

INSTITUCIÓN UNIVERSITARIA POLITÉCNICO GRANCOLOMBIANO
FACULTAD DE INGENIERÍA Y CIENCIAS BÁSICAS
MAESTRÍA EN INGENIERÍA DE SISTEMAS
GRUPO DE INVESTIGACIÓN FICB-PG
LÍNEA DE PROFUNDIZACIÓN: RECONOCIMIENTO DE PATRONES

BoardApp: Tablero digital inteligente como
herramienta en el aula de clases

PRESENTA:
EDWIN HERNANDO ZAPATA CIFUENTES
CÓDIGO: 1420020029

ASESOR TEMÁTICO:
Laura Juliana Cortes MSc. en Ingeniería de Sistemas y Computación

Enero 2018

ÍNDICE GENERAL

ÍNDICE DE TABLAS.....	3
ÍNDICE DE FIGURAS	4
RESUMEN	5
ABSTRACT	5
PALABRAS CLAVE	5
KEY WORDS	5
INTRODUCCIÓN	6
Planteamiento del Problema.....	7
Justificación.....	8
OBJETIVOS.....	9
Objetivo general.....	9
Objetivos específicos.....	9
REVISIÓN DE LITERATURA / ANTECEDENTES.....	10
Tableros digitales.....	10
Reconocimiento de gestos	11
Estudios relacionados:	12
TIC's y Educación	12
Tableros digitales y reconocimiento de gestos	13
Reconocimiento de gestos y tableros digitales en educación.....	14
ESTRATEGIA METODOLÓGICA	15
DESARROLLO E IMPLEMENTACIÓN	17
Construcción de Requisitos	17
Requisitos Funcionales.....	17
Requisitos No Funcionales	18
Captura de datos	19
Pre-procesamiento	20
Aprendizaje	21
Diseño y desarrollo de la aplicación.....	24
Descripción de la aplicación final.....	28
RESULTADOS.....	30
DISCUSIÓN Y CONCLUSIONES	32
REFERENCIAS	33
ANEXOS	36
Anexo 1: Tabla que contiene los gestos a considerar	36
Anexo 2: Código del árbol de decisión implementado en BoardApp	40

ÍNDICE DE TABLAS

TABLA 1. MÉTODOLÓGÍA DE DESARROLLO BOARDAPP	16
TABLA 2. RESULTADOS OBTENIDOS POR CLASSIFICATION LEARNER DE MATLAB.....	24
TABLA 3. GESTOS RECONOCIDOS POR BOARDAPP.....	28

ÍNDICE DE FIGURAS

ILUSTRACIÓN 1. TIPOS DE TABLEROS DIGITALES.....	10
ILUSTRACIÓN 2. EJE CRONOLÓGICO DE USO E INVESTIGACIÓN DE RECONOCIMIENTO DE GESTOS.	11
ILUSTRACIÓN 3. PROCESOS PARA CONSEGUIR EL RECONOCIMIENTO DE GESTOS.....	18
ILUSTRACIÓN 4. SCREENSHOTS DE BOARDAPP1.	20
ILUSTRACIÓN 5. IMPORTACIÓN DE DATOS CON CLASSIFICATION LEARNER DE MATLAB.....	22
ILUSTRACIÓN 6. SELECTOR DE CLASIFICADORES EN CLASSIFICATION LEARNER DE MATLAB.	22
ILUSTRACIÓN 7. CLASSIFICATION LEARNER DE MATLAB MOSTRANDO RESULTADOS DEL ENTRENAMIENTO DE DIVERSOS CLASIFICADORES.....	23
ILUSTRACIÓN 8.. OPCIONES DE EXPORTACIÓN DEL MODELO DEL CLASIFICADOR SELECCIONADO EN MATLAB.....	24
ILUSTRACIÓN 9. DIAGRAMA DE CLASES PARA BOARDAPP,	25
ILUSTRACIÓN 10. MATRIZ DE CONFUSIÓN DADA POR CLASSIFICATION LEARNER DE MATLAB.	26
ILUSTRACIÓN 11. ACTIVITY 1 BOARDAPP.....	28
ILUSTRACIÓN 12. ACTIVITY 2 BOARDAPP.....	29
ILUSTRACIÓN 13. ACTIVITY 3 BOARDAPP.....	29

RESUMEN

Este documento describe el diseño y desarrollo de una aplicación que le permite a un docente trabajar sus clases desde un dispositivo móvil, dejando de lado el tradicional tablero y aprovechando que existen tabletas y teléfonos inteligentes con la capacidad de identificar las posiciones de hasta 10 dedos sobre su superficie. Esto genera información que, con técnicas de *reconocimiento de patrones*, permite al usuario interactuar con el software por medio de la manipulación táctil de la pantalla. Así, por ejemplo, es posible acceder a una regla, a la opción de escribir o cambiar de color de puntero, por medio de gestos táctiles; sin necesidad de ir hasta un menú a escoger las opciones.

