

## DESIGN AND DEVELOPMENT OF AN INTERACTION SYSTEM IN ORDER TO BE IMPLEMENTED IN A SMART CLASSROOM

 CHRISTIAN ANDRÉS DIAZ LEÓN<sup>1</sup>

EDWIN MAURICIO HINCAPIÉ MONTOYA<sup>2</sup>

EDISON ANDRÉS GUIRALES ARREDONDO<sup>3</sup>

GUSTAVO ADOLFO MORENO LOPEZ<sup>4</sup>

### ABSTRACT

Smart Classrooms are a new educational paradigm aimed to determine, by a set of sensors, what is happening in the classroom and from this information and using artificial intelligence strategies infer what methodological variations or content must be applied by the teacher, in order to optimize the pedagogical practices. An important component of these classrooms is the use of new interaction strategies that facilitate the use of educational content. This paper describes the design and development of an interaction system, which uses different strategies to interact with contents proposed for the smart classroom. These interaction strategies are based on gestures, interactive surfaces and gestural touch. Finally, the article proposes experimental tests to validate the different forms of interaction described.

**Keywords:** Smart classrooms; Human computer interaction; Gesture recognition; Interactive surfaces.

## DISEÑO Y DESARROLLO DE UN SISTEMA DE INTERACCIÓN PARA SU IMPLEMENTACIÓN EN UN AULA DE CLASE INTELIGENTE

### RESUMEN

Las aulas de clase inteligentes son un nuevo paradigma educativo cuyo objetivo es determinar mediante un conjunto de sensores que está sucediendo dentro del aula, y a partir de esta información —y usando estrategias de inteligencia artificial— inferir qué variaciones metodológicas o de contenido deben ser aplicadas por parte del docente, con el fin de optimizar las prácticas pedagógicas. Un componente importante de estas aulas es el uso de nuevas estrategias de interacción, que facilitan el manejo de los contenidos pedagógicos utilizados. En este artículo se presenta el diseño y desarrollo

---

<sup>1</sup> Ingeniería Biomédica, Magíster en Ingeniería Informática, Candidato a Doctor en Ingeniería. Docente Investigador, Institución Universitaria Salazar y Herrera.

<sup>2</sup> Ingeniero en Instrumentación y Control, Magíster en Matemáticas Aplicadas, Doctor en Ciencias de la Ingeniería. Director del Centro de Investigación, Institución Universitaria Salazar y Herrera.

<sup>3</sup> Estudiante Ingeniería de Sistemas Institución Universitaria Salazar y Herrera.

<sup>4</sup> Licenciado en Educación, Magíster en Educación. Director Académico Institución Universitaria Salazar y Herrera.



*Author's Mailing Address:* Díaz León, C.A. (Christian Andrés): Calle 70 Sur No. 38 - 305, Apto 1906, Conjunto Loma Linda, Sabaneta, Antioquia / Tel.: (574) 2888016. Correo electrónico: c.diaz@iush.edu.co

*Paper history:*

Paper received: 27-IV-2015 / Approved: 14-X-2016

Available online: February 30, 2017

Open discussion until April 2018



de un sistema de interacción, que usa diferentes estrategias para interactuar con los contenidos propuestos por el aula de clase inteligente. Estas estrategias de interacción están basadas en gestos, superficies interactivas no instrumentadas y sistemas de pulsación gestual. Finalmente, el artículo propone pruebas experimentales para validar las diferentes formas de interacción propuestas.

**PALABRAS CLAVE:** Aulas de clase inteligente; interacción hombre – computador; reconocimiento de gestos; superficies interactivas.

## DESENHO E DESENVOLVIMENTO DE UM SISTEMA DE INTERAÇÃO PARA SUA IMPLEMENTAÇÃO NUMA SALA DE CLASSE INTELIGENTE

### RESUMO

As salas de classe inteligentes são um novo paradigma educativo que visa determinar mediante um conjunto de sensores o que está acontecendo na sala de aula, e a partir dessa informação e utilizando estratégias de inteligência artificial inferir quais variações metodológicas ou de conteúdo devem ser aplicadas pelo professor, para otimizar as práticas pedagógicas. Um componente importante destas aulas é o uso de novas estratégias de interação que facilitam o uso de conteúdos pedagógicos utilizados. Neste artigo apresenta-se o desenho e desenvolvimento de um sistema de interação, que utiliza diferentes estratégias para interagir com conteúdos propostos para a sala de aula inteligente. Estas estratégias de interação estão baseadas em gestos, superfícies interativas não instrumentadas e sistema de pulsación gestual. Finalmente, o artigo propõe testes experimentais para validar as diferentes formas de interação descritas.

**PALAVRAS-CHAVE:** Salas de classe inteligente; Interação homem – computador; Reconhecimento de gestos; Superfícies interativas.

---

### 1. INTRODUCTION

The objective of smart classrooms is to improve pedagogical practices within them by means of analysis of the context of the classroom and its adaptation, using different ways of content presentation and/or varying the teaching methodologies used (Chen y Huang, 2012; Papatheodorou et al., 2010). When we refer to context in this sense, we refer to all those metrics that allow us to determine what is happening in the classroom, for example, the type of activity (teacher oriented or group work), student disposition or attention level with respect to the class, among others (Chen y Huang, 2012). Stemming from this information, a smart classroom determines what must be improved

and makes recommendations to the teacher. These recommendations include anything from a change in teaching practice to a change in the way the content should be presented (for example, using slides or augmented reality).

A very important component for these systems is one which offers diverse ways of interaction with the content, for example, gesture recognition systems (Bailly et al., 2012). Nowadays, these smart classrooms propose the use of different forms of student-teacher interaction with the content for the purpose of offering more life-like ways of teaching which reinforce memory and comprehension of complex and abstract concepts. Consider that the new interaction paradigms are grouped into two

major categories: (i) those which require physical interaction between the user and the system and (ii) those which do not require physical interaction between the user and the system. By system, we refer to the smart classroom and its content and by physical interaction we refer to when the user comes in physical contact with some component of the system to indicate his intention or what he wishes to do.

