunto de muestras. En el segundo, se almacenan todas las muestras recibidas en un único fichero que contendrá la secuencia

de movimientos completa. Para poder aislar las muestras de cada movimiento en el primer caso, se ha añadido al gateway un detector de movimientos basado en la cercanía o lejanía de los valores de aceleración a la media de la señal, utilizando valores de calibración para configurar el sistema.

Los otros dos sistemas seleccionados para la comparativa se basan en la aplicación de algoritmos de clasificación de Machine Learning (como veremos a continuación, se han probado una serie de modelos de clasificación y finalmente se ha optado por la utilización de aquel que ofrecía mayor precisión en el proceso de validación) y en la búsqueda de señales en función de su similitud con una señal de referencia, mediante la utilización de funciones de cálculo de distancia entre señales.

Tanto estos dos sistemas como la propuesta de este trabajo se han implementado en Matlab R2018a. En las siguientes secciones se describen cada una de las implementaciones, y posteriormente se definen los experimentos realizados.

4.4.1. Método basado en la distancia entre señales

Este método de detección de movimientos se ha construido en base al cálculo de distancias Euclídeas cuadradas entre señales. Para ello, se comparan las muestras de cada señal con las de una señal de referencia. Si las distancias entre ambas son suficientemente pequeñas, el sistema determina que se ha encontrado un determinado movimiento y las muestras se clasifican como tal, devolviéndose los instantes de inicio y fin del movimiento y el valor calculado de distancia.

Para la implementación de este método, nos hemos apoyado en la función *findsignal* de Matlab [249], ya que precisamente hace está búsqueda entre dos señales de entrada.

En la Figura 4.7 se puede ver un ejemplo de este método. A la izquierda se puede ver la señal de referencia y a la derecha, la señal sobre la que se produce la búsqueda en azul, y en rojo las muestras que la función ha determinado que son suficientemente similares a la señal buscada.

Este método, por sus propias características, permite la detección de señales de un único tipo de movimiento en cada caso, pero por si solo no es capaz de clasificar

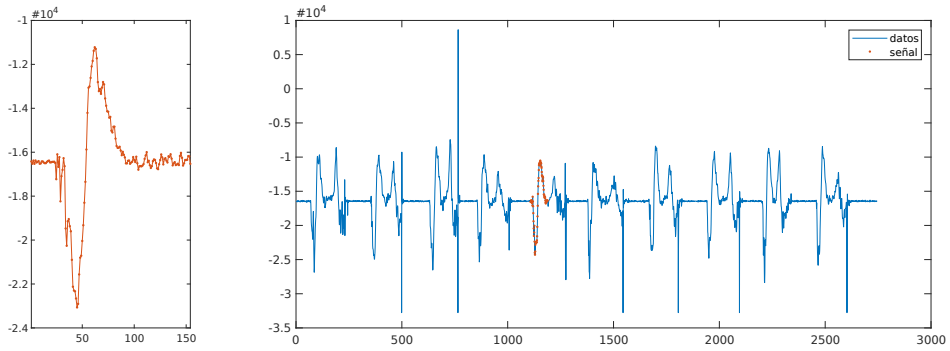

Figura 4.7: Ejemplo del funcionamiento de la función de búsqueda de similitud de señales.

diferentes movimientos en una misma secuencia de aceleraciones. Para poder usarlo como clasificador, se han generado una serie de módulos de detección, uno por cada movimiento a detectar, y se ha utilizado en cada uno de ellos una señal de referencia del movimiento específico. Esta señal de referencia se consigue a través del conjunto de datos de entrenamiento definido en la sección anterior, obteniendo una señal media de todas las repeticiones de cada movimiento.

A la salida de estos módulos se tienen una serie de secuencias de muestras detectadas como parte de movimientos. Dado que cada módulo procesa la señal de entrada de forma independiente, es posible que se tengan muestras detectadas como parte de más de un tipo de movimiento en la salida. Para tomar una decisión sobre qué movimiento es más probable en cada caso, se ha determinado que el movimiento clasificado finalmente será aquel cuya distancia sobre la señal de referencia correspondiente sea menor. De esta forma, es posible tener a la salida del sistema una única estructura de datos indicando las muestras de la señal original que pertenecen a cada uno de los posibles movimientos.

La principal desventaja de este sistema radica en que la función de búsqueda de señales similares no es capaz de determinar *a priori* la distancia máxima a partir de la cual un conjunto de muestras no se debería tener en cuenta como posible movimiento detectado. La función toma este valor como parámetro, o alternativamente, es necesario indicarle a la función el número de señales a buscar.

4.4.2. Método basado en un modelo Support Vector Machine (SVM)

En la sección 4.2 se ha podido comprobar cómo la mayoría de los trabajos relacionados con la detección de actividades humanas se basan en técnicas de Machine Learning. Esto es debido a que estas técnicas ofrecen una alta precisión en la clasificación de actividades repetitivas, donde es posible determinar una serie de atributos o características de cada una y se pueda posteriormente entrenar un modelo de aprendizaje en base a éstas.

Por tanto, una de las aproximaciones llevadas a cabo en este trabajo para la posible clasificación de los movimientos realizados con los Smart Toys es la utilización de alguno de estos mecanismos.

Existen una gran cantidad de algoritmos y modelos disponibles para la clasificación de elementos en base al aprendizaje supervisado. Por ello, y con el objetivo de utilizar el mejor método posible, se han probado una serie de métodos distintos utilizando en todos los casos un grupo de características comunes obtenidas a partir de los conjuntos de datos de entrenamiento definidos al principio de la sección 4.4.

Estos atributos, siguiendo los ejemplos de los trabajos relacionados, se han clasificado de acuerdo a si utilizan características basadas en el dominio del tiempo o características basadas en el dominio de la frecuencia. De entre las primeras, se han seleccionado atributos como la media, los valores máximos y mínimos de aceleración, la desviación estándar, el valor de asimetría (*skewness*), el valor de curtosis, la desviación media absoluta (MAD), el valor de entropía y la posición y altura de los picos de la señal. En el dominio de la frecuencia se han seleccionado como atributos los picos más altos de la transformación de Fourier de la señal (FFT) así como sus valores de frecuencia, y los resultados de realizar un análisis cepstral complejo (CCEPS).

Los atributos anteriores se han calculado para cada una de las señales de entrenamiento, y cada vector de atributos se ha marcado como perteneciente al movimiento adecuado. A continuación, se ha aplicado esta matriz de atributos a una serie de algoritmos de aprendizaje supervisado, configurando la validación a partir de un sistema de validación cruzada con 5 iteraciones.

Los algoritmos probados han sido los siguientes:

- Algoritmos basados en árboles de clasificación, con modelos basados en árboles "finos", "medios" y "gruesos".
- Algoritmos basados en Máquinas de Vectores de Soporte (Support Vector Machines (SVM)), con modelos lineales, cuadráticos y cúbicos.
- Algoritmos basados en "k vecinos más próximos" (k-Nearest Neighbors (kNN)), también con modelos "finos", "medios" y "gruesos".
- Algoritmos basados en discriminantes lineales y cuadráticos.

De entre todos estos métodos, se ha seleccionado aquel que, en el proceso de validación ha ofrecido un mayor grado de acierto, que ha sido el modelo basado en una Máquina de Vectores de Soporte Cuadrática (Quadratic SVM), con un 95.3 % de precisión en el acierto. En la Figura 4.8 se puede ver la matriz de confusión resultante de la validación del modelo (cada clase corresponde con los tres movimientos definidos, "up", "down" y "stack" respectivamente, tanto en filas como en columnas). Se puede observar que, en el proceso de validación, la precisión obtenida en la detección de cada uno de los movimientos es muy similar, sólo ligeramente inferior para los movimientos hacia abajo ("down").

El modelo basado en Máquinas de Vectores de Soporte (SVM), es un algoritmo utilizado frecuentemente en problemas tanto de clasificación como de regresión. El modelo, en su entrenamiento representa cada muestra de entrada en un espacio de tal forma que las clases a clasificar se separen lo más posible a través de un hiperplano (un plano de n dimensiones). Este hiperplano se denomina Vector Soporte. Una vez realizado el entrenamiento, la clasificación se produce dependiendo del espacio (separado por dichos vectores) en el que se pueda representar.

Hay que tener en cuenta que este porcentaje tan alto de acierto se basa en la validación con las mismas señales de entrenamiento, es decir, señales aisladas de cada movimiento que ya se ofrecen al modelo correctamente "recortadas" de la actividad completa.

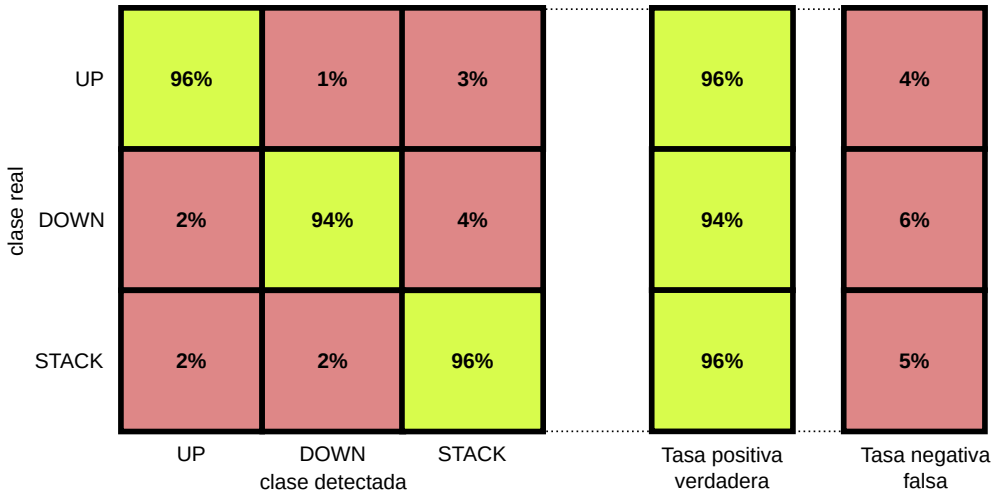

Figura 4.8: Matriz de confusión con los resultados de la validación del modelo de Máquina de Vectores de Soporte cuadrático (precisión media del 95,3 %).

Sin embargo, esto plantea un problema para la utilización del método basado en SVM con las señales de entrada de las actividades de juego que pretendemos clasificar, ya que estas señales estarán formadas por varios movimientos incluidos en una única señal continua.

Para solucionar esto, se planteó inicialmente la posibilidad de ir generando ventanas de tamaño fijo de muestras de la señal original, y utilizar estos vectores de muestras en el modelo de clasificación. Se ha probado este sistema con varios tamaños de ventana y con ventanas solapadas (es decir, ventanas en las que las últimas muestras se repetirán en la muestra anterior). Sin embargo, esta aproximación ha devuelto resultados muy pobres en cuanto a la detección de los movimientos. Es muy difícil determinar el tamaño óptimo de estas ventanas para todos los posibles movimientos, más aún cuando estos movimientos se producirán a intervalos irregulares durante las actividades.

Finalmente, se ha utilizado un sistema de predetección de los movimientos en conjunción con el modelo SVM. Esta predetección se basa en los datos de calibración del juguete del que provienen los datos de la actividad, y, en base a la aceleración media de la señal, permite determinar qué muestras pertenecen a algún movimiento y qué muestras no. Cada una de estas señales formadas por un subconjunto de la señal completa son los valores que se incluyen en el modelo de clasificación.

4.4.3. Método propuesto

Además de los dos métodos anteriores, se ha llevado a cabo una implementación completa del método propuesto en la sección 4.3. Para la generación de cada uno de los N módulos de extracción de secuencias y búsqueda de patrones se han utilizado los mismos conjuntos de datos de referencia usados en los dos métodos anteriores, en este caso, para obtener un patrón de referencia de cada uno de los movimientos, y para optimizar las variables de entrada de los módulos en cada caso, a través del método propuesto, basado en un Algoritmo Genético.

La salida de cada uno de estos módulos se utiliza en el clasificador implementado para obtener finalmente los instantes de inicio y fin de cada movimiento y su clasificación.

4.4.4. Experimentos realizados

Para cada una de las implementaciones anteriores, se han definido una serie de experimentos que permiten comparar los métodos y determinar la precisión de nuestra propuesta en las tareas de detección y clasificación.

Para ello, primero se han determinado una serie de movimientos básicos que se puedan repetir durante una actividad de juego utilizando los Smart Toys prototipo definidos en este trabajo. Los prototipos utilizados han sido los Smart Cubes definidos en la sección 2.4.2, y como actividad se ha utilizado la misma que se utilizó en los experimentos de las pruebas piloto de validación, es decir, el apilamiento de cubos.

De esta actividad se han extraído los siguientes tres movimientos básicos, que pueden también verse de forma gráfica en la Figura 4.9:

- Movimiento hacia arriba ("Up", Figura 4.9-a):
    1. El movimiento se inicia con el Smart Cube reposando estable sobre una superficie plana.
    2. Cogiendo el juguete con una mano, se realiza un movimiento vertical hacia arriba desde la superficie, de unos 20 o 25 cm aproximadamente.