ABSTRACT

This document describes the design and development of an app that allows to a teacher to work in his/her classes with a mobile device, leaving aside the traditional whiteboard and taking advantage of tablets and smartphones existence with the capability of identifying the positions of approximately 10 fingers in contact with the surface. This generates information that, using *pattern recognition* techniques, enables to the user to interact with the software throughout of the handling of the touch screen, thus for example, to access to a ruler, to writing option or change of the color pointer, through gestures; without the need of going to a menu and to select the options.

PALABRAS CLAVE

Dispositivo Móvil, Docente, Gestos táctiles, Pantalla, Reconocimiento de patrones, Tablero.

KEY WORDS

Movil Device, Pattern recognition, Screen, Teacher, Touch gestures, Whiteboard,

INTRODUCCIÓN

Las pantallas táctiles son reconocidas y usadas globalmente, ellas permiten la manipulación de dispositivos a través de gestos “manuales” naturales (Kumar, 2015). Es así como cada vez más resulta común encontrar este tipo de tecnologías en el hogar, el trabajo y actualmente, en la escuela. Por lo tanto, se asumiría que los docentes innovan con el objetivo de mejorar la calidad de aprendizaje de los estudiantes (Marcelo, 2013). Sin embargo, muchas herramientas digitales incorporadas en el aula pierden su utilidad al tornarse complejas o simplemente, incrementando las tareas para los docentes y siendo que “el mayor potencial que las Nuevas Tecnologías de la Información y la Comunicación (NTIC) viene derivado de las capacidades de manipulación, almacenamiento y distribución de la información de una manera fácil, rápida y accesible para todas las personas”, no cumplen con el objetivo de la tecnología. No obstante, “la popularidad de las pantallas táctiles aumenta y el comportamiento del reconocimiento táctil se vuelve cada vez más y más importante” (Abdulkhakim, 2016).

Se reconoce que algunos usuarios tienen dificultades con la incorporación del uso de gestos en *smartphones* debido a la falta de habilidades o conocimientos básicos (Sato. et al, 2016). En este punto, surge la necesidad de crear una aplicación que funcione como tablero digital e implemente reconocimiento de gestos táctiles, facilitándole al usuario herramientas de dibujo y escritura que aporten a su usabilidad, todo ello en un entorno educativo. La intención es ayudar a los usuarios a adaptarse a un instrumento de tipo tecnológico, basado en patrones táctiles naturales, como si estuvieran escribiendo o dibujando con un elemento clásico.

En vista de que el reconocimiento de gestos aporta a la sencillez en la interacción entre el usuario y el sistema (Rautaray et al, 2012), se convierte entonces en uno de los tópicos a desarrollar en este documento. Además una solución que se base en ello y a pesar de que existen muchos tableros digitales e interactivos en el mercado, es económico en cuanto a recursos: una Tablet o Smartphone con sistema Android 4.4 o superior, un televisor y un cable HDMI, como se comentará más adelante. Pero ¿por qué digitalizar un tablero? Porque las pizarras digitales colaboran tanto a los docentes como a los estudiantes a través de la interacción creativa y productiva para las dos partes (Marzano, 2009).

Así, este documento describe inicialmente la propuesta BoardApp y su justificación como herramienta TIC en aulas de clase. Luego se muestra su diseño y desarrollo, y como Matlab (Software Matemático) se utilizó para apoyar el proceso de reconocimiento de gestos de que hace la aplicación por medio de la herramienta *Classification Learner*. Se procede entonces con el planteamiento del problema de investigación, el cual es el eje principal de este trabajo.

Planteamiento del Problema

En países en desarrollo como Colombia, se está evidenciando una grieta tecnológica denominada “analfabetismo digital”, la cual se presenta porque a pesar de que la tecnología avanza constantemente, los conocimientos y uso por parte de los usuarios se limita por diversas razones, una de ellas son los altos costos que la misma representa (Palma, 2011). Por otro lado, algunos docentes con metodologías tradicionales consideran que la tecnología es muy compleja y les es difícil integrarla a sus clases (UNESCO, 2004).

Como se mencionó en la sección anterior, se requiere de herramientas de bajo costo comparadas con la implementación de un aula interactiva por ejemplo, además de permitir ser usada de forma intuitiva dada la capacidad de reconocimiento de gestos que se puede agregar a una aplicación, y así realizar ciertas tareas de manera básica. Siendo esto posible, la navegación puede dejar por fuera la tradicional forma WIMP (Window-Icon-Menu_Pointer); Un ejemplo de estas funcionalidades es la posibilidad de cambiar el color de escritura con un movimiento, usar “regla y compás” virtuales para dibujar al realizar determinada acción con los dedos sobre la pantalla, hacer *scroll*, *zoom*, etc.