Several papers have proposed the use of multi-tactile surfaces, which are in the first interaction category, as a principle for interaction between the user and the exposed content in the smart classroom (Bailly et al., 2012; Zhao et al., 2014; Nacher et al., 2014; Novotny et al., 2013). In (Nacher et al. 2014), the development of applications that use tactile surfaces as a learning mechanism for pre-kindergarten children (2-3 years of age) is described. These applications show a better assimilation in coordination and relationship of ideas after having experienced training with these technologies. Also, different types of interaction have been combined with forms of presentation (Novotny et al., 2013). In this case, the authors use tactile systems with augmented reality for the viewing of virtual assets, as an alternative to learning about places and events in history which have disappeared in determined places and cities. Other papers, albeit in a lesser amount, have focused on applying the second type of interaction described in this article. For example, in (Seo y Yeol, 2013), an augmented reality solution is proposed which allows the manipulation of multimedia objects by using Kinect. On the other hand, several types of applications that do not use tactile surfaces to interact with a system are described in (Krejov et al., 2014), wherein the idea is proposed that this type of interaction is much more intuitive and easier for the user to learn.

However, choosing a type of interaction for a specific content is related to the task at hand. For this reason, this article proposes the use of interaction paradigms from both categories to be implemented in the interaction system of a smart classroom. The

proposed interaction system in this article is made up of three modules which allow users to interact with content using three different paradigms. The gesture module allows the recognition of different gestures in order to map them to a determined action in the classroom. This module was described in the authors' paper (Author et al., 2015), wherein validation as an interaction system in the smart classroom was also made. The gestural touch module allows for the interaction with the content without having to touch any surface. It is enough to move the hand and subsequently make a movement as if one wanted to press a button. For this purpose, this module integrates several libraries for the deep data recognition described in (Seo y Yeol, 2013; Al Delail et al., 2012). The interactive surface module allows a much more natural interaction since the person will touch the surface where the image with which he will interact is projected. The user will be able to zoom, move and rotate the image by using only his hands (Bailly et al., 2012; Zhao et al., 2014). It's important to note these surfaces are not instrumented. This module has the characteristic of monitoring a person's hands and fingers at all times since the user will be able to make commands with these for the manipulation of the content. For example, the user will be able to click a key, zoom, point or activate options. The module also monitors the hand depth level (Novotny et al., 2013; Krejov et al., 2014). In contrast to state of the art, this article concatenates the different modules in an interaction system providing the user a sensation of an intuitive interaction with the system.

## 2. MATERIAL AND METHODS

### *Description of the interaction system*

The role the system of interaction plays within the smart classroom is allowing the students and teacher to interact with the established content in order to teach in an easy and natural way. Using the three interaction forms proposed in this article, the user will be able to move, activate

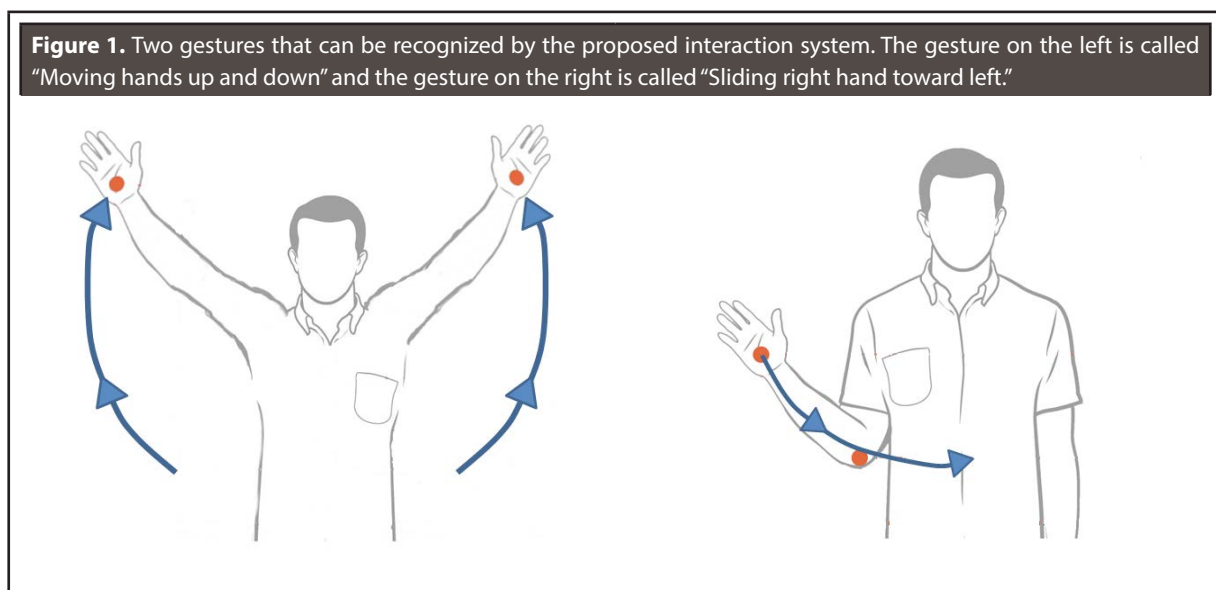
and choose the diverse content in the appropriate manner. This section describes the three systems that make up the interactive classroom modules proposed in the article.

The first proposed interaction module is the gesture recognition module which is in charge of characterizing previously recorded movements (gesture calibration) and subsequently, based on the trajectory of these movements, recognizing the person's gestures or movements. This is achieved by applying a mathematical algorithm called Dynamic Time Warping (DTW) which allows for both the comparison and the statistical calculation for gesture recognition regardless of height, physical make-up and gender of person, since these are differentiating factors that affect each human's movement (Seo y Yeol, 2013; Krejov et al. 2014). The implementation of this technique helps the processing of information captured by the camera and helps data calibration to have a much faster response time, avoiding mathematical inference rules that may take us to a slower processing of data. Subsequently, the movement is homologated with certain commands for interaction with implemented modules in Unity3D (Goldstone, 2011). Unity3D is a platform for the development of applications in virtual reality, augmented reality, video games and viewing and

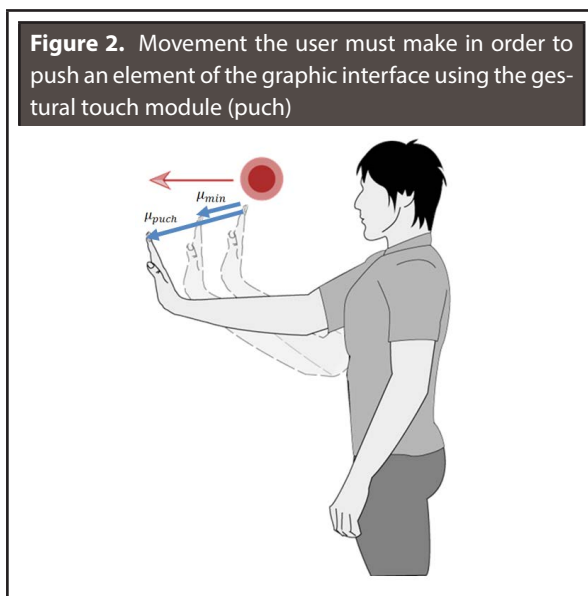
it is the smart classroom component in which the different content types are displayed. In Figure 1, two types of gestures that can be carried out within the classroom and are presented and the gesture recognition systemisable to recognize and map them to a command for interaction with the content.