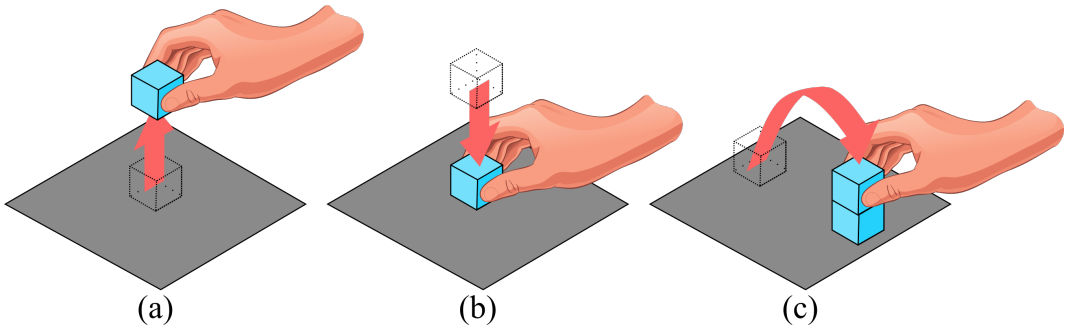

Figura 4.9: Movimientos utilizados durante la experimentación para su detección y clasificación (movimiento hacia arriba (a), hacia abajo (b) y movimiento de apilamiento (c).

3. El movimiento se da por finalizado cuando el cubo se estabiliza en el aire, aún en la mano del jugador.

- Movimiento hacia abajo ("Down", Figura 4.9-b):

  1. El movimiento se inicia con el Smart Cube en la mano del jugador, que lo sostiene en el aire.
  2. A continuación se produce un movimiento vertical hacia abajo, hacia una superficie estable.
  3. El movimiento termina cuando el cubo se estabiliza en la superficie a la que se dirigía el movimiento.

- Movimiento de apilamiento ("Stack", Figura 4.9-c):

  1. En este caso, el movimiento se inicia de forma similar al movimiento hacia arriba, con el Smart Cube reposando estable sobre una superficie plana.
  2. A continuación se produce un movimiento hacia arriba sosteniendo el cubo con la mano, realizando un movimiento en arco hacia otro punto de la superficie.
  3. Finalmente, se deja reposar el cubo sobre ese nuevo punto, que puede estar a la misma altura del inicio del movimiento o no (ya que se trata de un movimiento de apilamiento).

Con estos movimientos definidos, se ha procedido a generar el conjunto de datos de

referencia de cada uno. Se ha pedido a 5 personas diferentes que realizaran de forma repetida cada movimiento, hasta tener 300 señales de muestra de cada uno.

Esas 900 muestras se han utilizado en cada uno de los métodos como señales de referencia: En el método de búsqueda de similitud entre señales, se utilizan las señales de referencia para la generación de las señales de comparación y cálculo de distancias. En el método basado en SVM se utilizan las señales para la extracción de los atributos y por tanto para entrenar el modelo, y finalmente en nuestra propuesta se utilizan las señales para la construcción de los módulos de búsqueda de patrones de cada movimiento.

En este último caso, las señales se han utilizado como elementos sobre los que ejecutar la función de optimización basada en el Algoritmo Genético. En la tabla 4.1 se pueden ver los resultados de aplicar dicha función sobre cada uno de los 3 movimientos definidos para los experimentos.

Tabla 4.1: Tabla con los resultados de optimización para cada una de las variables de entrada y cada uno de los movimientos utilizados en los experimentos.

Variables		**Movimiento**		
		Up	**Down**	**Stack**
bufferSize		3	5	4
Parametros de Savitzky-Golay	*order*	1	1	1
	framelen	9	11	11
optSampleRate (Hz)		37.0541	23.8327	15.9446
zeroThreshold (G)		0.237906	0.06103	0.031598
meanThreshold (G)		0.209924	0.098393	0.037888
repeatThreshold (%)		88.62	88.63	78.81

Como se puede observar en la tabla, los valores son diferentes para cada uno de los movimientos, aunque hay algunos resultados similares: El valor óptimo de *bufferSize* está alrededor de entre 3 y 5 muestras que se pueden mantener en estado no definitivo, dependiendo del movimiento. El orden polinomial del filtro de suavizado se ha fijado en 1 para los tres movimientos, siendo el tamaño de las tramas algo menor para el movimiento hacia arriba que para los otros dos. Las variaciones entre movimientos se pueden observar especialmente en las tasas variables de muestreo para la segunda interpolación (variable *optSampleRate*, así como en los valores de los umbrales de deci-

sión (*zeroThreshold*, *meanThreshold* y *repeatThreshold*, pese a que este último presenta valores similares en los movimientos hacia arriba y hacia abajo.

Como se comentó en la definición del algoritmo de optimización, a la salida de la función no se obtienen únicamente estos valores optimizados, sino que también se obtiene el valor de fitness más bajo alcanzado (recuérdese que este valor indica el número de veces que se repite la misma secuencia sobre el total de secuencias calculadas). Para las muestras utilizadas en los experimentos, el valor de fitness a la salida del optimizador fue 0 para los tres movimientos.

Además, esta función devuelve la propia secuencia de tendencias de aceleración patrón en cada caso, que será la más repetida (en el caso del fitness 0, es la secuencia obtenida para todas las señales de entrada). Teniendo en cuenta que hemos fijado los valores de tendencia *accelValues* como el conjunto formado por 1 para aceleraciones positivas, -1 para aceleraciones negativas y 0 para la ausencia de aceleraciones (véase la expresión 4.2), las secuencias patrón de cada movimiento son:

- Para el movimiento hacia arriba ("Up"): $\begin{bmatrix} 0 & -1 & 1 & 0 & 1 & -1 & 0 \end{bmatrix}$
- Para el movimiento hacia abajo ("Down"): $\begin{bmatrix} 0 & 1 & -1 & 1 & 0 \end{bmatrix}$
- Para el movimiento de apilación ("Stack"): $\begin{bmatrix} 0 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$

Estas secuencias serán las que se buscarán en las señales de entrada para detectar y clasificar cada uno de los posibles movimientos.

Una vez se tienen los tres métodos configurados y "entrenados", se han definido tres experimentos para validarlos:

- Experimento 1: Se ha tomado una señal de cada tipo de movimiento (hacia arriba, hacia abajo y de apilamiento) de forma aislada, de forma similar a las señales de referencia usadas durante el entrenamiento. Los tres métodos de clasificación se han probado con cada una de las tres señales.
- Experimento 2: Se han tomado una serie de movimientos repetitivos de forma continua de cada uno de los movimientos. Los movimientos hacia arriba y hacia

abajo, dada su dependencia, se han tomado en una única secuencia que combina ambos (un movimiento hacia arriba seguido de un movimiento hacia abajo), mientras que la secuencia de repeticiones de movimientos de apilamiento se ha hecho de manera independiente. Por tanto, cada una de las dos secuencias de movimientos repetitivos se ha probado con cada uno de los tres métodos.

- Experimento 3: Se ha tomado una única secuencia compuesta por repeticiones de los tres movimientos posibles. Esta secuencia también ha sido probada con los tres métodos de clasificación.

Además, para cada una de las señales utilizadas en los experimentos, se ha realizado una selección visual de los instantes de inicio y fin de cada movimiento, así como una clasificación manual de los mismos. Esto permite tener una referencia ideal sobre la que comparar cada uno de los tres métodos de clasificación.

4.5. Resultados

En este apartado se describen los resultados de los experimentos definidos en la sección 4.4. Como se ha indicado al final de dicha sección, se han determinado para cada uno de los experimentos los movimientos a detectar, así como su posición en cada señal. En la tabla 4.2, se pueden observar los valores de cada tipo de movimiento que se deben detectar de manera óptima en cada uno de los experimentos.

Tabla 4.2: Valores de referencia óptimos para cada movimiento y experimento

Experimento	**Tipo de señal**	**Detección óptima**		
		Up	**Down**	**Stack**
1	*Hacia arriba (Up) única*	1	0	0
	Hacia abajo (Down) única	0	1	0
	Apilación (Stack) única	0	0	1
2	*Serie hacia arriba y hacia abajo (Up-Down)*	10	10	0
	Serie de movimientos de apilamiento (Stack)	0	0	10
3	*Serie con los tres movimientos (Up-Down-Stack)*	6	6	6

En la tabla, la columna etiquetada como “Detección óptima”, indica el número

exacto de movimientos de cada tipo que se deberían clasificar en cada una de las señales de entrada de cada experimento para determinar que el clasificador ha alcanzado un 100 % de acierto.

Los resultados de cada uno de los experimentos se pueden ver de forma gráfica en las Figuras 4.10, 4.11 y 4.12.

En la Figura 4.10 se puede observar como hay nueve gráficas, una por cada señal aislada en cada uno de los tres métodos implementados. Las líneas azules punteadas representan la señal original a clasificar, y la línea roja la clasificación en cada caso, siendo la altura de la línea la que determina el tipo de movimiento. La primera fila muestra los resultados para el método de búsqueda de similitud de señales, la segunda para el método basado en SVM y la última indica la clasificación de nuestra propuesta. En la Figura 4.11, se pueden ver los resultados del segundo experimento. En este caso se tienen dos gráficas por fila, al haber únicamente dos secuencias de movimiento (movimientos hacia arriba y hacia abajo y movimientos de apilamiento). En la Figura 4.12, se tiene una gráfica por fila para la secuencia única que contiene los tres tipos de movimiento.

Los resultados gráficos que se observan en estas figuras se pueden ver en la tabla 4.3.

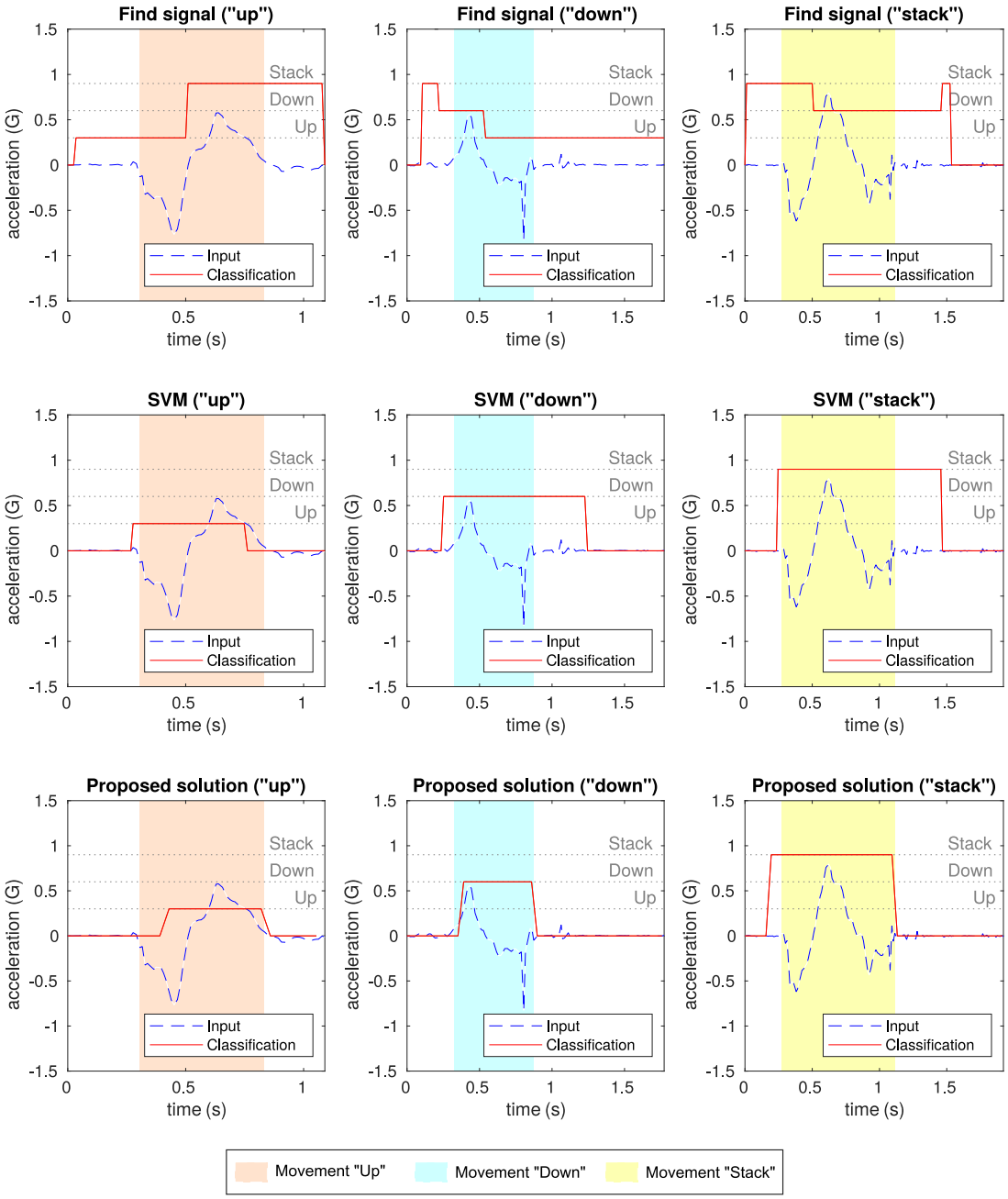

Figura 4.10: Gráficas con los resultados para el experimento 1: En cada fila se tiene una gráfica por cada uno de los movimientos aislados del experimento, y en cada columna se ven los resultados de cada método.