No se puede olvidar que “el uso de las TICs en la educación se está centrando casi exclusivamente en la enseñanza, fomentando el aprendizaje significativo a través de herramientas tecnológicas” (Ahedo, Danvila, n.d.). Convertir entonces un dispositivo de uso diario en una herramienta para enseñar y mostrar contenidos en el aula de clases, siendo amigable con los nuevos usuarios que posiblemente no tienen una gran habilidad en su manejo, propone el interrogante: ¿Cómo implementar el reconocimiento de gestos en una aplicación, para posibilitar al usuario herramientas necesarias para dibujar y escribir en una pizarra digital?

Justificación

El uso de tecnología de “punta” es generalmente costoso (Sabogal & Velásquez, 2014) y compleja en su uso, creando así una brecha entre la tecnología y la educación. Adicionalmente, dentro de la tabla de enfermedades laborales típicas establecida por el Ministerio de Educación Nacional (decreto 1655 de 2015) se encuentran varias dolencias relacionadas con el movimiento repetitivo, cargas físicas, riesgo musculo esquelético o vibración del cuerpo, generadas por el uso frecuente del tablero convencional. De esta manera, nace esta propuesta que busca encontrar un punto de encuentro entre la tecnología y la educación, con beneficios tanto para los docentes como para los estudiantes.

En primera instancia, se presenta la necesidad de hallar una herramienta que se adapte a las condiciones del momento. Dado que los dispositivos con pantallas táctiles están presentes en diversas partes: el trabajo, el hogar, la escuela, la universidad, etc., se convierte en ventaja el disponer de sensores para la detección de gestos y el reconocimiento de actividades humanas (Farooq, 2014). Adicionalmente, se apoya la idea de Harrison (n.d.), que propone diseñar aplicaciones que reconozcan gestos para así facilitar su uso, debido a la naturalidad que implican.

De este modo, los profesores con estrategias metodológicas tradicionales, pueden percibir la tecnología como una herramienta útil y fácil de utilizar (Fernández, 2014). Así mismo, Fakuda (2008) indica que las pantallas táctiles o paneles *touch* tienen la ventaja de ser de fácil uso por ser altamente intuitivas e incluso pueden permitir la autenticación de usuarios dadas las formas en las que estos se acostumbran a usarlas.

Es así como el trabajo que se propone aquí es una herramienta didáctica que permita a docentes presentar contenidos a sus estudiantes usando la proyección de una pantalla táctil, la cual tenga la capacidad de reconocer gestos sobre su superficie y así buscar una mejora en la interacción humano-máquina (Farooq, 2014). Finalmente, la aplicación mencionada pretende aportar un instrumento asequible, que sea sencillo en su manejo por parte del usuario, sin que éste tenga que levantar su mano para escribir en un tablero tradicional y así poder reducir el esfuerzo mecánico del aparato locomotor durante el desempeño de las tareas como medida para prevenir los trastornos mencionados anteriormente (Organización Mundial de la Salud [OMS], 2004).

OBJETIVOS

Objetivo general

Crear una aplicación que funcione como tablero digital, implementando reconocimiento de gestos táctiles, para ofrecerle al usuario herramientas de dibujo y escritura en un entorno educativo.

Objetivos específicos

- Diseñar e implementar un algoritmo para el reconocimiento de gestos táctiles.
- Desarrollar una aplicación disponible para dispositivos Android, que emule un tablero convencional con herramientas de dibujo y escritura para el usuario.
- Desarrollar una herramienta tecnológica que permita grabar el contenido consignado en una pizarra y el audio, para su posterior distribución y/o reproducción.

REVISIÓN DE LITERATURA / ANTECEDENTES

Tableros digitales

En esta sección se busca introducir una breve historia de los tableros digitales y cómo se han tornado útiles para la educación en el país. Rodríguez (2014) comenta que la empresa *Smart Technologies* fue la primera en fabricar la pizarra digital interactiva en el año 1991, ello dio paso a un desarrollo investigativo constante en esta disciplina. Así mismo, las pizarras interactivas o tableros digitales han tenido cambios que se han basado en las necesidades de cada población particular.

Por otro lado, este autor clasifica en su trabajo tres tipos de pizarras interactivas: tableros con tecnología de infrarrojos y ultrasonidos, tableros pasivos y tableros activos (Ilustración 1).