The second proposed interaction module is the gestural touch module (puch). This module allows persons to interact with content at a determined distance stemming from movement, without needing to touch any object or surface. The model's characteristic is the automatic recognition of the person who will manipulate the system and it can even recognize the right or left hand, visually indicating the hand with which the person will manipulate the system. During the development of the gestural touch module, it was determined that the most important factor for the correct functioning is calculating, in front of the Kinect device, the person's distance and based on said distance, determine the hand depth while moving them forward in order to signal and activate the multimedia content. Additionally, the model needs to be sufficiently robust to support a certain margin of error and avoid that any natural movement made by the user be able to activate the different configured comands.

**Figure 1.** Two gestures that can be recognized by the proposed interaction system. The gesture on the left is called "Moving hands up and down" and the gesture on the right is called "Sliding right hand toward left."



**Figure 2** shows the gesture a person needs to make in order to push any element in the user's graphic interface. The figure also shows two variables defined as  $\mu_{min}$  and  $\mu_{puch}$  which characterize the movement thresholds from which the activation of the puch gesture in the module is defined. If the movement is greater than  $\mu_{min}$  and less than  $\mu_{puch}$  it means that the puch event is activated and dependant on the control that makes part of the interface, a key, is activated.



In the development of the interactive surface module, several variables were taken into account which could affect hand recognition, such as lighting conditions, surface refraction and textures, along with the different surfaces where the interaction might take place. For this reason, advanced libraries were used for the processing of images and characteristics since the Kinect programming modules offer very basic tools in this respect. As such, the use of OpenCV was decided. This library is compatible with C, C++, Python and Java programming languages. Among its main applications are object recognition, camera calibration and applications for robotic vision. To integrate OpenCV with the Kinect device, OpenNiconrollers were used. These drivers not only

provide connection to the Kinect cameras, but also to the microphones for sound recognition.

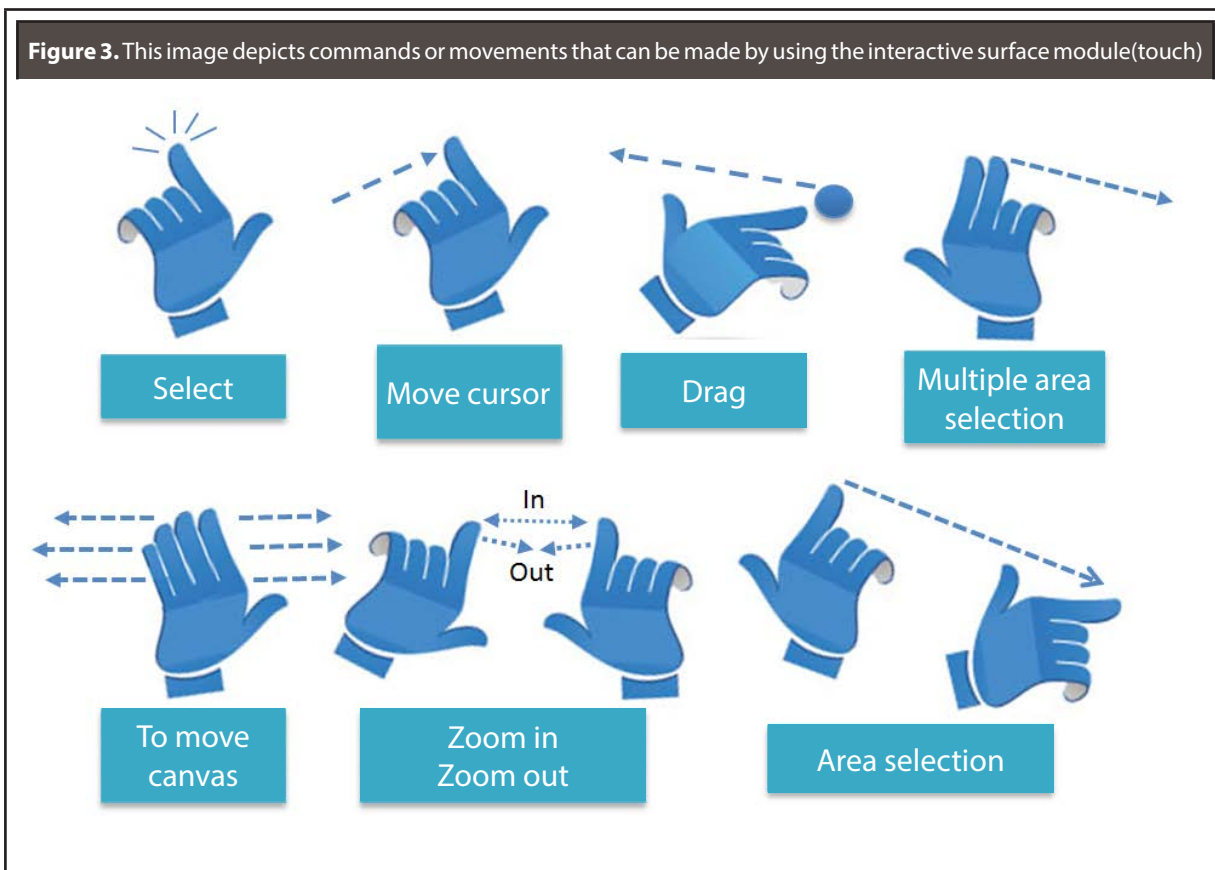
The interactive surface system recognizes each finger of the hand of the person who is manipulating the system. This translates to commands that are programmed for specific tasks, be it operation systems interaction, specific software (drawing, image viewer) or the content modeled in 3D, such as using the video game engine called Unity, for example (Goldstone, 2011).