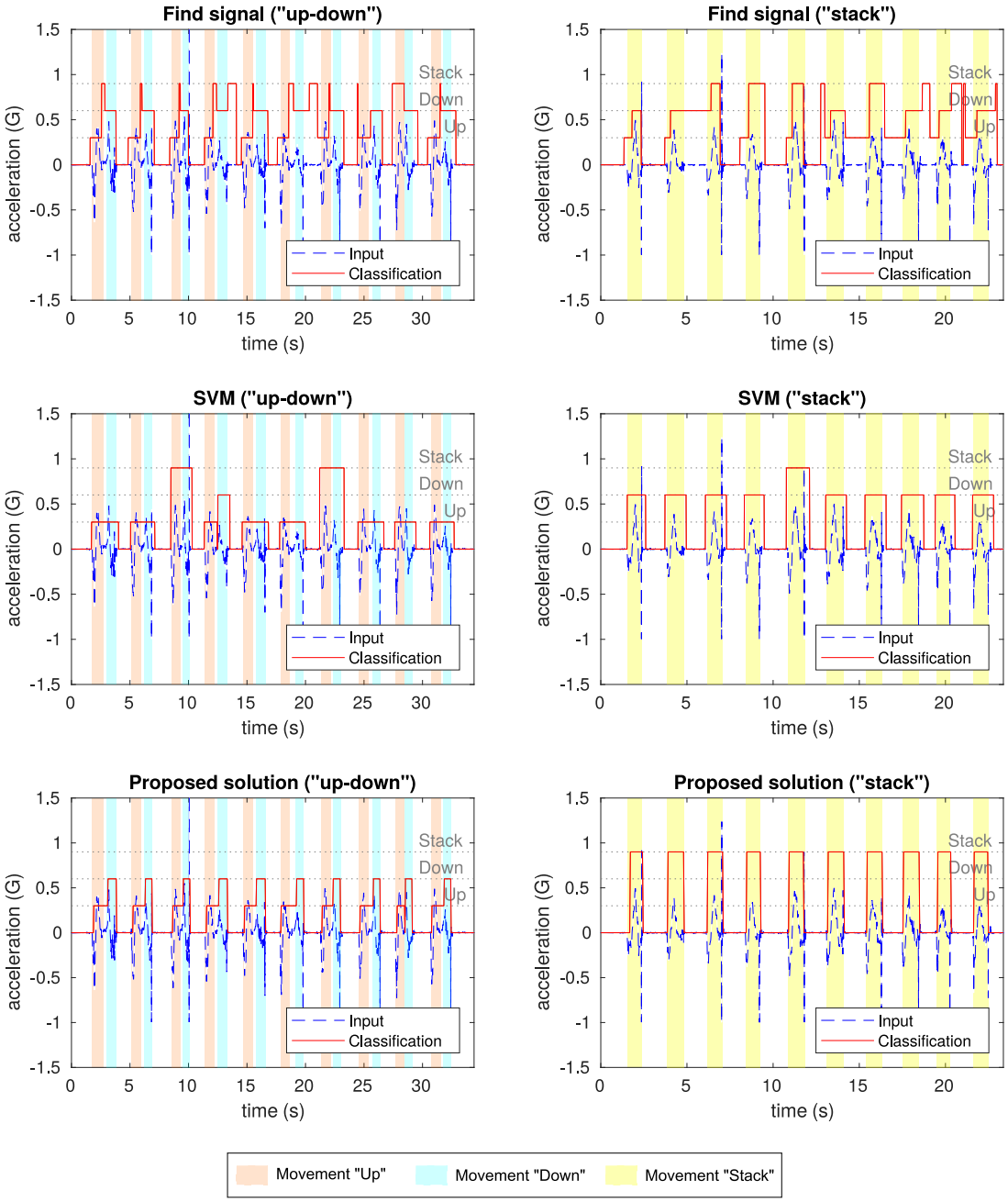

Figura 4.11: Gráficas con los resultados para el experimento 2: En cada fila se tiene una gráfica por cada una de las secuencias de movimientos (arriba y abajo y apilación) del experimento, y en cada columna se ven los resultados de cada método.

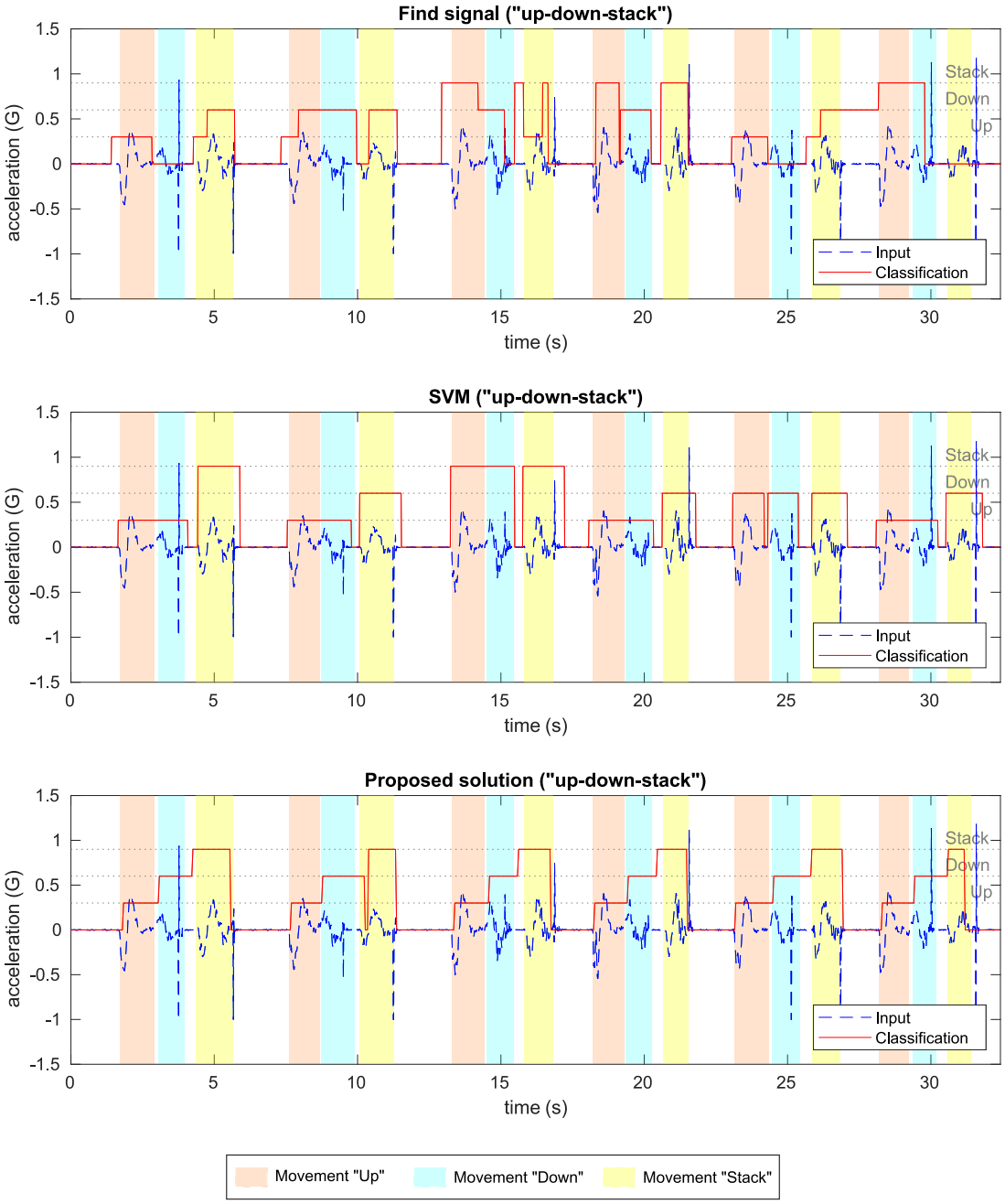

Figura 4.12: Gráficas con los resultados para el experimento 3: En cada fila se tiene una gráfica por cada una de las secuencias de los tres movimientos detectada por cada uno de los métodos.

Tabla 4.3: Resultados de los experimentos con los tres métodos comparados.

Experimento	Método de clasificación	Tipo de Señal	Movimiento Detectado			Tasa de Error*	Distancia**	
			Up	Down	Stack		Media	Desv. estándar
1	*Búsqueda de similitud de señales*	*"Up" aislado*	1	0	1	1	1.1269	-
		"Down" aislado	1	1	1	2	0.7942	-
		"Stack" aislado	0	1	2	2	1.0261	-
	Quadratic Support Vector Machine	*"Up" aislado*	1	0	0	0	0.1913	-
		"Down" aislado	0	1	0	0	0.8087	-
		"Stack" aislado	0	0	1	0	0.4513	-
	Solución propuesta	*"Up" aislado*	1	0	0	0	0.2813	-
		"Down" aislado	0	1	0	0	0.1631	-
		"Stack" aislado	0	0	1	0	0.1130	-
2	*búsqueda de similitud de señales*	*Secuencia "Up-Down"*	9	10	12	0.6316	0.4659	0.2633
		Secuencia "Stack"	9	6	9	5	0.1657	0.2225
	Quadratic Support Vector Machine	*Secuencia "Up-Down"*	8	1	2	0.2222	0.6018	0.7807
		Secuencia "Stack"	0	9	1	9	0.0412	0.1304
	Solución propuesta	*Secuencia "Up-Down"*	10	10	0	0	0.3701	0.2321
		Secuencia "Stack"	0	0	10	0	0.1465	0.0674
3	*búsqueda de similitud de señales*	*Secuencia "Up-Down-Stack"*	6	6	6	1.25	0.2203	0.3369
	Quadratic Support Vector Machine	*Secuencia "Up-Down-Stack"*	4	6	3	0.8571	0.4213	0.4529
	Solución propuesta	*Secuencia"Up-Down-Stack"*	6	6	6	0	0.2796	0.1279

* La Tasa de error es la relación entre el número de movimientos detectado erróneamente y el número de movimientos detectaso correctamente.
** La Distancia es el valor absoluto de la diferencia entre los instantes de tiempo de inicio y fin del movimiento real y el detectado, divididos por la duración del movimiento real de referencia. En el experimento 1, la media corresponde a un único movimiento, y por tanto no se calcula la desviación estándar.

El número de movimientos detectados no es una métrica suficiente para determinar la precisión de cada uno de los métodos. Hay que tener en cuenta también el grado de coincidencia entre el movimiento detectado y el movimiento real en cuanto a su duración y a su posición en la señal original.

Para medir estos dos factores, hemos incluido dos mediciones más en los resultados, tal y como se puede ver en la tabla 4.3:

- Tasa de error: Es la relación entre el número de movimientos detectados de forma errónea (en cuanto a tipo de movimiento o su posición) y el número de movimientos detectados correctamente, tal y como se puede ver en la ecuación 4.7.

$$Tasa\ de\ error = \frac{Movimientos\ detectados\ erróneamente}{Movimientos\ detectados\ correctamente} \tag{4.7}$$

- Distancia: Se determina como el valor absoluto de las diferencias entre los instantes de tiempo de inicio y fin de los movimientos reales y los detectados, tal y como se muestra en la ecuación 4.8:

$$Distancia = \frac{|t_{bd} - t_{br}| + |t_{ed} - t_{er}|}{t_{er} - t_{br}} \tag{4.8}$$

  Donde t_{bd} y t_{br} son los instantes de tiempo de inicio del movimiento detectado y real, respectivamente, y t_{ed} y t_{er} son los instantes de tiempo de final de movimiento equivalentes.

Dado que hay más de un movimiento por señal en algunos de los casos de los experimentos, se ha optado por mostrar en la tabla de resultados el valor medio de la distancia, así como la desviación estándar en cada caso.

Los resultados para cada uno de los métodos de clasificación son:

- Para el método de búsqueda de señales, se tiene un alto número de movimientos detectados de manera errónea. Se puede observar que este método presenta problemas tanto para clasificar adecuadamente las señales aisladas del experimento 1, como para las secuencias de señales de los otros dos experimentos. Si

prestamos atención a las distancias entre los movimientos reales y los detectados, se tiene una gran variedad de resultados. En algunos casos, se tienen distancias mayores que 1, lo cual implica que el valor real y el detectado están a más de una unidad de distancia, pero también se pueden ver algunos resultados mucho más precisos, como por ejemplo la distancia media en el experimento 3, que es de tan solo 0,2203.

- El método basado en el modelo SVM cuadrático ofrece unos resultados especialmente precisos en el experimento 1, cuando se prueba con señales aisladas. Esto es debido precisamente a que el entrenamiento del modelo se produjo con señales muy similares a las utilizadas durante el experimento, lo cual facilita el trabajo del modelo. Sin embargo, cuando se utilizan secuencias de señales como las de los experimentos 2 y 3, se tienen altas tasas de error y una gran variabilidad en cuanto a las distancias con respecto a los movimientos reales. El problema se puede achacar a la incapacidad del método para detectar el inicio y fin de los movimientos en secuencias compuestas por más de uno de éstos. Además, el mecanismo propuesto para predetectar las señales que componen cada movimiento antes de su clasificación no ofrece una suficiente precisión para que el método sea confiable.

- Finalmente, en el sistema propuesto, se puede observar que el valor de la tasa de error es 0 en los tres experimentos. Además, las distancias con respecto a los movimientos reales están acotadas, aunque es cierto que son algo mayores que utilizando otros métodos en algunos casos. Esto se debe a que, al no existir detecciones erróneas, el valor de la distancia se calcula sobre un número mayor de señales, y por tanto el valor medio baja. Este valor medio, sin embargo, no varía demasiado de un experimento a otro, siendo el mayor valor 0,3701, es decir, en ese caso la señal se encuentra a un 37,01 % de distancia en cuanto a posición y tamaño del movimiento real. Además, observando las desviaciones típicas de este sistema, se puede ver que son menores que las de otros dispositivos, lo cual indica una mayor cercanía en general a los movimientos reales en este caso.