Ilustración 1. Tipos de tableros digitales¹

Actualmente, la mayoría de estos tableros interactivos presentan un alto costo, aunque generan una ayuda relevante para el aprendizaje de los estudiantes. Sabogal y Velásquez (2014) en *Implementación de un Tablero Digital de Bajo Costo* exponen el proceso y unidades necesarias para la construcción y uso práctico de un tablero digital de bajo costo. Esta investigación se llevó a cabo con la ayuda de un computador con sistema operativo Windows, un apuntador infrarrojo y el control remoto de un Wii de Nintendo. Se evidencia igualmente que se requieren de herramientas adicionales a las que usualmente se encuentran en un aula de clases moderna para su implementación.

Finalmente, y contextualizando este proyecto al ambiente de Colombia, se destaca la necesidad de reducir las grietas tecnológicas existentes en diferentes áreas educativas y una

¹ Ilustración elaborada por el autor de este documento con base en la información obtenida de: BECTA, I. Advice (2004) Getting the most from your Interactive Whiteboard, 7. y Rodríguez A.V. (2014). Consideraciones y lineamientos para fomentar el uso de los tableros digitales interactivos en los ambientes de aprendizajes educativos en los municipios de Medellín e Itagüí, 18.

de las maneras de hacerlo, es introducir las tecnologías de la información y la comunicación (TIC) como un medio común, por ejemplo, con tableros digitales, los cuales permiten que el docente comunique sus conocimientos de forma más natural e intuitiva (Sabogal y Velásquez, 2014).

Reconocimiento de gestos

El acceso tecnológico en diferentes áreas crece rápidamente y esto se demuestra en las múltiples alternativas de interfaces para los usuarios, que día a día cobran interés en investigaciones afines (Dittmar, 2015). Entre ellas, se encuentran el uso de reconocimiento de gestos, reconocimiento de habla, interfaces adaptables, entre otras (Reddy R. 1996). Este proyecto tiene como eje principal el uso de reconocimiento de gestos, ello le facilitará al usuario herramientas de fácil acceso para el desarrollo de una clase.

Con respecto al reconocimiento de gestos, Myers (2001) ofrece un eje cronológico del avance en términos investigativos y comerciales de este tópico (Ilustración 2). Allí, se especifica que Teitelman (1964) desarrolló el primer reconocedor de gestos manejable, mientras que Tom Ellis's presentaba una demostración del reconocimiento de gestos en el mismo año con una tableta Rand (1964, financiada por ARPA). Finalmente, los estudios continúan con editores de texto basados en gestos (1980) y en donde se inicia un sistema comercial que hace uso del reconocimiento de gestos desde 1970 (p.49).

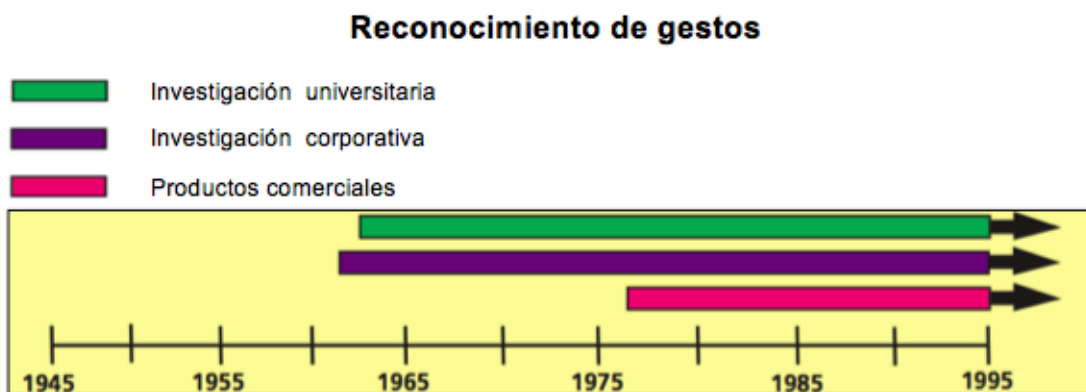


Ilustración 2. Eje cronológico de uso e investigación de reconocimiento de gestos².

El reconocimiento de gestos asiste a la naturalidad deseada para la interacción humano-computadora (Rautaray Et al, 2012). Así lo manifiesta también Baudel T. y Beaudouin-Lafon M. (1993) en su estudio *Control remoto de objetos utilizando gestos con manos libres* (título en idioma original: *Remote control of objects using free-hand gestures*). Los autores implementan un control remoto de computador utilizando gestos de manos y haciendo uso de un guante VPL que se encuentra ligado al mismo. Este guante permite el reconocimiento

² Tomado, modificado y traducido de: Myers, B. (2001). A Brief History of Human Computer Interaction Technology. Interactions. 46.

de la flexibilidad de los dedos y la posición de la mano, para ejecutar acciones en un documento como: cambiar de *página o capítulo: siguiente o anterior, subrayar ideas*, entre otras funciones. Por su parte, este proyecto busca que se haga un reconocimiento similar, pero sin hacer uso de medios externos como guantes digitales, considerado como desventaja dentro de las mismas conclusiones del trabajo mencionado.