Additionally, the interactive Surface system provides information when there is contact from any of the fingers on the calibrated surface or the one recognized as the work space. These main difficulties in recognizing contact with calibrated surfaces were light refraction, image capturing angle recognition with the Kinect device and shiny surfaces. These problems were solved by applying optimal distance calculation algorithms and orthogonolization which allowed the minimization of margins of error so that the system could be functional and practical in the classroom (Arranz, 2013). This article shows the use of these algorithms applied to the processing of images from different angles and heights in order to subsequently make up a map or document of the analyzed zone which will offer reference information subsequently.

Due to the importance of this photographic information, image quality must be optimal. Different movements that can be made in order to interact with content using module 3 of the interaction system are shown in **Figure 3**.

### *Description of method used to calculate depth and identify surfaces*

Following, two methods used both for the gestural touch module and the interactive surface module are described. As mentioned above, the method used for gesture recognition is DTW, which was described in (Author et al., 2015).



**Depth calculation**

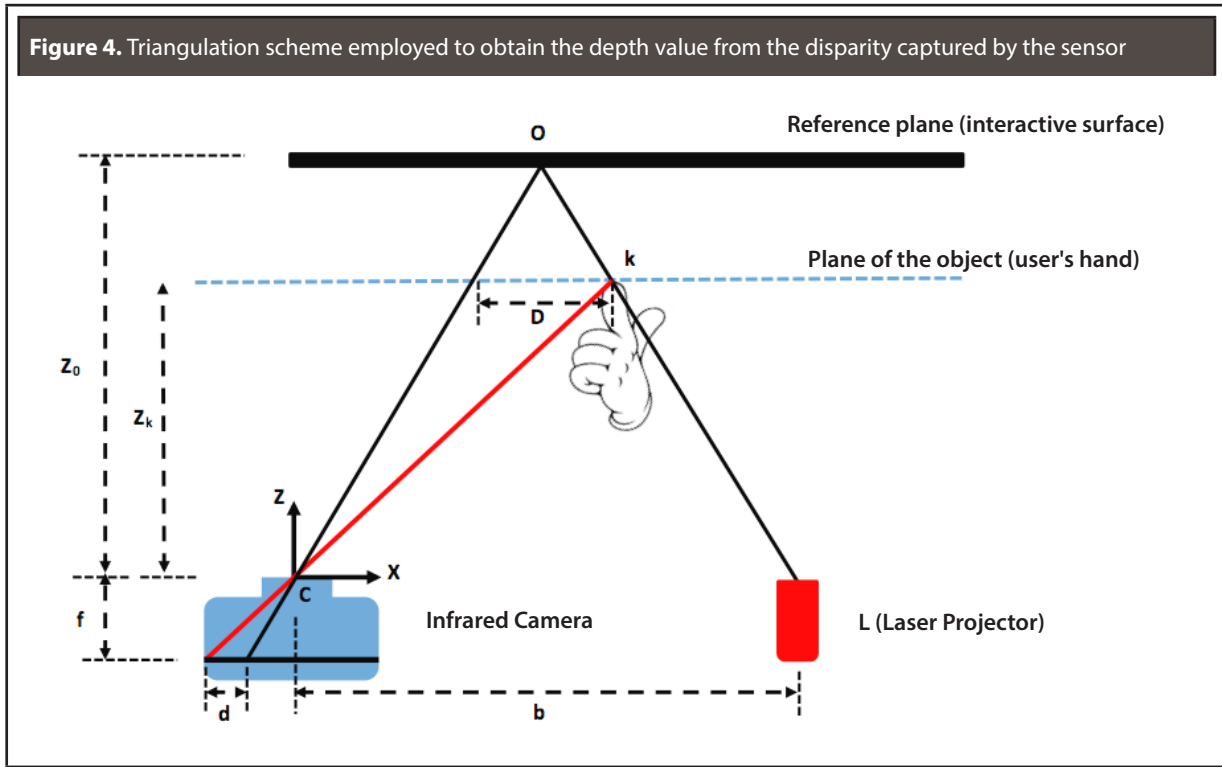
To describe the mathematical procedure by which the Kinect will be able to obtain coordinates in the space of a given object, we used **Figure 4** as a guide and based ourselves on the description presented by (Magallón, 2013). We can observe the triangular relationship between the object point  $k$  and the measured disparity  $d$ . It's important to note that the three-dimensional system of the origin of coordinates is situated in the infrared camera with the  $Z$  orthogonal axis on the plane of the image and directed toward the object  $y$ , the  $X$  axis, base line  $b$  which joins the camera to the projector perpendicular to the previous one.

In **Figure 4**, we can see that the object plane is located at a distance of  $Z_k$  less than the distance reference plane  $Z_0$ . For this reason, when the pattern

is projected from  $L$ , the projected point that, on the plane of reference seen by  $C$  is situated on  $o$ , will now be situated on  $k$ . From  $C$  we can observe a movement on the  $X$  axis to the right of magnitude  $D$ . If the object were located farther than the plane of reference, the movement would have been toward the left. What the sensor measures and registers is not directly the distance  $D$  of the space of the object but the disparity  $d$ . For each point  $k$  he parameter is measured and a disparity map is obtained. From an analysis of the triangles in the figure, the following equations may be deduced:

$$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0} \tag{1}$$

$$\frac{d}{f} = \frac{D}{Z_k} \tag{2}$$



Where  $Z_k$  is the distance (depth) and point  $k$  of the object is the variable we wish to obtain;  $b$  is the length of the base understood as the distance between the camera and the projector;  $f$  is the focal distance of the infrared sensor;  $D$  is the real distance of movement from point  $k$  in the space of the object and  $d$  is the observed disparity in the space of the image. Substituting  $D$  from **Equation 2** in **Equation 1** and clearing the variable we obtain:

$$Z_k = \frac{Z_0}{1 + \frac{Z_0}{f_b} d} \quad (3)$$

The parameters  $Z_0$ ,  $f$  and  $b$  can be obtained applying a calibration process.

The coordinate  $Z$  of each point with  $f$  defines the scale of the image for this point. The planimetric coordinates of the object in each point can be calculated from the image coordinates and the scale:

$$X_k = -\frac{Z_k}{f} (x_k - x_0 + \delta x) \quad (4)$$

$$Y_k = -\frac{Z_k}{f} (y_k - y_0 + \delta y)$$

Where  $x_k$  and  $y_k$  are the coordinates on the image of the point,  $x_0$  and  $y_0$  are the coordinates of the main point, that is, of the offset of the image and  $\delta x$  and  $\delta y$  refer to the corrections made as a product of the distortion of the lens. Both the offset values and corrections can be obtained from the calibration process.