A partir de estos resultados, y de las características de cada método se pueden realizar una serie de afirmaciones sobre cada uno:

- El método de búsqueda de señales tiene una muy importante desventaja para su uso como clasificador, tal y como se ha descrito en la sección 4.4.1, ya que no es capaz de determinar la distancia máxima para considerar un conjunto de muestras como posible correspondencia con la señal de referencia. En las pruebas realizadas, se han especificado el número de señales a buscar para obtener un resultado realista, pero en un entorno real donde el número de movimientos en cada caso no pueda ser determinado *a priori*, esto no podría realizarse. En lugar de esto, se debería determinar una distancia máxima media por movimiento para fijar el límite de búsqueda de señales. En cualquier caso, esta limitación hace que en este método se produzca una "sobreestimulación" del clasificador, ya que los módulos de búsqueda están forzados a encontrar un número determinado de señales, lo cual lleva a un alto número de detecciones erróneas.

- El método basado en la técnica de aprendizaje supervisado Support Vector Machine ha presentado también una importante desventaja, en este caso en la capacidad de detectar los instantes de inicio y fin de movimiento en secuencias que contienen varios movimientos de forma no uniforme. Se ha mitigado el problema utilizando un sistema de predetección basado en la búsqueda de los instantes de tiempo donde las aceleraciones pasan de estar cercanas a 0, a tener valores grandes, a partir de los datos de calibración de los Smart Toys. Este sistema no es suficientemente preciso, especialmente para movimientos que no empiecen o finalicen de forma abrupta, o en movimientos realizados de forma lenta.

- El movimiento de apilamiento puede ser considerado como una mezcla parcial de los movimientos hacia arriba y hacia abajo, ya que "incluye" estos dos movimientos en sí mismo. Esto implica que sea más difícil de clasificar correctamente tanto por el método de búsqueda de similitud de señales como por el método basado en SVM. Se puede ver en los resultados que en varias ocasiones se clasifican estos movimientos como un par de movimientos hacia arriba y hacia abajo. Sin embargo, nuestra propuesta basada en la obtención de un patrón de referencia sí es capaz de distinguir este tipo de movimiento correctamente, al ser capaz de encontrar una secuencia de aceleraciones que lo caracteriza de forma distinta a los otros dos movimientos.

- Además de los buenos resultados ofrecidos por el método propuesto (ha sido

el único capaz de detectar y clasificar de forma correcta todos los movimientos durante los experimentos), presenta la importante ventaja de ser completamente autónomo, al no requerir un proceso de predetección de los movimientos ni necesitar ser configurado para la búsqueda de un número determinado de señales.

- Una ventaja adicional de este método radica en la utilización de un algoritmo paso a paso que sólo tiene en cuenta en cada iteración la muestra "actual" de la señal de aceleración y, dado el caso, las muestras anteriores. Por tanto, la adaptación de este sistema para ser utilizado en tiempo real (e incluso ser instalado en los propios dispositivos) es relativamente inmediata. Esto es una gran ventaja ya que el sistema podría ser utilizado para detectar, clasificar y almacenar los movimientos según se están produciendo.

4.6. Resumen y consideraciones finales

En este capítulo se ha mostrado el diseño de una metodología específica para la detección y clasificación automática de movimientos durante la realización de actividades de juego a partir del uso de los Smart Toys diseñados en este mismo trabajo. Esta propuesta se deriva de la necesidad de ofrecer, más allá de los datos obtenidos por los sensores, herramientas que permitan ayudar a los expertos en desarrollo infantil a evaluar cómo se han llevado a cabo las actividades de la forma más automatizada posible.

Para determinar la propuesta, se ha llevado a cabo un estudio del estado del arte en la detección y clasificación automática de diferentes tipos de actividades humanas a partir de datos proporcionados por sensores. Las principales conclusiones de este estudio es que se utilizan mayoritariamente métodos derivados de la rama de la Inteligencia Artificial conocida como Machine Learning, y que las actividades que habitualmente se detectan son repetitivas y con una duración relativamente larga.

En el caso de las actividades de juego planteadas en esta Tesis, se tienen movimientos no excesivamente repetitivos y de corta duración, por lo que los planteamientos habituales no han sido adecuados para alcanzar los objetivos planteados. Por tanto, se ha ideado un sistema específico basado en las tendencias de aceleración medidas en los

Smart Toys, que permita obtener el patrón de aceleraciones que puede identificar cada uno de los posibles movimientos de forma unívoca, y se ha construido un sistema de clasificación en base a la comparación de estos patrones con las señales medidas en los movimientos.

Además, se ha planteado la utilización de un sistema de optimización basado en un Algoritmo Genético para la obtención de las variables óptimas a utilizar en el procesado de señales para cada uno de los movimientos.

La comparación de este sistema con el algoritmo de Machine Learning que mejor resultados de validación ha proporcionado durante su entrenamiento (una máquina de vectores de soporte) y con un sistema basado en el cálculo de distancias entre señales, se ha podido concluir que el sistema propuesto es mucho más fiable en cuanto al acierto y precisión de las detecciones.

Capítulo 5

Publicaciones relacionadas

Durante el trabajo de investigación que se ha llevado a cabo en esta Tesis, se han ido realizando también una serie de publicaciones científicas que han servido para evaluar y validar el trabajo realizado en cada uno de los objetivos por parte de revisores externos. Además, han permitido la difusión del trabajo en distintos ámbitos. A continuación se describen y referencian estos trabajos:

- "Smart Toys designed for detecting developmental delays" [250] es un artículo publicado en la revista *Sensors*. En él se describen partes de la plataforma y especialmente se describen los prototipos de Smart Cubes que se han implementado, así como el procesado de los datos de los sensores que se lleva a cabo en ellos.
- En la solicitud de patente (en trámite de aceptación) "Sistema de sondas inteligentes de monitorización aplicado a objetos de uso cotidiano" [251] se describe el diseño de los Smart Cubes propuestos.
- En la solicitud de patente (en trámite de aceptación) titulada "Sistema de monitorización de actividades con clavijas" [252] se describe el diseño del prototipo de la tabla de espigas propuesta.
- La tabla de espigas y su aplicación en la monitorización de la destreza manual ha sido objeto también de una presentación en la *18ª Conferencia europea en*

Psicología del Desarrollo (18th European Conference on Developmental Psyhology) con el título "Technology contributing to child studies: Toddlers' manual dexterity using an electronic pegboard" [253].

- Otra descripción de la plataforma y los Smart Toys diseñados se ha presentado en las *VI Jornadas de Jóvenes Investigadores* de la Universidad de Alcalá con el título "Desarrollo de juguetes inteligentes para detectar problemas de desarrollo en niños" y se ha publicado en forma de capítulo del libro de dichas jornadas [254].

- El estudio y diseño de los sistemas de comunicaciones utilizados en la plataforma de Smart Toys y los prototipos se basan en el trabajo realizado durante una estancia de investigación en la *Università Campus Bio-Medico di Roma (UCBM)*. Los resultados del trabajo en dicha estancia se han publicado con el título "A wearable system for real-time continuous monitoring of physical activity" en la revista *Journal of Heealthcare Engineering* [255].

- En el artículo "A Smart Toy to enhance the decision-making process at children psychomotor delay screenings: A pilot study" [256], publicado en la revista *Journal of Medical Internet Research (JMIR)* se describen las pruebas piloto realizadas y sus resultados.

- Los resultados del análisis de los datos de las pruebas piloto y la verificación del funcionamiento de los Smart Toys han sido presentados en forma de póster con el título "Verifying sensors in Smart Toys designed to help professionals in the early detection of psychomotor developmental disorders" [257] en el *5º Simposio Internacional en Ciencia de Sensores (5th International Symposium on Sensor Science, I3S)*.

- Un diseño preliminar del mecanismo propuesto para el control de acceso unificado se ha presentado de forma oral con el título "Aplying an unified Access Control for IoT-based agent systems" [258] en la *8º Conferencia Internacional en Computación orientada a servicios (IEEE 8th Service-Oriented Computing and Applications, SOCA)*.

- Se ha llevado a cabo también una presentación oral de la evolución del mecanismo de control de acceso y su aplicación en plataformas como la definida en este trabajo: "Protecting sensors in an IoT environment by modelling communicatiosn

as resources" [259] en el *5º Simposio Internacional en Ciencia de Sensores (5th International Symposium on Sensor Science, I3S)*

- El diseño ampliado y completo del mecanismo de control de acceso, así como las pruebas y los resultados de validación correspondientes, se ha publicado en la revista *Sensors*, con el título "Access Control mmechanism for IoT environments base on modelling communication procedures as resources" [260].

- El sistema de cifrado y autenticación de la plataforma de Smart Toys, junto con otros mecanismos de seguridad se han descrito en el artículo titulado "Secure communications and protected data for a Internet of Things Smart Toy platform" [261], que ha sido enviado a la revista *IEEE IoT Journal*, estando pendiente de aceptación a la finalización de este libro.

- El mecanismo propuesto para la detección y clasificación automática de movimientos se ha descrito en el artículo titulado "A novel method for automatic detection and classification of movement patterns in short duration playing activities" [262] que ha sido enviado a la revista *IEEE Access*, estando aceptado para su publicación a la finalización de este libro.

En total se han publicado cuatro artículos en revistas indexadas en el *Journal Citations report (JCR)*, tres de ellas ubicadas en el primer cuartil de sus respectivos ámbitos, un capítulo de libro, se han realizado dos solicitudes de patente españolas y se han llevado a cabo cuatro comunicaciones en congresos internacionales. Finalmente, al terminar este libro hay otros dos artículos relacionados con la Tesis enviados a revistas indexadas en JCR (también en el primer cuartil ambas) pendientes de revisión.

Capítulo 6

Discussion, conclusions and future work

In this final chapter we present a discussion of the results obtained in the Thesis, and we determine the main conclusions that can be extracted from them. Section 6.1 includes the discussion and conclusions for each one of the three main contributions in separated sections. Finally, we include in section 6.2 some of the possible future work lines that can be derived from the results of the Thesis.

6.1. Discussion and conclusions

The main goal of this Thesis was, as stated in the introduction chapter (chapter 1), the creation of new tools, methodologies and processes for the generation of valuable information for experts in children development. The work has relied heavily in the tools that the experts already use for the early assessment of delays or difficulties in motor, cognitive, language or sensory development. The results presented in each chapter of this book show that this goal has been reached by defining and developing a viable, secure and usable Smart Toys platform and by providing novel methodologies for the automatic detection and classification of the movements that compose an activity.

In the next sections, we discuss the results obtained in each chapter and the following specific conclusions of each contribution.

6.1.1. Definition of a Smart Toy IoT platform

The design of the architecture of the platform has allowed us to obtain useful tools for the experts to use in their evaluations, and also has provided us with a technological base in which we have been able to start developing the mechanisms and methodologies which would contribute to the expert's evaluation.

The designed platform, following a reference model, allows to be easily extended by defining new Smart Toys compatible with the architecture, and the Internet-based servers allow the creation of services and resources which can be easily used by client applications.

Although the Smart Toys platform has been developed and is ready to be used (as the tests with the prototype Smart Toys have stated), the implementation carried out during the Thesis can be seen as an experimental proof of concept. Further developments for the definition of services and platform management could still be added to it. The design of the architecture eases this task as it already defines how the platform should be implemented and extended. For the platform to be fully functional from a IoT point of view, it should be deployed using Cloud environments to ensure the scalability of the services designed for it.

The prototypes built have been designed following the requirements of the experts and tested in real-life scenarios, such as children schools, which gave us important feedback about the data obtained from the Smart Toys.

Using the results of the factor analysis carried out, we conclude that the possible variables measured from the sensors in the Smart Toys can be grouped in at least three categories: trembling, speed and accuracy. The group of variables which measure the shakings (trembling) could be seen *a priori* as a very useful source of information, it has not noticeable significance in the overall relationship between the measured data and the experts' manual performance assessment, despite it being one of the metrics

used by experts in the manual evaluation tests.

On the other hand, these results show that speed is negatively related to the overall performance. This means that lower speeds in movement result in better performances. This is an important result to consider when designing methodologies for the automatic generation of valuable information for the experts. Nevertheless, the variables that measure the accuracy of the movements are directly related with the overall performance as expected. All these measures should be automatically retrieved from each movement carried out during the activity, which reinforces the need of designing methods to automatically classify these movements.

The regression analysis was also performed to establish a relationship between the measured variables and the age. The results show, however, that only the accuracy related variables (lowest speeds and accelerations and the number of movements) are related to the age of the children. The older the child is, the highest accuracy is measured.

The limited number of samples which could be obtained during the pilot tests have restricted the analysis which could be carried out in this part of the Thesis. A higher number of tests and samples could lead to obtain more accurate relationships between the data and specific development issues.

Nevertheless, the results obtained are very promising, as they show there is a relationship between the variables which can be measured by sensors and the performance scores that the experts consider. These results are important for the design of future Smart Toy prototypes and for the definition of automatic information retrieving methodologies.

6.1.2. Security and privacy in a Smart Toy environment

The security methods and mechanisms which have been incorporated to the Smart Toys platform allow us to ensure that both the communications between close devices and the communications through Internet are carried out with a minimum risk of being intercepted and compromised. This has allowed us to reach the second main objective

of the Thesis. The important concerns about the security and privacy aspects of the platform had led to the definition of this separated objective which was intended to avoid the main vulnerabilities that could threaten the platform and its data.

We have relied the security on standard algorithms and methods as much as possible, to avoid introducing unknown possible security breaches. Nevertheless, we have proposed specific security mechanisms for the most vulnerable parts of the platform by providing confidentiality and authentication methods, and we have suggested a novel access control method which makes it easier to protect of the information generated in the system.

The proposed solution for authentication and confidentiality in the communications between the Smart Toys and the gateways tries to avoid the unauthorized access to the subsystems and the protection of the communication channels.