Estudios relacionados:

TIC's y Educación

La UNESCO (2005) reconoce que el uso de TIC's aporta en gran medida a la consecución de las metas propuestas, destacándose que estos elementos tecnológicos “son motores del crecimiento e instrumentos para el empoderamiento de las personas, que tienen ondas repercusiones en la evolución y el mejoramiento de la educación”. Por lo que de esta forma el aprendizaje de su uso y su uso para el aprendizaje se han convertido en necesidades sociales de cada nación.

En Colombia inicialmente, diversas instituciones de educación formal que poseían equipos informáticos, los utilizaban con la finalidad de adquirir y enviar información, y/o en la mayoría de los casos, para enseñar y desarrollar técnicas de oficina, dejándose casi de lado la posibilidad que tienen estas herramientas para facilitar el proceso de educación, esto, se podría afirmar, fue producto del poco conocimiento por parte del educador de este medio tan sofisticado. La gran mayoría de educadores no conocen aún las herramientas y ventajas que las TIC's aportan al proceso educativo, otros tal vez no se atreven a “tocarlas” por miedo a errores en el proceso de enseñanza que dejen permanentemente vacíos conceptuales en el educando. En palabras de Galvis (1998, pág. 135) “La insipiente preparación de los docentes en informática educativa, entre otras cosas, han hecho que predominen otros usos educativos del computador”.

Actualmente estas actitudes han cambiado por parte de los educadores, incluso su formación incluye cursos que les permiten conocer las TIC's y utilizarlas adecuadamente, además de la cantidad creciente de artículos especializados en la Web que aportan al conocimiento y aplicación del computador y otras herramientas tecnológicas en el aula de clases. En Colombia se creó la página www.eduteka.org, especializada en artículos de interés sobre estos tópicos. Además de prestar asesoría académica a docentes y estudiantes sobre el adecuado uso de herramientas como la Web, calculadoras graficadoras, celulares, entre otros catalogados como TIC's.

Todo ello evidentemente porque presenta beneficios importantes al proceso de enseñanza y aprendizaje, García (2003) menciona que el uso de materiales informáticos trae como ventajas: “motivación por las tareas académicas, continua actividad intelectual, desarrollo de la iniciativa, aprendizaje a partir de errores, actividades cooperativas, individualización y aprendizaje autónomo”.

Según la UNESCO (2005) hay tres enfoques que han tomado los diferentes gobiernos para que a través de la educación se dé el desarrollo social que cada nación necesita, estos

enfoques son: “Nociones básicas de TIC, profundización del conocimiento y generación del conocimiento”. Por lo que cualquier software educativo, incorporado por los currículos nacionales, debería tener en cuenta el desarrollo en el estudiante tanto de habilidades computacionales, como la generación de la capacidad para construir conocimiento, profundizar y generar nuevo.

Aquí se deben recalcar los elementos “del sistema de enseñanza: alumno, currículo, profesor...” (Salinas, 2004) que son afectados directamente por la implementación de TICs en el aula de clases, además de que modifica el aprendizaje, entendido como “el proceso de construcción de significados y de atribución de sentido a los contenidos” (Coll, 2004). Coll (2004) indica por ejemplo que la presencia del término *e-aprendizaje*, (“interpretado como la utilización de las nuevas tecnologías multimedia e internet con el fin de promover y mejorar la calidad del aprendizaje”), muestra que efectivamente hay un cambio en cuanto a educación se refiere, si se habla del uso de TICs para la enseñanza.

Lo expuesto hasta el momento puede ser peligroso en cuanto da pie a pensar “que materiales altamente sofisticados proporcionan la calidad” (Salinas, 2004). Allí Coll (2004) y Salinas (2004) concuerdan en que para el uso de TIC’s en educación debe haber un “proceso de innovación pedagógica”, en palabras de Salinas: “cualquier proyecto que implique la utilización de TIC’s, implica cambios metodológicos, de formación de docentes...”. Coll por su parte menciona: “La incorporación de TIC’s a los procesos formales y escolares de enseñanza y aprendizaje pueden modificar (...) las prácticas educativas”, ello incluye modificación de estrategias didácticas, modelos metodológicos y por ende modelos pedagógicos.

Tableros digitales y reconocimiento de gestos

La interacción humano-computadora se ha vuelto altamente popular (Karray, et al. 2008), y tomando en consideración este avance, para el caso de tableros digitales y reconocimiento de gestos, se pueden destacar diferentes trabajos que han contribuido en el ámbito tecnológico y educativo, y que a su vez, registran relación con este proyecto.