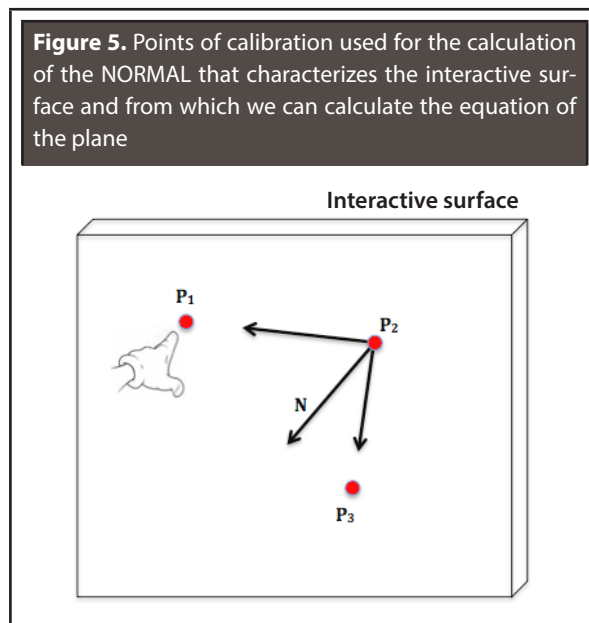
With the known calibration parameters, we can complete the relation between the mid points of the image  $(x, y, d)$  and the coordinates of the object  $(X, Y, Z)$  of each point. This way we can generate a cloud of points of each image of disparity. The disparity herein described is a measure of inverse depth. Greater values of the same mean shorter distances.

But the value of disparity the Kinect returns,  $d'$  is normalized with an offset according to the relation:

$$d' = d_{\text{offset}} - 8d \quad (5)$$

### Surface detection

The method of surface detection is used by the interactive surface module. To detect the surface, a mathematical equation is used which calculates the equation of a plane from three points that make up part of the surface. These three points are captured by the Kinect, applying the previously described method. This is done during a calibration stage in which the user indicates to the system three points of the surface that will be interactive. In **Figure 5**, we can observe the interactive surface that will be detected and the three points  $P_1, P_2, P_3$ , chosen by the user to calculate the plane of the interactive surface.



From these points the normal  $N$  to the plane can be calculated, with the following equation:

$$N = \frac{P_3 - P_2}{\|P_3 - P_2\|} \times \frac{P_1 - P_2}{\|P_1 - P_2\|} \quad (6)$$

$$N = (nx, ny, nz) \quad (7)$$

Considering the equation of a plane can be expressed in the following manner:

$$Ax + By + Cz + D = 0 \quad (8)$$

We can obtain the equation of the plane that characterizes the interactive surface from the normal calculated and a point that makes part of the plane, using the following expressions:

$$A = nx \quad (9)$$

$$B = ny \quad (10)$$

$$C = nz \quad (11)$$

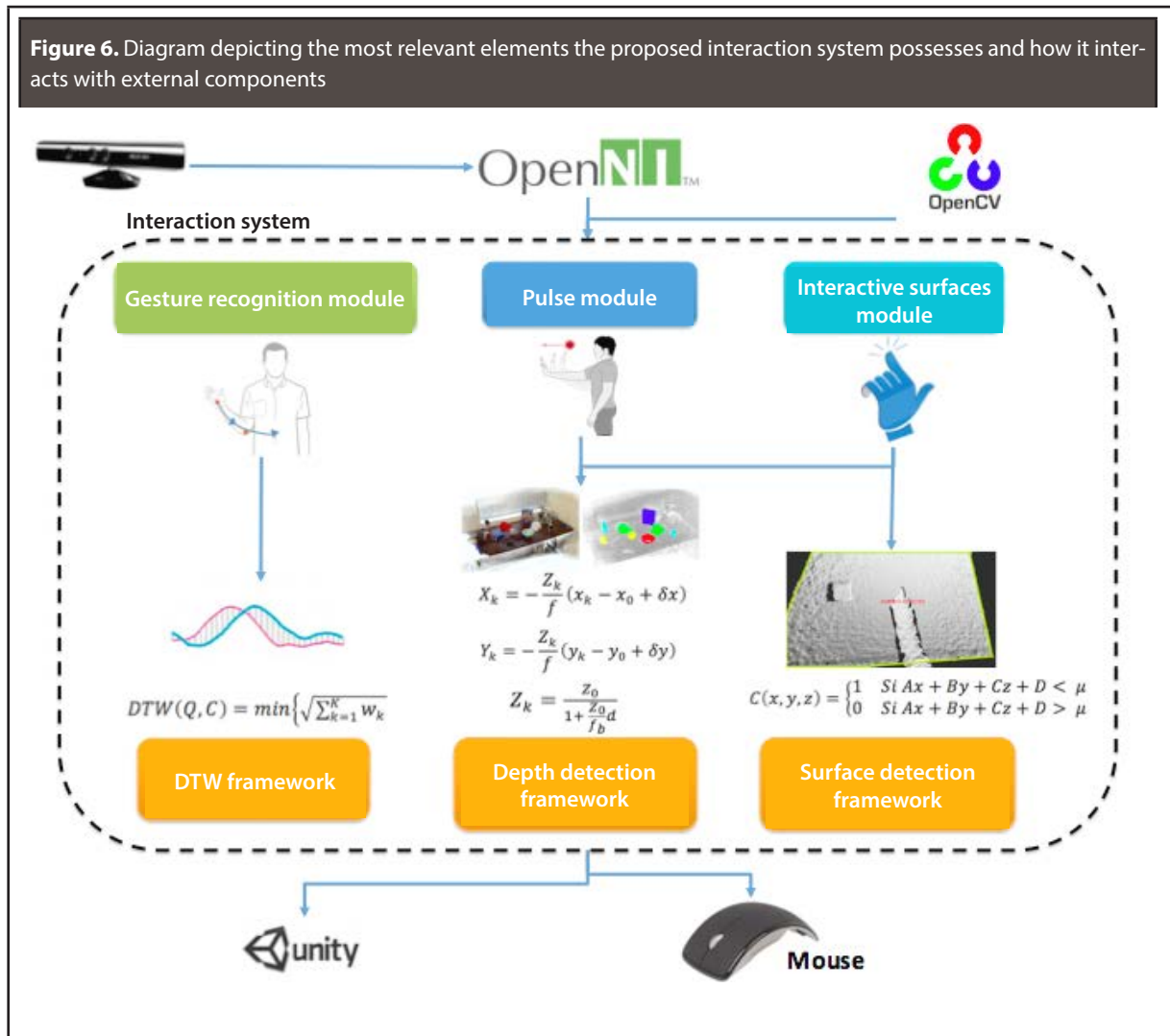
$$D = -Ax_1 - By_1 - Cz_1 \quad (12)$$

Where  $P_1 = (x_1, y_1, z_1)$  y  $nx, ny$  and  $nz$  are the components of the calculated normal. Once the plane representing the interactive surface from the calibration points has been calculated to see if the user is making contact with fingers, the current coordinate of the fingers on the equation of the plane is replaced and if it is equal to 0, it means the user is making contact with the surface. Since Kinect can make errors in the precision of the calculation of depth, a threshold of  $\mu$  is established which corresponds to the distance between the coordinate captured by the Kinect and the plane of the interactive surface, in which it is considered that contact exists. This is represented through the following expression:

$$C(x, y, z) = \begin{cases} 1 & \text{Si } Ax + By + Cz + D < \mu \\ 0 & \text{Si } Ax + By + Cz + D > \mu \end{cases} \quad (13)$$

Where  $C(x, y, z) = 1$  means the user is contacting the interactive surface and  $C(x, y, z) = 0$  means the contrary.  $x, y$  and  $z$  are the coordinates of the point that defines the user's fingers.





### Description of the interaction system architecture

This system possesses an architecture that allows for the optimization of all Kinect device resources, as well as the computer device's. The aim is to offer a good flow of the system at the time of interaction with the other components with which it should integrate, for example, operating system and visualization software, among others.

The Open CV libraries for the interaction modules allow us to optimize code routines and to improve precision and the success with which the

independent user's intention of the applied paradigm is measured. **Figure 6** schematically shows in a block diagram, each one of the components that make up the interaction system.

The interaction system is made up of three modules: gesture recognition, gestural touch and interactive surfaces. Next to each module on Figure 6, there is an image with the type of movement the user must make in order to interact with the content using each paradigm. Each one of these modules uses one or several functions to measure the user's intention and map it to a command. These basic functions are DTW framework, depth

detection framework and surface detection framework. The first framework was described in (Arranz, 2013) and the last two frameworks were described in the previous section. Additionally, the interaction system uses OpenNI to be able to extract data coming from Kinect and OpenCV, such as image processing libraries in order to ease procedures for the described frameworks. Key equations that allow gesture recognition, hand depth determination and interactive surface detection can be seen with these frameworks. Finally, the system interacts with Unity3D to control content displayed, such as video games, virtual environments, viewing and augmented reality. Or it interacts with the operating system component that controls the mouse in order to control generic applications such as Paint or PowerPoint.

### 3. DISCUSSION AND RESULTS

For the experimental tests described in this section, a Kinect device connected to a computer were used. Said computer had the following characteristics: 4Gb RAM and dual core or above processor. For this test, the Kinect is placed at a 2-meter distance from the user with medium lighting (ambient). The interaction system with the pulsation and interactive surface module was installed on the computer for the purpose of carrying out the experimental test.

**Table 1** describes the general manner the tests can be carried out with the push module in order to establish faults and strengths of this type of interaction with the already posed educational content.

**TABLE 1. PARAMETERS TO BE USED IN THE EXPERIMENTAL TEST THAT WILL BE DEVELOPED WITH THE PUSH MODULE**

Experimental test for the push module	
In this framework, the user directs the cursor in the application and selects content, simulating the pushing of a button	
1) Precision could be measured when controlling the cursor.	How: taking times for locating links in the shape of buttons of the framework and executing them.
2) Recognition of left and right hands.	How: making repeated changes of hands and users, thus evidencing blocking or inconsistencies.
3) measuring minimum and maximum distance and system efficacy.	How: taking different distance samples from the Kinect to the system operator, determining minimum and maximum ranges of operation.

For this experimental test, there were 5 users which age range is 23 to 35 and average height of 1.72 m. **Table 2** shows the characteristics of each one of the evaluated users.

**TABLE 2. CHARACTERISTICS OF PERSONS WHO CARRY OUT TEST IN THE PUSH MODULE**

User	Sex	Age	Height
1	Female	23	1,68
2	Female	16	1,60
3	Male	28	1,80
4	Male	35	1,75
5	Male	24	1,81

These persons were initially placed at a distance of 1 meter to the Kinect. Taking the time a trained person takes as base, the average time an untrained person takes to action each one of the application's buttons was measured, 3 in this case, Physics, Chemistry and Astronomy. For this test, the framework must recognize the hand that

will guide the cursor independent of the person's physical build or his height. Each user carried out 3 repetitions of the test. The results obtained for this test are presented in **Table 3**. Considering that, on average, an expert takes 24 seconds in achieving the task, we can observe that, on average, a group of untrained users achieves the task in a time.

**TABLE 3. RESULTS OBTAINED IN EACH REPETITION OF THE TEST FOR EACH USER TRYING OUT THE PULSATION MODULE**

User	Repetition 1 (seconds)	Repetition 2 (seconds)	Repetition 3 (seconds)	Average (seconds)
1	24	22	28	25
2	25	27	24	25
3	29	23	18	23
4	31	27	23	27
5	28	21	22	24

In addition, because the application should have the possibility of quickly updating the configuration of the person who uses it, plus identifying which hand is using the pointer, an experimental test was applied to evaluate this characteristic of the interaction module. For this test, the 5 participants will be located at a 1-meter distance from the Kinect and will change each other's positions so that only one participant can be captured by the Kinect at any one time. This will be done by the 5 participants 5 times each, 50 seconds, guaranteeing that the application be able to self-calibrate and adapt to the

person using the system without getting blocked or making a recognition error.

Participants repeat the first test at 1.5 meters, 2 meters, 3.5 meters and 4 meters.

The results obtained from the above test allow us to pose that the system is stable in the self-calibration and identification process of the user who interacts with the system.

**Table 5** describes the proposed experiment for the evaluation of interactive surfaces. For the experimental test, the Kinect was placed at 1.5 meters from the surface we wish to convert to a tactile surface. Irregular, flat, matte and painted surfaces of any color can be used. Subsequently, once the Kinect is placed, the surface recognition will be initiated. This is done for the purpose allowing the module to be able to differentiate hands, surface distance and lighting conditions that might alter the Kinect's reading.