The study of the state of the art in authentication and confidentiality in IoT platforms shows that there are many proposals for providing this features to the communications of IoT devices. In many cases, the algorithms and mechanisms used depend on the protocols and the technologies of the devices. In our case, the Smart Toys do not use standard application protocols, neither they use a communication technology which enforces certain security mechanism (such as Bluetooth or 6LowPAN for instance). Taking this into account, we have determined the use of standard algorithms and methods (AES in CBC mode, MAC functions, etc.) but trying to produce the lowest overhead in terms of time and processing needs in the devices. To achieve that, we have proposed the division of the MAC in as much slices as communication channels we have. This solution takes advantage of the communications hardware selected for the prototype Smart Toys but could coexist with other communications systems used in different devices. Moreover, although the mechanism has been implemented and tested, using AES and SHA-3 as cryptographic algorithms it could be considered algorithm-agnostic, whilst it relies in a symmetric key scheme.

The sliced MAC method provides a lower degree of security than using a "non-slicing" approach, but it has been determined enough for the activities which will be carried out with the Smart Toys, as the keys will be used in just a few message exchanges during a short period of time, and discarded for the next activities. The high

number of possible keys in each case (as calculated in section 3.6.1) determines that it is secure enough to provide the adequate authentication and confidentiality range necessary for this kind of scenarios.

The results of the tests show that the computational time cost, although it is significant for a microcontroller of 8 bits and 16 MHz, it is low for the time requirements of the playing activities scenario, where there are not many messages sent per unit of time. Specifically, the results show that is possible to provide preprocessed data directly from the Smart Toys while using the authentication and confidentiality mechanisms, and that these mechanisms can also be used to provide raw data at a reasonable transmission rate.

It is important to note that although possible denegation of service attacks on the Smart Toys platform were identified, no specific methods have been proposed to mitigate them, because the risk of the attack is lower than the possible data loss (it would only affect a set of Smart Toys in a very brief period of time).

Regarding the access control proposal, it has allowed the platform to relay on a single mechanism for any resource which can be accessed within the system. The same type of policies, rules and management tasks can be used to control the access to any data in the platform by extending the UMA profile to the Smart Toys environment.

This extension has been carried out by modelling the communication channels of the Smart Toys and gateways as resources in the platform, and then applying the protection scheme to them in an analogous way to Web APIs.

The proposal, as it has been described in section 3.6.2, has been implemented and tested for its validation. The tests show that the processing time of the authorization and access control module represent just a small part of the overall time of the communication process (around 0.23 ms). These values depend on the distance and processing load of the authorization servers, which could present a scalability problem if there are many devices interacting with them. This drawback can be mitigated by using Internet-based server federation schemes which would allow the replication of authorization servers. The specific federation mechanism is out of the scope of this Thesis but could be part of a more detailed platform design proposed as future work

in section 6.2.

It is also important to note that there is an additional time which must be used when it is necessary to perform the validation of the access control permissions in each action (the introspection action). In a Cloud environment, this time values can be bounded around the 15 % of the total time used. It could be possible to obtain lower values by using the best possible configuration in this kind of environments (that is, placing the services in a cluster in a near geographic zone, enabling priority routes, etc.).

Additionally, the results show that, in the same scenario in terms of number of messages and conditions, the publishing/subscription communications scheme offers a lower energy consumption than REST communication schemes, which reinforces the idea of providing a unified access control scheme for both methods, as the publishing/-subscription scheme will be more suitable for the Smart Toys environment, and REST will be more suitable for the service APIs.

From the publishing/subscription perspective, the actions needed for the communication are exactly the same using the access control scheme or without not using it. Therefore there is an important advantage of the system because with this schema, there is not necessary any modification of the communication protocol.

The overhead in energy consumption is also bounded to the time needed for the introspection process (T_{P_Intros}), and could be reduced by putting the Smart Toy in a standby state when it is not performing activities.

After analyzing the results, we can say that the delays added to the system by using the access control scheme are acceptable and therefore the system is viable and provides enough advantages. One of these advantages is the minimum interaction required between the Smart Toys owner and the devices. The user only needs to know the RPT token to modify the access policies using a Web interface. Another advantage of this proposal, when compared with other existent solutions, are that it uses a generic approach, while most of the commercial solutions are platform-bounded or protocol-bounded. In general, the unified access control scheme can be applied to many scenarios and different IoT protocols, providing an easy-to-use interface to users which

makes it a very interesting contribution to the designed platform.

6.1.3. Movement detection and classification

The third contribution of this Thesis has been the definition of a novel methodology for the detection and classification of movements in playing activities in an autonomous way. This methodology was designed to accomplish the third objective defined in the Thesis. It aims to provide the experts with autonomous and automatic mechanisms which could be used by them in the assessment of children development.

The proposed methodology, based on the extraction of acceleration trends for each type of movement, and then in the comparison between these patterns and the input acceleration sequences from the playing activities, has been implemented and tested.

The results of this methodology show that it has been able to detect successfully different movements performed with the Smart Toys, demonstrating a high accuracy both in the detection of the time instants for the beginning and ending of the movement and in the classification of the type of movement. The high accuracy has been proven when he system has been used in isolated signals of each movement and when the input signals contained sequences of different movements.

The methodology has been compared with other widely used detection and classification systems, one based on the calculation of Euclidean distances between signals and the other based on a well-known Machine Learning algorithm: The Support Vector Machines. Although these are widely used mechanisms, and the SVM model reached a high accuracy (above the 95 %) during the training process, our proposal has offered better results. Our proposal was the only method which has been able to correctly classify the 100 % of the movements performed during the tests. In some specific cases the distances between the detected movements and the real movements are lower in the other compared mechanisms, but in general the accuracy of the proposed method is higher.

There are some specific problems which affect The other two methods considered in this comparison and that are mitigated by our proposal. For instance, the Signal

Similarity Search method presents important disadvantages because it is not able to determine how many movements it should search in a sequence of accelerations. This means that the number of detected signals can be overestimated or underestimated, as it must rely on *a priori* parameters. An estimation of the largest distance which should be considered for the signal search could be determined using a training process, but it still would be a value too dependent on the training data and will not be useful for new sequences of movements.

The Machine Learning based methods share another important disadvantage in our scenario, which has been proven with the use of the Support Vector Machine model: these methods are accurate in the classification of movements if the training set and the extracted features are good enough, but the detection of short duration non-repetitive movements in signals composed by an arbitrary number of movements is not something that can achieve correctly by themselves. In our case we had to add an extra detection algorithm so just the isolated signals are passed to the classification method, and the final accuracy would depend on the accuracy of the detection algorithm.

The results for the three selected movements show that, in all cases, there are movements which, being more complex, could be more difficult to classify. This can be seen specially in the stacking movement which, in the two other compared methods are sometimes classified as a pair of up and down movements. In our case, however, the possibility of finding specific different patterns for each movement makes it easier for our proposal to correctly classify them.

The proposed methodology can also be used in "real time", without needing the full acceleration signal available in each step of the algorithm. This is also an important advantage to take into account in the Smart Toys platform, where it could be used for giving information about the movements to the experts while the activity is being performed.

The optimization algorithm used by our methodology to optimize the specific variables to be used for each type of movement has shown that it is possible to obtain specific detectors for each possible movement performed while obtaining the most repeated pattern for each one.

In general, the experiments carried out and the obtained results suggest that the methodology is sound and can be used for the generation of useful data in the Smart Toys platform, by providing the experts with an automatic classification of the movements performed. This could be combined with the other information received from toys (about speeds, accuracy of the movements, or trembling, as shown in chapter 2).

6.2. Lines of future work

The good results obtained in this Thesis in areas such as the automatic detection and classification of movements can be considered just the starting point for a wider research line which will be explored in future works.

As we already stated in section 6.1.1, the designed platform has been developed as an experimental prototype version, and therefore, it should be expanded by defining services, activities and new Smart Toys. Following this idea, the Smart Toy designs which have not been included in the first iteration of the Thesis (puzzle, ball, rattle, etc.) should be finished and the prototypes should be tested in a similar fashion to what is presented in this work.

Building a high number of sets of Smart Toys in a near future could also open interesting research lines. These toys could be used in different scenarios like homes or schools and would provide an enough data base to offer a more expense validation of the toys, by determining new relationships between the data and the children development. This high quantity of data could also be analyzed, and new services could be built based on the analysis.

The final goal of such research would be the generation of alerts and other similar services which could inform and advise parents, teachers and experts about the development of children.

There can be also identified lines of future work in the security issues addressed in the second contribution of this Thesis. For instance, the proposed authentication and confidentiality mechanism can be extended to new communications devices, ensuring every Smart Toy in the platform can apply it. The access control mechanism, on

the other hand has been tested over the MQTT protocol. Although the use of other protocols such as CoAP or AMQP should be relatively easy, they can be implemented to provide more options for the communications in the Smart Toys platform, while proving the protocol-agnostic nature of the proposal.

Another interesting future work line would be the exploration of possible specific access policies which could be defined for this environment, and how they should be defined in the platform to ensure the ease of use while providing the higher possible granularity in the access control.

Finally, the promising results obtained in the automatic detection and classification of movements has lead us to define future work lines. For instance, it would be important to extend the methodology, so it can benefit from the use of different types of sensors such as gyroscopes, light sensors, etc., which would lead to the detection of more complex movements in different activities. The acceleration patterns can also be made more complex, for instance defining more than three possible trend states, which could lead the system to be able to detect and classify more subtle and complex movements. A real time implementation of the system could also be useful to test it while the activities are being performed.

Bibliografía

[1] F. P. Hughes, *Children, play, and development.* Sage, 2009.

[2] I. P. Samuelsson and M. A. Carlsson, "The playing learning child: Towards a pedagogy of early childhood," *Scandinavian journal of educational research*, vol. 52, no. 6, pp. 623–641, 2008.

[3] G. Guyton, "Using toys to support infant-toddler learning and development," *YC Young Children*, vol. 66, no. 5, p. 50, 2011.

[4] H. Akbari, B. Abdoli, M. Shafizadeh, H. Khalaji, S. Hajihosseini, and V. Ziaee, "The effect of traditional games in fundamental motor skill development in 7-9 year-old boys," *Iranian Journal of Pediatrics*, vol. 19, no. 2, pp. 123–129, 2009.

[5] A. C. Bundy, S. Shia, Q. Long, and L. J. Miller, "How does sensory processing dysfunction affect play?" *The American Journal of Occupational Therapy*, vol. 61, no. 2, p. 201, 2007.

[6] W. Cools, K. De Martelaer, C. Samaey, and C. Andries, "Movement skill assessment of typically developing preschool children: A review of seven movement skill assessment tools," *Journal of sports science & medicine*, vol. 8, no. 2, p. 154, 2009.

[7] R. M. Folio and F. R. R., *PDMS-2: Peabody Developmental Motor Scales (2nd Edition).* Pro-ed, Austin Texas, 2000.

[8] K. Kuhlthau, F. Orlich, T. A. Hall, D. Sikora, E. A. Kovacs, J. Delahaye, and T. E. Clemons, "Health-related quality of life in children with autism spectrum disorders: Results from the autism treatment network," *Journal of autism and developmental disorders*, vol. 40, no. 6, pp. 721–729, 2010.

[9] S. I. Greenspan and S. Wieder, "Developmental patterns and outcomes in infants and children with disorders in relating and communicating: A chart review of 200 cases of children with autistic spectrum diagnoses," *Journal of Developmental and Learning disorders*, vol. 1, pp. 87–142, 1997.

[10] C. Assaiante, S. Mallau, S. Viel, M. Jover, and C. Schmitz, "Development of postural control in healthy children: a functional approach," *Neural plasticity*, vol. 12, no. 2-3, pp. 109–118, 2005.

[11] Federación Estatal de Asociaciones de Profesionales de Atención Temprana (GAT), *Libro blanco de la atención temprana.* Real Patronato sobre Discapacidad, 2005.

[12] B. Sawyer, "From cells to cell processors: the integration of health and video games," *IEEE Computer Graphics and Applications*, vol. 28, no. 6, 2008.

[13] N. Bayley and G. Reuner, *Bayley scales of infant and toddler development: Bayley-III.* Harcourt Assessment, Psych. Corporation San Antonio, Tex, USA, 2006, vol. 7.

[14] G. H. Roid and J. L. Sampers, *Merrill-Palmer-revised: scales of development.* Stoelting, 2004.

[15] World Health Organization, "World report on disability," *World Health Organization*, vol. 15, 2013.

[16] C. S. Parshuram, J. Hutchison, and K. Middaugh, "Development and initial validation of the bedside paediatric early warning system score," *Critical care*, vol. 13, no. 4, p. R135, 2009.

[17] Bright Futures Steering Committee and Medical Home Initiatives for Children With Special Needs Project Advisory Committee, "Identifying infants and young children with developmental disorders in the medical home: An algorithm for developmental surveillance and screening," *Pediatrics*, vol. 118, no. 1, pp. 405–420, 2006, pmid:16818591.

[18] L. Da Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[19] K. Ashton *et al.*, "That 'internet of things' thing," *RFID journal*, vol. 22, no. 7, pp. 97–114, 2009.

[20] D. V. Dimitrov, "Medical internet of things and big data in healthcare," *Healthcare informatics research*, vol. 22, no. 3, pp. 156–163, 2016.

[21] Y. Yuehong, Y. Zeng, X. Chen, and Y. Fan, "The internet of things in healthcare: An overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3–13, 2016.

[22] M. Jiang, H. Shang, Z. Wang, H. Li, and Y. Wang, "A method to deal with installation errors of wearable accelerometers for human activity recognition," *Physiological measurement*, vol. 32, no. 3, p. 347, 2011.

[23] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[24] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *IEEE Wireless communications*, vol. 17, no. 6, 2010.