Xuesong, Xin, y Qin (2015) muestran con su investigación, cómo los tableros electrónicos permiten llegar de forma acertada a los estudiantes de carreras relacionadas con el diseño gráfico. Los paneles táctiles les permiten manipular estructuras en tiempo real. Por otro lado, se encuentran estudios con diversos alcances y aplicaciones en cuanto al tópico de reconocimiento de gestos, como el expuesto por Shahzad, Liu, y Samuel (2017) “que proponen, implementan y evalúan un ambiente de gestos y firmas basado en un esquema de autenticación de dispositivos táctiles” (p. 26).

Teniendo en cuenta que los equipos electrónicos están inmersos en el diario vivir, Guarneri, Capra, Farinella, Cristaldi, y Battiato (2013) proponen un sistema de reconocimiento de formas sobre pantallas táctiles, ello con la intención de formular soluciones económicas en cuanto a recursos de computo, necesarios en los dispositivos móviles que, como se menciona anteriormente, cada día impregnan más la cotidianidad de las personas. Aportes en torno a este tópico constituyen una gran ayuda para el diseño de diversas aplicaciones.

Hsu, Cheng, y Yu (2014) proponen un algoritmo de reconocimiento de gestos basado en ambas manos, utilizando además la información del acelerómetro triaxial y la pantalla táctil de un dispositivo móvil. A diferencia, el presente proyecto no posee una identificación de mano izquierda o derecha, lo que quiere decir que si se realiza el gesto correspondiente a la respuesta esperada, no se requiere que se haga con una mano específica, generando comodidad para los usuarios pero perdiendo un poco la diversidad gestual que se podría tener con más descriptores. Ello por el momento en términos de practicidad y simplicidad. Otro ejemplo del uso de varios descriptores lo propone *Ma K. , Li J. , Han J., Hu Y. and Xu W (n.d.) quienes* exponen un método para mejorar la forma de reconocer información por parte de la pantalla, haciendo que cada punto tocado tenga una propiedad o significado según el dedo que lo genere.

Reconocimiento de gestos y tableros digitales en educación

Teniendo en cuenta las relaciones entre el reconocimiento de gestos y los tableros digitales, se refiere ahora al aporte de estos en el campo educativo. Harrison (n.d.) propone el diseño de aplicaciones que implementen este tipo de tecnologías para posibilitar su uso, ello dada la facilidad y naturalidad con que los usuarios se adaptan a éstas. Así las tecnologías táctiles pueden contribuir a resolver necesidades educativas utilizándolas como herramientas en el aula de clases (Salinas, 2004).

Marzano (2009) es otro de los autores que apoya el uso de tableros digitales como ayuda para que los estudiantes alcancen sus objetivos educativos. El nivel de interacción que proveen y las herramientas que aportan tanto para el docente como para ellos, contribuyen enormemente al desarrollo de las clases de forma interactiva, significativa para el estudiante y productiva para el profesorado.

Por último, Rodríguez (2014) realiza un aporte importante a este proyecto ya que se acerca a la realidad colombiana, al exponer factores críticos que afectan la educación en el país, para ello propone “recopilar, indagar, analizar y sistematizar distintas posibilidades didácticas, buenas prácticas, métodos de enseñanza y aprendizaje, estrategias de formación docente, problemas e inconvenientes del tablero que permitan generar unos lineamientos encaminados a potencializar esta tecnología y propiciar su uso” (p.13). Concluyendo así que los docentes poseen una buena aceptación frente a los tableros interactivos, ya que apoyan sus prácticas pedagógicas, permitiendo la autonomía del docente.

Lo presentado anteriormente pretende resaltar la importancia de este trabajo como aporte en diversos campos, desde lo tecnológico a lo educativo. Se inició con un reconocimiento histórico, que permite entender con brevedad los avances en tecnologías relacionadas con el reconocimiento de gestos, seguido de una contextualización en áreas educativas, desde bajo costo hasta practicidad. Se busca también, implementar algoritmos que dependiendo de su desempeño, puedan ser objeto de nuevas investigaciones y contribuyan a mejorar el desempeño de las aplicaciones en dispositivos móviles en futuros trabajos.

ESTRATEGIA METODOLÓGICA

La construcción de un software como se hizo en este trabajo requiere desde la ingeniería, toda una serie de procesos y acciones como la selección de una metodología de desarrollo, conocimiento y aplicación de tópicos de ingeniería de requisitos, entre otros (Pressman, 2005). Aquí se usó una metodología propia que integra principios y fases de diferentes metodologías estándar, pero que no sigue alguna en particular, esto por haber sido desarrollado de forma individual.