Each time a recognition is to be done with the interactive surface module, refreshing the work space and verifying that the surface changes when each finger touches it will be enough to know the Kinect is correctly identifying hand depth, as well as surface contact. **Table 5** shows the elements that can be evaluated for the experimental test proposed in this case.

**TABLE 4. RESULTS OF THE TEST MEASURING THE ADAPTATION OF THE PROPOSED SYSTEM. THE LABEL IDENTIFIED ALLUDES TO THE SUCCESSFUL CALIBRATION OF THE IDENTIFICATION OF THE USER. THE PROCESSING LABEL MAKES ALLUSION TO FAILURES IN THE USER IDENTIFICATION TIMES WHICH MUST NOT BE ABOVE 50 SECONDS**

Time	Distance	Repetition1	Repetition2	Repetition3	Repetition4	Repetition5
50 s	1.5 metros	identified	identified	identified	identified	identified
50 s	2 metros	identified	identified	identified	identified	identified
50 s	3.5 metros	identified	processing	identified	identified	identified
50 s	4 metros	identified	identified	identified	identified	identified

**TABLE 5.** PARAMETERS TO BE USED IN EACH EXPERIMENTAL TEST TO BE DEVELOPED WITH THE TACTICAL SURFACE MODULE

Proposed test for the interactive surface framework

In this framework, the user drives the application cursor and selects the content making direct touches to the surface and executing a command with fingers.

1) The precision could ne measured and the cursor could be controlled	How: placing the Kinect at different distances and executing self-calibration for these, this way verifying that it recognizes hands adequately.
2) Surface recognition	How: placing the Kinect on the projected surfaces with varying textures, then verifying that hand recognition and surface touch are correct
3) Measurement of minimum and maximum distance and system efficacy	How: taking different samples and angles of operation

**Figure 7.** Experimental configuration used to evaluate the interactive surface tool

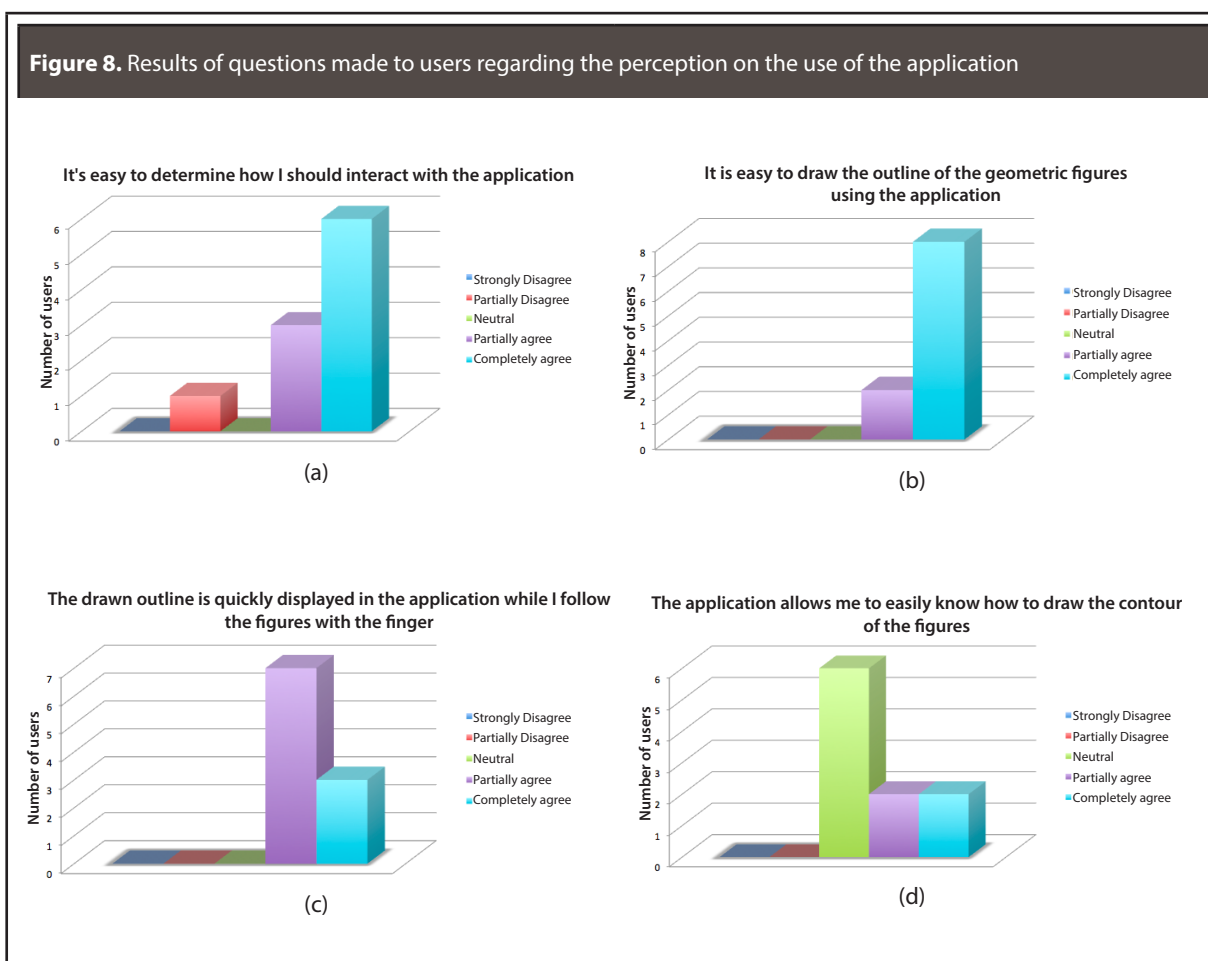


For the purpose of evaluating of the elements posed on **Table 5**, an experimental test was carried out in which the Kinect was located at 1.5 meters from the interactive surface. In this case, the interactive surface is a table on which the outline drawing of the three geometric shapes (square, triangle and circle) was located. The described configuration can be seen in **Figure 7**.

Windows Paint was used as the drawing tool. Once the user finished drawing the outlines of the three shapes, time was taken and the user answered a survey about his perception using the tool. Additionally, the reference image was saved, that is, the pattern drawn on the interactive surfaces. The outlines drawn by the users were also saved for the purpose of contrasting differences in the trajectory drawn by each one of them and the reference geometric figures. Ten total users were evaluated. Their ages ranged from 19 to 34. Four of them were female and six were male. All the participants had previous experience using the Paint tool and two users had previous experience using the Kinect device.