[25] G. Agosta, L. Borghese, C. Brandolese, F. Clasadonte, W. Fornaciari, F. Garzotto, M. Gelsomini, M. Grotto, C. Fr, and D. Noferi, "Playful supervised smart spaces (p3s)–a framework for designing, implementing and deploying multisensory play experiences for children with special needs," in *Digital System Design (DSD), 2015 Euromicro Conference on.* IEEE, 2015, pp. 158–164.

[26] E. K. M. Lau and S. C. S. YU, "Education system using connected toys," Patent US20 170 053 550A1, 2017.

[27] E. McReynolds, S. Hubbard, T. Lau, A. Saraf, M. Cakmak, and F. Roesner, "Toys that listen: A study of parents, children, and internet-connected toys," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* ACM, 2017, pp. 5197–5207.

[28] J. K. Tang and P. C. Hung, *Computing in Smart Toys.* Springer, 2017.

[29] H. D'Hooge and M. Goldstein, "History of the smart toy lab and intel play toys," *Intel Technology Journal*, vol. 2001, no. Q4, 2001.

[30] J. A. Marsh, "The internet of toys: A posthuman and multimodal analysis of connected play," *Teachers College record (1970)*, vol. 119, no. 15, 2017.

[31] "Ecosistema de detección ubicua, atención y estimulación temprana para niños con trastornos del desarrollo (educere)," accessed 10/08/2018. [Online]. Available: https://educeremus.wordpress.com/

[32] M. L. Martín Ruiz, M. Á. Valero Duboy, M. Linden, S. Nunez Nagy, and Á. Gutiérrez García, "Foundations of a smart toy development for the early detection of motoric impairments at childhood," *International Journal of Pediatric Research*, vol. 1, no. 2, pp. 1–5, 2015.

[33] P. P. Ray, "A survey on internet of things architectures," *Journal of King Saud University-Computer and Information Sciences*, 2016.

[34] P. Sethi and S. R. Sarangi, "Internet of things: architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, 2017.

[35] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, "Research on the architecture of internet of things," in *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, vol. 5. IEEE, 2010, pp. V5–484.

[36] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *Frontiers of Information Technology (FIT), 2012 10th International Conference on*. IEEE, 2012, pp. 257–260.

[37] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[38] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[39] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*. IEEE, 2014, pp. 1–8.

[40] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of things: Integrating internet of things and cloud computing and the issues involved," in *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on*. IEEE, 2014, pp. 414–419.

[41] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[42] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[43] C. Sun, "Application of rfid technology for logistics on internet of things," *AASRI Procedia*, vol. 1, pp. 106–111, 2012.

[44] Y. Wang and K. Cao, "A proactive complex event processing method for large-scale transportation internet of things," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, p. 159052, 2014.

[45] M. Soliman, T. Abiodun, T. Hamouda, J. Zhou, and C.-H. Lung, "Smart home: Integrating internet of things with web services and cloud computing," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2013, pp. 317–320.

[46] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[47] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[48] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of internet-of-things platforms," *Computer Communications*, vol. 89, pp. 5–16, 2016.

[49] S. Krco, B. Pokric, and F. Carrez, "Designing iot architecture (s): A european perspective," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 79–84.

[50] "Future internet core platform (fi-ware)," accessed 23/07/2018. [Online]. Available: https://www.fiware.org

[51] F. Ramparany, F. G. Marquez, J. Soriano, and T. Elsaleh, "Handling smart environment devices, data and services at the semantic level with the fi-ware core platform," in *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE, 2014, pp. 14–20.

[52] M. Presser, P. M. Barnaghi, M. Eurich, and C. Villalonga, "The sensei project: Integrating the physical world with the digital world of the network of the future," *IEEE Communications Magazine*, vol. 47, no. 4, pp. 1–4, 2009.

[53] "Semantic-service provisioning for the internet of things using future internet research by experimentation (spitfire)," accessed 23/07/2018. [Online]. Available: https://cordis.europa.eu/project/rcn/94963_es.html

[54] "Internet of things architecture (iot-a) fp7 project," accessed 23/07/2018. [Online]. Available: https://cordis.europa.eu/project/rcn/95713_es.html

[55] "onem2m: Standards for m2m and the internet of things," accessed 7/08/2018. [Online]. Available: http://www.onem2m.org

[56] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common m2m service layer platform: Introduction to onem2m," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[57] D. Holloway and L. Green, "The internet of toys," *Communication Research and Practice*, vol. 2, no. 4, pp. 506–519, 2016.

[58] V. Dureau, "Networking smart toys," Patent US20 050 111 823A1, 2005.

[59] C. G. Rivas, "System for coordinating behavior of a toy with play of an online educational game," Patent US20 100 093 434A1, 2010.

[60] S. Schmid, M. Gorlatova, D. Giustiniano, V. Vukadinovic, and S. Mangold, "Networking smart toys with wireless toybridge and toytalk," in *Infocom*, 2011.

[61] P. Ihamaki, S. Pori, and K. Heljakka, "Smart, skilled and connected in the 21th century: Educational promises of the internet of toys (iotoys)," in *2018 Hawaii University International Conferences in Arts, Humanities, Social Sciences & Education*, 2018.

[62] K. Cagiltay, N. Kara, and C. C. Aydin, *Smart toy based learning*, ser. Handbook of research on educational communications and technology. Springer, 2014, pp. 703–711.

[63] L. Plowman and R. Luckin, "Interactivity, interfaces, and smart toys," *Computer*, vol. 37, no. 2, pp. 98–100, 2004.

[64] G. Roberts-Holmes, "Playful and creative ict pedagogical framing: a nursery school case study," *Early Child Development and Care*, vol. 184, no. 1, pp. 1–14, 2014.

[65] W.-N. Wang, V. Kuo, C.-T. King, and C.-P. Chang, "Internet of toys: an e-pet overview and proposed innovative social toy service platform," in *Computer Symposium (ICS), 2010 International*. IEEE, 2010, pp. 264–269.

[66] L. Berglin, "Spookies: Combining smart materials and information technology in an interactive toy," in *Proceedings of the 2005 conference on Interaction design and children*. ACM, 2005, pp. 17–23.

[67] N. Kara, C. C. Aydin, and K. Cagiltay, "Design and development of a smart storytelling toy," *Interactive Learning Environments*, vol. 22, no. 3, pp. 288–297, 2014.

[68] H. Guo, H. Trotteberg, A. I. Wang, and M. Zhu, "Temps: A conceptual framework for pervasive and social games," in *Digital Game and Intelligent Toy Enhanced Learning (DIGITEL), 2010 Third IEEE International Conference on*. IEEE, 2010, pp. 31–37.

[69] M. C. Domingo, "An overview of the internet of things for people with disabilities," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 584–596, 2012.

[70] M. Vega-Barbas, I. Pau, J. Ferreira, E. Lebis, and F. Seoane, "Utilizing smart textiles-enabled sensorized toy and playful interactions for assessment of psychomotor development on children," *Journal of Sensors*, vol. 2015, 2015.

[71] F. Taffoni, V. Focaroli, F. Keller, and J. M. Iverson, "A technological approach to studying motor planning ability in children at high risk for asd," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 3638–3641.

[72] S. Mironcika, A. de Schipper, A. Brons, H. Toussaint, B. Krose, and B. Schouten, "Smart toys design opportunities for measuring childrenś fine motor skills development," in *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 2018, pp. 349–356.

[73] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments," in *Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, 2001, pp. 132–138.

[74] G. Corbellini, K. Aksit, S. Schmid, S. Mangold, and T. Gross, "Connecting networks of toys and smartphones with visible light communication," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 72–78, 2014.

[75] I. Glaropoulos, S. Mangold, and V. Vukadinovic, "Enhanced ieee 802.11 power saving for multi-hop toy-to-toy communication," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. IEEE, 2013, pp. 603–610.

[76] M. Konkel, V. Leung, B. Ullmer, and C. Hu, "Tagaboo: a collaborative children's game based upon wearable rfid technology," *Personal and Ubiquitous Computing*, vol. 8, no. 5, pp. 382–384, 2004.

[77] F. Albinali, M. S. Goodwin, and S. Intille, "Detecting stereotypical motor movements in the classroom using accelerometry and pattern recognition algorithms," *Pervasive and Mobile Computing*, vol. 8, no. 1, pp. 103–114, 2012.

[78] D. P. Cliff, J. J. Reilly, and A. D. Okely, "Methodological considerations in using accelerometers to assess habitual physical activity in children aged 0–5 years," *Journal of Science and Medicine in Sport*, vol. 12, no. 5, pp. 557–567, 2009, pmid:19147404.

[79] C. Verplaetse, "Inertial proproceptive devices: Self-motion-sensing toys and tools," *IBM Systems Journal*, vol. 35, no. 3.4, pp. 639–650, 1996.

[80] F. Cordella, F. Taffoni, L. Raiano, G. Carpino, M. Pantoni, L. Zollo, E. Schena, E. Guglielmelli, and D. Formica, "Design and development of a sensorized cylindrical object for grasping assessment," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*. IEEE, 2016, pp. 3366–3369.

[81] Y. Satoh, S. Tanaka, and Y. Katoh, "Interactive toy figure with sound-activated and pressure-activated switches," Patent US5 324 225A, 1994.

[82] E. Taylor and K. Michael, "Smart toys that are the stuff of nightmares," *IEEE Technology and Society Magazine*, vol. 35, no. 1, pp. 8–10, 2016.

[83] K. Watry, "Interactive cloud-based toy," 2017, patent number US9833725B2.

[84] K. Michael and A. Hayes, "High-tech child's play in the cloud: Be safe and aware of the difference between virtual and real," *IEEE Consumer Electronics Magazine*, vol. 5, no. 1, pp. 123–128, 2016.

[85] A. Bassi, M. Bauer, M. Fiedler, and R. v. Kranenburg, *Enabling things to talk*. Springer, 2013.

[86] N. Rozanski and E. Woods, *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley, 2012.

[87] S. Meyer, A. Ruppen, and L. Hilty, "The things of the internet of things in bpmn," in *International Conference on Advanced Information Systems Engineering*. Springer, 2015, pp. 285–297.

[88] O. Rejeb, R. Bastide, E. Lamine, F. Marmier, and H. Pingaud, "A model driven engineering approach for business continuity management in e-health systems," in *Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on*. IEEE, 2012, pp. 1–7.

[89] "The third network: Lifecycle service orchestration vision," MEF: Metro Ethernet Forum, Tech. Rep., 2015.

[90] *ATMega328/P 8-bit AVR Microcontrollers*, ATMEL, 11 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[91] *MCP111/112*, Microchip, 2016. [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/20001889F.pdf

[92] *nRF24L01+: Single chip 2.4 GHz transceiver*, Nordic Semiconductor, 9 2008, v 1.0. [Online]. Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf

[93] Bluetooth SIG, "Bluetooth core specification v5. 0," *Bluetooth SIG: San Jose, CA, USA*, 2010.

[94] M. Gast, *802.11 wireless networks: the definitive guide*. O'Reilly Media, Inc., 2005.

[95] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of ipv6 packets over ieee 802.15. 4 networks," Tech. Rep., 2007.

[96] "Zigbee document 053474r13," *ZgBee Standards Organization*, 2006.

[97] *MPU-9150 Product Specification Revision 4.3*, Invensense Inc., 9 2013. [Online]. Available: https://www.invensense.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf

[98] *NORPS-12 CdS photocell*, Silonex. [Online]. Available: https://docs-emea.rs-online.com/webdocs/0034/0900766b8003484c.pdf

[99] *MPU-6000 and MPU-6050 Product Specification Revision 3.4*, Invensense Inc., 9 2013. [Online]. Available: https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

[100] J. Tiffin and E. J. Asher, "The purdue pegboard: norms and studies of reliability and validity." *Journal of applied psychology*, vol. 32, no. 3, p. 234, 1948.

[101] J. Desrosiers, R. Hebert, G. Bravo, and E. Dutil, "The purdue pegboard test: normative data for people aged 60 and over," *Disability and rehabilitation*, vol. 17, no. 5, pp. 217–224, 1995.

[102] L. D. Costa, H. G. Vaughan Jr, E. Levita, and N. Farber, "Purdue pegboard as a predictor of the presence and laterality of cerebral lesions." *Journal of consulting psychology*, vol. 27, no. 2, p. 133, 1963.

[103] *Photointerrupter LTH301-32*, Lite-On Inc., 16 2011. [Online]. Available: https://optoelectronics.liteon.com/upload/download/DS-55-96-0005/LTH-301-32DS.pdf

[104] *CD4021B CMOS 8-Stage Static Shift Register*, Texas Instruments, 03 2010. [Online]. Available: http://www.ti.com/lit/ds/symlink/cd4021b-q1.pdf

[105] *Raspberry Pi 3 model B*, RS Components. [Online]. Available: https://docs-europe.electrocomponents.com/webdocs/14ba/0900766b814ba5fd.pdf

[106] "Rabbitmq," accessed 18/08/2018. [Online]. Available: https://www.rabbitmq.com/

[107] "Mosquitto mqtt broker," http://mosquitto.org/, accessed: 2018-02-25.

[108] "Django, the web framework for perfectionists with deadlines," accessed 18/08/2018. [Online]. Available: https://www.djangoproject.com/

[109] "Apache cordova," accessed 18/08/2018. [Online]. Available: https://cordova.apache.org/

[110] I. F. Mondragón, P. Campoy, C. Martinez, and M. A. Olivares-Méndez, "3d pose estimation based on planar object tracking for uavs control," in *Robotics and automation (ICRA), 2010 IEEE international conference on*. Ieee, 2010, pp. 35–41.