Por ejemplo, se mezclaron etapas de la metodología en cascada: Análisis de requisitos, diseño del programa, codificación y ejecución de pruebas, con elementos del método por prototipos y el modelo espiral (Pressman, 2005). La siguiente tabla muestra las etapas que se desarrollaron y sus respectivos entregables:

Etapas	Descripción	Entregable
1. Planteamiento del Problema	Se estableció inicialmente el problema a resolver con la solución TI.	- Anteproyecto de Grado con Objetivos y problema a resolver.
2. Construcción de Requisitos	Elaboración y recolección de requisitos propuesto por el asesor temático y el investigador.	- Listado de requisitos funcionales y no funcionales.
3. Diseño de prototipo para recolección de datos	Por ser un problema de reconocimiento de patrones, se requería de un conjunto variado de datos para así generar un clasificador adecuado.	- App que permitiere la recolección de los siguientes datos por gesto y dedo: Dedo usado, posición X sobre la pantalla, posición Y sobre la pantalla y tiempo desde el inicio del gesto.
4. Proceso de reconocimiento de patrones	Etapas para definición e implementación del clasificador. Éstas se especifican más adelante.	- Código del clasificador.
5. Diseño de la aplicación	Elaboración de Diagrama de Clases y Algoritmos para la ejecución de tareas.	- Diagrama de clases
6. Codificación	Escritura del código para la construcción de la aplicación.	- Aplicación BoardApp Preliminar

7. Ejecución de Pruebas	Prueba de diferentes prototipos, con sus funcionalidades. Retroalimentación para conseguir una versión final de BoardApp.	- Aplicación BoardApp Final
-------------------------	---	-----------------------------

Tabla 1. Metodología de Desarrollo BoardApp

Así se describen en la siguiente sección los procesos y etapas mencionados, omitiendo la primera que ya tuvo su desarrollo en las primeras secciones de este documento.

DESARROLLO E IMPLEMENTACIÓN

Construcción de Requisitos

Como se mencionó en la sección anterior, se iniciará mostrando el desarrollo de cada una de las etapas dadas en la metodología. Primero la elaboración de requisitos, la cual fue un proceso llevado a cabo por la asesora temática y el investigador de este documento, dado el problema de investigación se formularon las características *Funcionales* y *No Funcionales* de BoardApp, se destacan los siguientes:

Requisitos Funcionales

La aplicación BoardApp debe:

1. Permitir que se dibuje sobre la pantalla, accediendo a esta opción a través de un gesto.
2. Permitir grabar el audio de una sesión de clases para luego ser exportado: guardado en la memoria interna del dispositivo.
3. Permitir grabar lo dibujado sobre la pantalla, junto con el audio, en formato de video, para guardado en la memoria interna del dispositivo.
4. Permitir cambiar el color de escritura con un gesto.
5. Permitir el acceso a la regla virtual, por medio de un gesto, para trazar líneas rectas.
6. Permitir que se quite la regla de la pantalla ejecutando algún gesto o acción.
7. Permitir usar un borrador para eliminar las zonas escritas que se desean eliminar, accediéndolo a través de un gesto.
8. Permitir que se deje de borrar una parte del contenido de la pantalla realizando algún gesto o acción.
9. Permitir borrar todo el contenido de la pantalla con un gesto.
10. Permitir dibujar circunferencias, reconociendo cuando el usuario desee dibujar una circunferencia por medio de un gesto.
11. Permitir modificar el tamaño y centro de las circunferencias antes de que queden plasmadas en la pantalla.
12. Permitir tomar *Screenshots* por medio de un gesto.
13. Guardar los *ScreenShots* en una carpeta del almacenamiento interno del dispositivo.

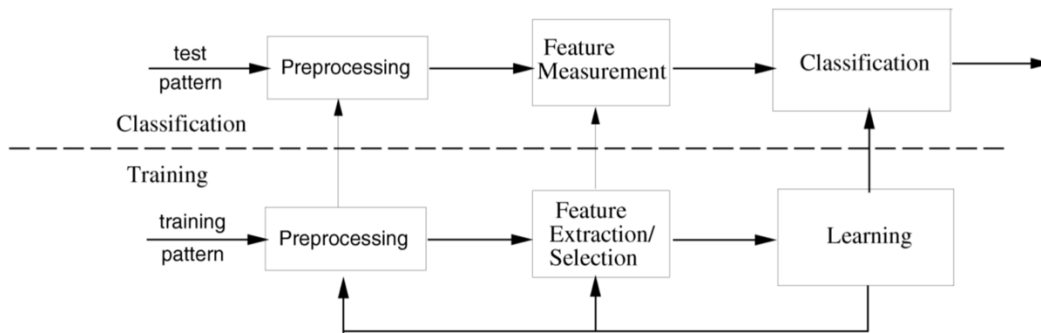
14. Permitir la reproducción de un video introductorio con los gestos que reconoce la aplicación.

Requisitos No Funcionales

La aplicación BoardApp debe:

1. Ejecutarse en dispositivos con sistema operativo Android.
2. Tener como pantalla principal una que sea de fondo blanco para escribir sobre ella.
3. Usarse únicamente de forma horizontal (*Landscape*).
4. Disponer de opciones para permitir al usuario omitir el video introductorio.