We can observe in **Figures 8a** to **8d** the results of the questions asked to each one of the users. The Likert scale question scheme was used to evaluate each one of the affirmations. From the results, we can observe that 90% of the users found it easy to determine how to interact with the application (**Figure 8a**). Notwithstanding, in **Figure 8d**, 60% has no opinion regarding whether or not it is easy to learn how to draw the outlines on the application. This is because, on the application, to draw an outline, one must use only the index finger and keep the other four fingers folded into the hand. Some users made the gesture incorrectly, for example, extending all fingers, which made the use of the application difficult.

**Figure 8.** Results of questions made to users regarding the perception on the use of the application



However, when users are conscious of how they should make the hand gesture, drawing outlines was simple for them, as is shown in **Figure 8b**. Finally, users agreed that as they drew the outline of the geometric shapes, the application quickly drew said outline in the Paint application (see **Figure 8c**).

Lastly, during the outline drawing task, time of task execution by the users was taken, as well as the number of errors. On average, users made less than one error during the delineation of the outline and took 10 seconds delineating the outline of the three shapes. Task execution time by users is considered good, taking into account an expert user completes the task in 8 seconds. On the other hand,

the majority of errors users made were a product of an incorrect gesture to delineate the outline of the geometric shapes.

On **Table 6**, we can observe the shapes drawn by the users, the number of errors each one made and the time it took to complete the task.

#### 4. CONCLUSIONS

The data output by the different tests applied to the interactive surface and multimedia gestural touch modules can be used to define that the architecture proposed, including hardware and software, allows an adequate interaction of the user with the smart classroom content displayed, using three types of interaction paradigms. This

allows for the selection of the interaction paradigm which best adjusts to the content developed.











In the case of interaction paradigms for the gesture recognition module and the gestural touch module, the proposed mathematical model and algorithms that are part of the framework are sufficiently robust for the needed interaction. However, the interactive surface module requires a better algorithm to recognize when a finger comes into contact with the interactive surface, since, if a gesture that does not allow the fingertip to be seen clearly is used, the system makes errors when determining whether or not it is interacting with the surface.

Additionally, during the development of the experimental tests, users were asked if they considered that the interaction with the system using any of the three paradigms was easy. They responded that it was very easy to learn and interact with the proposed system. Finally, stemming from the experimental tests to evaluate the interaction with interactive surfaces, we pose developing a more robust system that will allow the recognition of hand gestures, despite slight differences in the users'. Considering the entire interaction system proposed as a future project, we pose evaluating in which cases inside the classroom it would be more convenient to use one type of interaction over another.

**ACKNOWLEDGEMENTS**

A special thanks to the Research Center of the Institución Universitaria Salazar y Herrera for their constant support for the present research project.

**TABLE 6. RESULTS OBTAINED WITH USERS WHEN DRAWING THE DESCRIBED GEOMETRIC SHAPES**

User	Outline drawing result	Time (seconds)	Errors
1		15	1
2		25	0
3		24	0
4		11	2
5		9	0
6		16	1
7		17	2
8		9	2
9		11	0
10		9	0

## REFERENCIAS

- Chen, C.-C.; Huang, T.-C. (2012). Learning in a u-Museum: Developing a context-aware ubiquitous learning environment. *Computers & Education*, 59(3), pp. 873 – 883.
- Papatheodorou, C.; Antoniou, G.; Bikakis, A. (2010). On the Deployment of Contextual Reasoning in Ambient Intelligence Environments. *Sixth International Conference on Intelligent Environments (EI)*, pp. 13 – 18.
- Bailly, G.; Müller, J.; Lecolinet, E. (2012). Design and evaluation of finger-count interaction: Combining multi-touch gestures and menus. *International Journal of Human Computer Studies*, 70(10), pp. 673 – 689.
- Zhao, J.; Soukoreff, R.; Ren, X.; Balakrishna, R. (2014). A model of scrolling on touch-sensitive displays. *International Journal of Human Computer Studies*, 72(12), pp. 805 - 821.
- Nacher, V.; Jaen, J.; Navarro, E.; Catala, A.; González, P. (2014). Multi-touch gestures for pre-kindergarten children. *International Journal of Human-Computer Studies*, 73, pp.7-12.
- Novotný, M.; Lacko, J.; Samuelčík, M. (2013). Applications of Multi-Touch Augmented Reality System in Education and Presentation of Virtual Heritage, *Procedia Computer Science*, 25, pp. 231-235.
- Seo, W.; Yeol Lee, J. (2013). Direct hand touchable interactions in augmented reality environments for natural and intuitive user experiences. *Expert Systems with Applications: An International Journal archive*. 40(9), pp. 3784-3793..
- Krejoy, P.; Gilbert, A.; Bowden, R. (2014). A Multitouchless Interface Expanding User Interaction, *IEEE Computer Society 0272-1716*, pp. 2-8.
- Autor. (2015). Descripción de un Sistema de Reconocimiento de Gestos Para su Implementación en una Aula de Clase Inteligente. Pendiente de Publicación.
- Al Delail, B.; Weruaga, L.; Zemerly, J. (2012). CAViAR: Context Aware Visual indoor Augmented Reality for a University Campus. *IEEE/WIC/ACM International Conferences*, pp 2-4.
- Goldstone, W. (2011). *Unity 3.x Game Development Essentials*. Packt Publishing, Second Edition.
- Arranz, J. (2013). Diseño, optimización y análisis de sistemas basados en técnicas láser, para el modelado geométrico, registro y documentación, aplicados a entidades de interés patrimonial, Tesis doctoral, pp. 217-2225, pp. 334.
- Magallón, M. (2013) *Sistema Interactivo para Manejo de electrodomésticos en Entornos Domésticos*. Trabajo de grado en Ingeniería de Telecomunicaciones, Universidad de Zaragoza.

**TO REFERENCE THIS ARTICLE /  
PARA CITAR ESTE ARTÍCULO /  
PARA CITAR ESTE ARTIGO /**

Díaz León, C.A.; Hincapié Montoya, E.M.; Guirales Arredondo, E.A.; Moreno López, G.A. (2016). Design and Development of an Interaction System in Order to be Implemented in a Smart Classroom. *Revista EIA*, 13(26), July-December, pp. 95-109. [Online]. Available at: <https://doi.org/10.24050/reia.v13i26.666>