[111] "Software ibm spss," accessed 18/08/2018. [Online]. Available: https://www.ibm.com/analytics/es/es/technology/spss/

[112] "Hp study reveals 70 percent of internet of things devices vulnerable to attack," July 2014, accessed 7 Feb 2017. [Online]. Available: http://www8.hp.com/us/en/hp-news/press-release.html?id=1744676

[113] A. Peterson, ""internet of things" compounded friday's hack of major websites," *The Washington Post*, October 2016.

[114] "U.s. internet disrupted as firm hit by cyberattacks," *CBS News*, October 2016, accessed 7/2/2017. [Online]. Available: http://www.cbsnews.com/news/internet-disrupted-dyn-hit-by-ddos-cyberattack/

[115] "Distributed denial of service attack against domain name service host highlights vulnerability of "internet of things" devices," *Federal Bureau of Investigation Cyber Bulletin*, October 2016, pin number=161026-001.

[116] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[117] G. Mascheroni and D. Holloway, Eds., *The Internet of Toys: A report on media and social discourses around young children and IoToys.* DigiLitEY, 2017.

[118] R. H. Weber, "Internet of things–new security and privacy challenges," *Computer law & security review*, vol. 26, no. 1, pp. 23–30, 2010.

[119] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of iot systems: Design challenges and opportunities," in *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design.* IEEE Press, 2014, pp. 417–423.

[120] Q. Gou, L. Yan, Y. Liu, and Y. Li, "Construction and strategies in iot security system," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing.* IEEE, 2013, pp. 1129–1132.

[121] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[122] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[123] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "Ieee 802.15. 4: a developing standard for low-power low-cost wireless personal area networks," *IEEE network*, vol. 15, no. 5, pp. 12–19, 2001.

[124] D. Culler and S. Chakrabarti, "6lowpan: Incorporating ieee 802.15. 4 into the ip architecture," *IPSO Alliance, White paper*, 2009.

[125] T. Winter, "Rpl: Ipv6 routing protocol for low-power and lossy networks," *Internet Engineering Task Force (IETF)*, 2012.

[126] A. J. Jara, P. Martinez-Julia, and A. Skarmeta, "Light-weight multicast dns and dns-sd (lmdns-sd): Ipv6-based resource and service discovery for the web of things," in *innovative mobile and internet services in ubiquitous computing (IMIS), 2012 sixth international conference on.* IEEE, 2012, pp. 731–738.

[127] R. Gupta and A. Banks, "Mqtt version 3.1.1 protocol specification," OASIS," OASIS Standard, 10 2014. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[128] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing*, vol. 10, no. 6, 2006.

[129] G. Pardo-Castellote and D. Chairman, "Omg data distribution service: Real-time publish/subscribe becomes a standard," *RTC Magazine*, vol. 14, 2005.

[130] M. Laine, "Restful web services for the internet of things," *[Online].Saatavilla*, 2012.

[131] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," *Internet Engineering Task Force (IETF)*, 2014.

[132] M. Kirsche and R. Klauck, "Unify to bridge gaps: Bringing xmpp into the internet of things," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*. IEEE, 2012, pp. 455–458.

[133] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "Dtls based security and two-way authentication for the internet of things," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2710–2723, 2013.

[134] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[135] "Cc2420 2.4 ghz ieee 802.15.4 / zigbee-ready rf transceiver," Texas Instruments, accessed 21/08/2018. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf

[136] C. Hennebert and J. Dos Santos, "Security protocols and privacy issues into 6lowpan stack: A synthesis," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 384–398, 2014.

[137] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. P. Vasseur, and R. Alexander, "Rpl: Ipv6 routing protocol for low-power and lossy networks," Tech. Rep., 2012.

[138] R. Gupta and A. Banks, "Mqtt version 3.1.1 protocol specification," OASIS," OASIS Standard, 10 2014. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.pdf

[139] P. Saint-Andre, "Extensible messaging and presence protocol (xmpp): Core," The Internet Engineering Task Force, Tech. Rep., 03 2011.

[140] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the internet of things," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 147–159, 2011.

[141] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 2, pp. 228–258, 2005.

[142] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 1, pp. 41–77, 2005.

[143] H. Pranata, R. Athauda, and G. Skinner, "Securing and governing access in ad-hoc networks of internet of things," in *Proceedings of the IASTED International Conference on Engineering and Applied Science, EAS*, 2012, pp. 84–90.

[144] N. Hong, "A security framework for the internet of things based on public key infrastructure," in *Advanced Materials Research*, vol. 671. Trans Tech Publ, 2013, pp. 3223–3226.

[145] J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, "A lightweight authentication protocol for internet of things," in *Next-Generation Electronics (ISNE), 2014 International Symposium on*. IEEE, 2014, pp. 1–2.

[146] T. Lackorzynski and S. Koepsell, "'hello barbie'-hacker toys in a world of linked devices," in *Broadband Coverage in Germany; 11. ITG-Symposium; Proceedings of*. VDE, 2017, pp. 1–7.

[147] V. Steeves, "It's not child's play: The online invasion of children's privacy," *University of Ottawa Law and Technology Journal*, vol. 3, no. 1, pp. 169–188, 2006.

[148] B. Nelson, "Children's connected toys: Data security and privacy concerns," *Office of Oversight and Investigations Minority Staff Report, US Senate Committee on Commerce, Science, and Transportation*, 2016.

[149] B. Yankson, F. Iqbal, and P. C. Hung, *Privacy Preservation Framework for Smart Connected Toys*, ser. Computing in Smart Toys. Springer, 2017, pp. 149–164.

[150] L. Rafferty, P. C. Hung, M. Fantinato, S. M. Peres, F. Iqbal, S.-Y. Kuo, and S.-C. Huang, *Towards a Privacy Rule Conceptual Model for Smart Toys*, ser. Computing in Smart Toys. Springer, 2017, pp. 85–102.

[151] L. G. de Carvalho and M. M. Eler, "Security requirements for smart toys," in *19th International Conference on Enterprise Information Systems (ICEIS)*, Porto, Portugal, 2017.

[152] G. Chu, N. Apthorpe, and N. Feamster, "Security and privacy analyses of internet of things toys," *arXiv preprint arXiv:1805.02751*, 2018.

[153] J. Valente and A. A. Cardenas, "Security & privacy in smart toys," in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*. ACM, 2017, pp. 19–24.

[154] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A. Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Networks*, vol. 112, pp. 237 – 262, 2017.

[155] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[156] G. Zhang and J. Tian, "An extended role based access control model for the internet of things," in *Information Networking and Automation (ICINA), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. V1–319.

[157] E. Barka, S. S. Mathew, and Y. Atif, "Securing the web of things with role-based access control," in *International Conference on Codes, Cryptology, and Information Security*. Springer, 2015, pp. 14–26.

[158] E. Yuan and J. Tong, "Attributed based access control (abac) for web services," in *Web Services, 2005. ICWS 2005. Proceedings. 2005 IEEE International Conference on*. IEEE, 2005.

[159] N. Ye, Y. Zhu, R.-c. Wang, R. Malekian, and L. Qiao-min, "An efficient authentication and access control scheme for perception layer of internet of things," *Applied Mathematics & Information Sciences*, vol. 8, no. 4, p. 1617, 2014.

[160] M. Hemdi and R. Deters, "Using rest based protocol to enable abac within iot systems," in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*. IEEE, 2016, pp. 1–7.

[161] X. Zhang, F. Parisi-Presicce, R. Sandhu, and J. Park, "Formal model and policy specification of usage control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 8, no. 4, pp. 351–387, 2005.

[162] P. N. Mahalle, B. Anggorojati, N. R. Prasad, R. Prasad, *et al.*, "Identity authentication and capability based access control (iacac) for the internet of things," *Journal of Cyber Security and Mobility*, vol. 1, no. 4, pp. 309–348, 2013.

[163] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1189–1205, 2013.

[164] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control," in *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on.* IEEE, 2003, pp. 120–131.

[165] A. Anderson, A. Nadalin, B. Parducci, D. Engovatov, H. Lockhart, M. Kudo, P. Humenn, S. Godik, S. Anderson, S. Crocker, *et al.*, "extensible access control markup language (xacml) version 1.0," *OASIS*, 2003.

[166] E. Hammer-Lahav, "The oauth 1.0 protocol," Internet Requests for Comments, IETF, RFC, 2010. [Online]. Available: https://tools.ietf.org/html/rfc5849

[167] E. Maler, D. Catalano, M. Machulak, and T. Hardjono, "User-managed access (uma) profile of oauth 2.0," *Kantara Initiative*, 2016.

[168] L. Seitz, G. Selander, and C. Gehrmann, "Authorization framework for the internet-of-things," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a.* IEEE, 2013, pp. 1–6.

[169] J. E. Kim, G. Boulos, J. Yackovich, T. Barth, C. Beckel, and D. Mosse, "Seamless integration of heterogeneous devices and access control in smart homes," in *Intelligent Environments (IE), 2012 8th International Conference on.* IEEE, 2012, pp. 206–213.

[170] D. Hardt, "The oauth 2.0 authorization framework," Internet Requests for Comments, IETF, RFC, 2012. [Online]. Available: https://tools.ietf.org/html/rfc6749

[171] S. Emerson, Y.-K. Choi, D.-Y. Hwang, K.-S. Kim, and K.-H. Kim, "An oauth based authentication mechanism for iot networks," in *Information and Communication Technology Convergence (ICTC), 2015 International Conference on.* IEEE, 2015, pp. 1072–1074.

[172] L. Seitz, G. Selander, E. Wahlstroem, S. Erdtman, and H. Tschofenig, "Authorization for the internet of things using oauth 2.0," *draft-ietf-ace-oauth-authz-00*, 2015.

[173] P. Fremantle, B. Aziz, J. Kopeckỳ, and P. Scott, "Federated identity and access management for the internet of things," in *Secure Internet of Things (SIoT), 2014 International Workshop on.* IEEE, 2014, pp. 10–17.

[174] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios," *IEEE sensors journal*, vol. 15, no. 2, pp. 1224–1234, 2015.

[175] D. Garcia-Carrillo, R. Marin-Lopez, A. Kandasamy, and A. Pelov, "A coap-based network access authentication service for low-power wide area networks: Lo-coap-eap," *Sensors*, vol. 17, no. 11, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/11/2646

[176] A. Celesti, M. Fazio, and M. Villari, "Enabling secure xmpp communications in federated iot clouds through xep 0027 and saml/sasl sso," *Sensors*, vol. 17, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/2/301

[177] A. Ouaddah, A. A. Elkalam, and A. A. Ouahman, "Towards a novel privacy-preserving access control model based on blockchain technology in iot," in *Europe and MENA Cooperation Advances in Information and Communication Technologies*, Á. Rocha, M. Serrhini, and C. Felgueiras, Eds. Cham: Springer International Publishing, 2017, pp. 523–533.

[178] M. J. Dworkin, "Sha-3 standard: Permutation-based hash and extendable-output functions," Tech. Rep., 2015.

[179] J. Kelsey, S.-j. Chang, and R. Perlner, "Sha-3 derived functions," *NIST Special Publication*, vol. 800, p. 185, 2016.

[180] "Mqtt version 3.1.1 becomes an oasis standard," https://www.oasis-open.org/news/announcements/mqtt-version-3-1-1-becomes-an-oasis-standard, 2014, accessed: 2018-02-25.

[181] "Kantara initiative," https://kantarainitiative.org, accessed: 2018-02-25.

[182] T. Hardjono, E. Maler, M. Machulak, and D. Catalano, "User-managed access (uma) profile of oauth 2.0," *Internet Engineering Task Force (IETF)*, 2015.

[183] ——, "User-managed access (uma) profile of oauth 2.0," Kantara Initiative," Recommendation, 04 2014.

[184] J. Richer, "Oauth 2.0 token introspection," Internet Engineering Task Force (IETF)," RFC, 05 2015.

[185] "Pycrypto - the python cryptography toolkit," accessed 13 Feb 2017. [Online]. Available: https://www.dlitz.net/software/pycrypto/

[186] "Arduinolabs - cryptographic library," accessed 13 Feb 2017. [Online]. Available: https://rweather.github.io/arduinolibs/crypto.html

[187] "Nodemcu v1.0 board," https://github.com/nodemcu/nodemcu-devkit-v1.0, accessed: 2018-02-25.

[188] R. A. Light, "Mosquitto: server and client implementation of the mqtt protocol," *Journal of Open Source Software*, vol. 2, no. 13, 2017.

[189] M. Bachry, "Mosquitto pyauth plugin," https://github.com/mbachry/mosquitto_pyauth, accessed: 2018-02-25.

[190] "Ina219 zerø-drift, bidirectional current/power monitor with i2c interface," http://www.ti.com/lit/ds/symlink/ina219.pdf, accessed: 2018-02-25.

[191] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors." *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1192–1209, 2013.

[192] P. Kumari, L. Mathew, and P. Syal, "Increasing trend of wearables and multimodal interface for human activity monitoring: A review," *Biosensors and Bioelectronics*, vol. 90, pp. 298–307, 2017.

[193] I. Bisio, A. Delfino, F. Lavagetto, and A. Sciarrone, "Enabling iot for in-home rehabilitation: Accelerometer signals classification methods for activity and movement recognition," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 135–146, 2017.

[194] J. R. Smith, K. P. Fishkin, B. Jiang, A. Mamishev, M. Philipose, A. D. Rea, S. Roy, and K. Sundara-Rajan, "Rfid-based techniques for human-activity detection," *Communications of the ACM*, vol. 48, no. 9, pp. 39–44, 2005.

[195] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Activity recognition using wearable sensors for elder care," in *Future Generation Communication and Networking, 2008. FGCN'08. Second International Conference on*, vol. 2. IEEE, 2008, pp. 302–305.

[196] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*. IEEE, 2009, pp. 138–143.

[197] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga, "Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey," in *Architecture of computing systems (ARCS), 2010 23rd international conference on*. VDE, 2010, pp. 1–10.

[198] A. Mannini, M. Rosenberger, W. L. Haskell, A. M. Sabatini, and S. S. Intille, "Activity recognition in youth using single accelerometer placed at wrist or ankle," *Medicine and science in sports and exercise*, vol. 49, no. 4, p. 801, 2017, pmid:27820724.

[199] A. G. Bonomi, A. H. Goris, B. Yin, and K. R. Westerterp, "Detection of type, duration, and intensity of physical activity using an accelerometer," *Medicine & Science in Sports & Exercise*, vol. 41, no. 9, pp. 1770–1777, 2009.

[200] W. Hu, D. Xie, T. Tan, and S. Maybank, "Learning activity patterns using fuzzy self-organizing neural network," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1618–1626, 2004.

[201] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, 2017.

[202] Z. Lv, X.-p. Wu, M. Li, and D. Zhang, "A novel eye movement detection algorithm for eog driven human computer interface," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 1041–1047, 2010.

[203] E. Fortune, V. Lugade, M. Morrow, and K. Kaufman, "Validity of using tri-axial accelerometers to measure human movement-part ii: Step counts at a wide range of gait velocities," *Medical engineering & physics*, vol. 36, no. 6, pp. 659–669, 2014.

[204] V. Lugade, E. Fortune, M. Morrow, and K. Kaufman, "Validity of using tri-axial accelerometers to measure human movement-part i: Posture and movement detection," *Medical engineering & physics*, vol. 36, no. 2, pp. 169–176, 2014.

[205] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1194–1201.

[206] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, p. 13, 2010.

[207] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

[208] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, "Self-contained position tracking of human movement using small inertial/magnetic sensor modules," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 2526–2533.

[209] J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 7, pp. 2998–3007, 2012.

[210] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors Journal*, vol. 17, no. 2, pp. 386–403, 2017.

[211] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31 314–31 338, 2015.

[212] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE transactions on information technology in biomedicine*, vol. 10, pp. 156–167, 2006.

[213] M. J. Mathie, B. G. Celler, N. H. Lovell, and A. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, no. 5, pp. 679–687, 2004.

[214] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 6, pp. 1780–1786, 2014.

[215] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2213–2220, 2008.

[216] E. Garcia-Ceja, R. F. Brena, J. C. Carrasco-Jimenez, and L. Garrido, "Long-term activity recognition from wristwatch accelerometer data," *Sensors*, vol. 14, no. 12, pp. 22 500–22 524, 2014.

[217] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on information technology in biomedicine*, vol. 10, no. 1, pp. 119–128, 2006.

[218] H. Gjoreski, M. Lustrek, and M. Gams, "Accelerometer placement for posture recognition and fall detection," in *2011 Seventh International Conference on Intelligent Environments*. IEEE, 2011, pp. 47–54.

[219] F. Chamroukhi, S. Mohammed, D. Trabelsi, L. Oukhellou, and Y. Amirat, "Joint segmentation of multivariate time series with hidden process regression for human activity recognition," *Neurocomputing*, vol. 120, pp. 633–644, 2013.

[220] L. Gao, A. K. Bourke, and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems," *Medical engineering & physics*, vol. 36, no. 6, pp. 779–785, 2014.

[221] S. Mehrang, J. Pietila, and I. Korhonen, "An activity recognition framework deploying the random forest classifier and a single optical heart rate monitoring and triaxial accelerometer wrist-band," *Sensors*, vol. 18, no. 2, p. 613, 2018.

[222] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[223] T. Brezmes, J.-L. Gorricho, and J. Cotrina, "Activity recognition from accelerometer data on a mobile phone," in *International Work-Conference on Artificial Neural Networks*. Springer, 2009, pp. 796–799.

[224] K. Altun, B. Barshan, and O. Tuncel, "Comparative study on classifying human activities with miniature inertial and magnetic sensors," *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, 2010.

[225] P. Casale, O. Pujol, and P. Radeva, "Human activity recognition from accelerometer data using a wearable device," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2011, pp. 289–296.

[226] A. Gijsberts, M. Atzori, C. Castellini, H. Muller, and B. Caputo, "Movement error rate for evaluation of machine learning methods for semg-based hand movement classification," *IEEE Trans.Neural Syst.Rehabil.Eng*, vol. 22, no. 4, pp. 735–744, 2014.

[227] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Aaai*, vol. 5, 2005, pp. 1541–1546.

[228] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[229] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*. Springer, 2004, pp. 1–17.

[230] F. R. Allen, E. Ambikairajah, N. H. Lovell, and B. G. Celler, "Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models," *Physiological Measurement*, vol. 27, no. 10, p. 935, 2006.

[231] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui, "Learning and detecting activities from movement trajectories using the hierarchical hidden markov model," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 955–960.

[232] J. Mantyjarvi, J. Himberg, and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 747–752.

[233] J. Baek, G. Lee, W. Park, and B.-J. Yun, "Accelerometer signal processing for user activity detection," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2004, pp. 610–617.

[234] M. M. Novicic, M. M. Jankovic, G. S. Kvascev, and M. B. Popovic, "Classification of forearm movements based on kinematic parameters using artificial neural networks," in *Telecommunication Forum (TELFOR), 2017 25th*. IEEE, 2017, pp. 1–4.

[235] K.-T. Song and Y.-Q. Wang, "Remote activity monitoring of the elderly using a two-axis accelerometer," in *Proceedings of the CACS Automatic Control Conference*, 2005, pp. 18–19.

[236] M. Sekine, T. Tamura, M. Akay, T. Fujimoto, T. Togawa, and Y. Fukui, "Discrimination of walking patterns using wavelet-based fractal analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 3, pp. 188–196, 2002.

[237] T. R. Burchfield and S. Venkatesan, "Accelerometer-based human abnormal movement detection in wireless sensor networks," in *Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*. ACM, 2007, pp. 67–69.

[238] D. Curone, G. M. Bertolotti, A. Cristiani, E. L. Secco, and G. Magenes, "A real-time and self-calibrating algorithm based on triaxial accelerometer signals for the detection of human posture and activity," *IEEE transactions on information technology in biomedicine*, vol. 14, no. 4, pp. 1098–1105, 2010.

[239] F. Foerster, M. Smeja, and J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring," *Computers in Human Behavior*, vol. 15, no. 5, pp. 571–583, 1999.

[240] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simulation Modelling Practice and Theory*, vol. 18, no. 4, pp. 446–455, 2010.

[241] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[242] A. Fougner, E. Scheme, A. D. Chan, K. Englehart, and O. Stavdahl, "A multi-modal approach for hand motion classification using surface emg and accelerometers," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 4247–4250.

[243] Y. Xue, Z. Ju, K. Xiang, C. Yang, and H. Liu, "Dexterous hand motion classification and recognition based on multimodal sensing," in *International Conference on Intelligent Robotics and Applications*. Springer, 2017, pp. 450–461.

[244] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.

[245] J. Kela, P. Korpip, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca, "Accelerometer-based gesture control for a design environment," *Personal and Ubiquitous Computing*, vol. 10, no. 5, pp. 285–299, 2006.

[246] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 18, no. 11, p. 1473, 2008.

[247] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[248] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[249] Mathworks, "Find signal location using similarity search," https://es.mathworks.com/help/signal/ref/findsignal.html, [Online; accessed 16-July-2018].

[250] D. Rivera, A. García, B. Alarcos, J. R. Velasco, J. E. Ortega, and I. Martínez-Yelmo, "Smart toys designed for detecting developmental delays," *Sensors*, vol. 16, no. 11, p. 1953, 2016.

[251] J. R. Velasco, B. Alarcos, A. García, D. Rivera, *et al.*, "Sistema de sondas inteligentes de monitorización aplicado a objetos de uso cotidiano," 2016, patent application number: P201600597.

[252] ——, "Sistema de monitorización de actividades con clavijas," 2016, patent application number: P201600945.

[253] K. van der Meulen, C. del Barrio, D. Rivera, J. E. Ortega, and J. R. Velasco, "Technology contributing to child studies: Toddlers' manual dexterity using an electronic pegboard," in *18th European Conference On Developmental Psychology*.

[254] D. Rivera and A. García, *Desarrollo de juguetes inteligentes para detectar problemas de desarrollo en niños*. Madrid Spain: Jornadas de Jovenes Investigadores, UAH, 2017.

[255] F. Taffoni, D. Rivera, A. La Camera, A. Nicolò, J. R. Velasco, and C. Massaroni, "A wearable system for real-time continuous monitoring of physical activity," *Journal of healthcare engineering*, vol. 2018, 2018.

[256] M. A. Gutiérrez García, M. L. Martín-Ruiz, D. Rivera, L. Vadillo, and M. A. Valero Duboy, "A smart toy to enhance the decision-making process at children's psychomotor delay screenings: A pilot study," *Journal of medical Internet research*, vol. 19, no. 5, 2017.

[257] D. Rivera, M. L. Martín-Ruiz, M. A. Gutiérrez García, and J. R. Velasco, "Verifying sensors in smart toys designed to help professionals in the early detection of psychomotor developmental disorders," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 1, no. 8, p. 800, 2017.

[258] D. Rivera, L. Cruz-Piris, G. Lopez-Civera, E. de la Hoz, and I. Marsa-Maestre, "Applying an unified access control for iot-based intelligent agent systems," in *Service-Oriented Computing and Applications (SOCA), 2015 IEEE 8th International Conference on*. IEEE, 2015, pp. 247–251.

[259] L. Cruz-Piris, D. Rivera, G. Lopez-Civera, E. de la Hoz, I. Marsa-Maestre, and J. R. Velasco, "Protecting sensors in an iot environment by modelling communications as resources," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 1, no. 8, p. 801, 2017.

[260] L. Cruz-Piris, D. Rivera, I. Marsa-Maestre, E. de la Hoz, and J. R. Velasco, "Access control mechanism for iot environments based on modelling communication procedures as resources," *Sensors*, vol. 18, no. 3, p. 917, 2018.

[261] D. Rivera, A. García, L. Martín-Ruíz, B. Alarcos, J. Velasco, and A. Gómez Oliva, "Secure communications and protected data for a internet of things smart toy platform," *IEEE IoT Journal (Submitted)*, 2018.

[262] D. Rivera, L. Cruz-Piris, S. Fernández, B. Alarcos, A. García, and J. Velasco, "A novel method for automatic detection and classification of movement patterns in short duration playing activities," *IEEE Access (Accepted)*, 2018.

Lista de acrónimos

6loWPAN	IPv6 over Low power Wireless Personal Area Networks
AAT	Authorization API Token
ABAC	Atribute Based Access Control
AES	Advanced Encryption Standard
AMQP	Advanced Message Queue Protocol
ANN	Artificial Neural Network
API	Application Programmable Interface
AS	Authorization Server
AWS	Amazon Web Services
BLE	Blueetooth Low Energy
BPMN	Bussiness Process Model and Notation
CapBAC	Capability Based Access Control
CBC	Cipher Block Chaining
CoAP	Constrained Application Protocol
DDS	Data Distribution Service
DFT	Discrete Fourier Transform
DoS	Denegation of Service
DTLS	Datagram Trasnsport Layer Security
DTW	Dynamic Time Warping
EDUCERE	Ecosistema de Detección Ubicua, atenCión y Estimulación tempRana para niños con trastornos del dEsarrollo
EMG	ElectroMyoGraphy
EOG	ElectroOculoGraphy
GA	Genetic Algorithm
GE	Generic Enabler
GMM	Gaussian Mixture Model
HMAC	Hash-based Message Authentication Code
HMM	Hidden Markov Model

HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
i2c	Inter-integrated Circuit
ICC	IntraClass Correlation
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things
IoT-A	Internet of Things-Architecture
IoToys	Internet of Toys
IV	Inicialization Vector
JCR	Journal Citation Report
KDF	Key Derivation Function
KMAC	KECCAK Message Authentication Code
kNN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
LDR	Light-Dependent Resistors
LED	Light-Emiting Diode
LiPo	Lithium Polymer
LSO	Lifecycle Service Orchestration
M2M	Machine to Machine
mDNS	multicast Domain Name System
MAC	Message Authentication Code
MQTT	Message Queuing Telemetry Transport
NFC	Near-Field Communication
OAuth	Open Authorization
OMG	Object Management Group
OrBAC	Organization Based Access Control
PAN	Personal Area Network
PAT	Protection API Token
PCB	Printed Circuit Board
PSD	Power Spectral Density
PUF	Physical Unclonable Function
RBAC	Role-Based Access Control
REST	REpresentational State Transfer
RFID	Radio-Frequency IDentification
RGB	Red-Green-Blue
RMS	Root Mean Square
RO	Resource Owner

RP	Requesting Party
RPL	Routing Protocol for Low power and lossy networks
RPT	Requesting Party Token
RS	Resource Set
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer
SDL	Security Development Lifecycle
SHA-3	Secure Hash Algorithm 3
SiP	System in Package
SPI	Serial Peripheral Interface
SPITFIRE	Semantic-service Provisioning for the Internet of Things using Future Internet Research by Experimentation
SSL	Secure Sockets Layer
SVM	Support Vector Machine
TLS	Transport Layer Security
UCON	Usage CONtrol access control
UMA	User-Managed Access
UML	Unified Modeling Language
URL	Uniform Resource Locator
WHO	World Health Organization
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
XMPP	eXtensible Messaging and Presence Protocol