Después de establecidos, se definió como eje central del proyecto el *reconocimiento de gestos* a través del reconocimiento de patrones. Para lograr que un dispositivo electrónico (en este caso una tableta o *smartphone*) logre esta tarea, se desarrollaron las etapas de: captura de datos, pre-procesamiento de los datos, selección del algoritmo o herramienta de clasificación (Clasificador), el proceso de aprendizaje de la herramienta, implementación de la herramienta y pruebas de validación (Proceso de Clasificación), descrito por Jain (2000) y que se resume en la siguiente ilustración:



*Ilustración 3. Procesos para conseguir el reconocimiento de gestos.*³

A continuación, se describirá de los procesos mencionados los que se siguieron para el reconocimiento de gestos. Primero, se especifica cómo se realizó la captura de datos y qué tipos de datos se capturaron, siendo ello lo correspondiente a la etapa 3 de la metodología (Tabla 1).

³ Ilustración tomada de: Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence, 22(1), 4-37.

Captura de datos

Para llegar al reconocimiento de un gesto se requiere de un conjunto de datos, inicialmente unos de *entrenamiento*, otros de *validación* y los propios que se producirán en la aplicación en tiempo de ejecución. Los dos primeros se deben obtener para comenzar con el diseño y desarrollo del clasificador (Jain, 2000), y ellos son los que se describirán en esta sección.

Se definieron primero los gestos que la aplicación reconocería (Anexo 1) y luego, para garantizar la variedad en los datos, que se recolectarían un total de 60 muestras, cada una realizada por un usuario diferente y en donde éste haría los 17 gestos propuestos. Aquí se hizo necesario crear una aplicación con la que el usuario pudiese interactuar y capturar los datos necesarios.

La aplicación para la recolección de los datos se desarrolló en Android Studio, llamada “BoardApp1” es posible instalarla en tabletas y *smart phones*. Ella inicialmente indica al usuario su finalidad, luego muestra un video con el gesto y posteriormente le permite al usuario replicarlo (Ilustración 1). Al realizarse un gesto sobre la superficie del dispositivo la app captura la cantidad de dedos sobre la pantalla, identifica cada dedo asignándole un valor de 0 a 9, aunque los gestos solo requieren de máximo 5 dedos, y de cada uno de ellos su posición x y y . A la vez se guarda el tiempo desde iniciado el gesto hasta su finalización.

Para lograr reconocer el inicio y finalización del gesto, Android incluye funciones que permiten detectar ello:

```
MotionEvent.ACTION_DOWN
```

```
MotionEvent.ACTION_POINTER_DOWN
```

```
MotionEvent.ACTION_UP
```

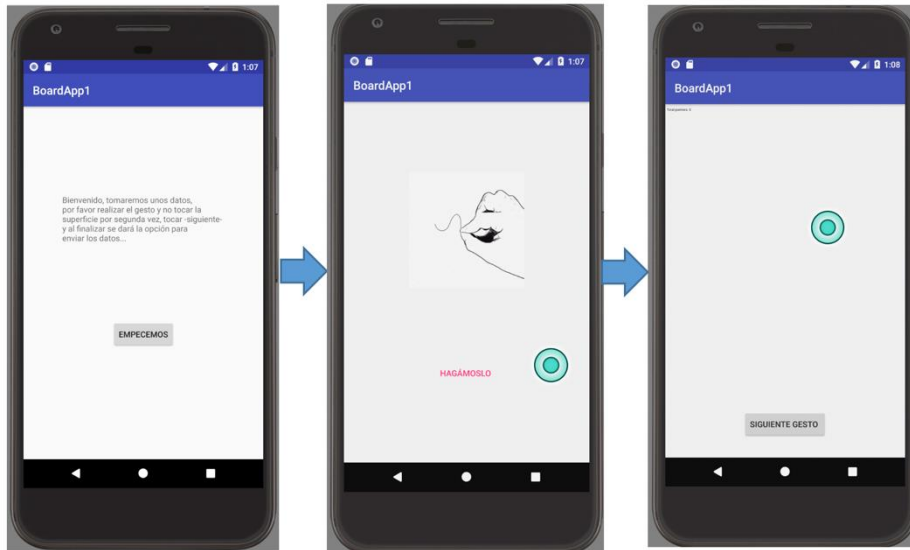


Ilustración 4. ScreenShots de BoardApp1.

Cada conjunto de datos generado por un usuario realizando un gesto se denominó: *Ejemplo*, que como se mencionó se recolectaron un total de 60 ejemplos por gesto. Cada gesto fue guardado como un grupo de 4 datos: Identificador de dedo, componente x de la posición del dedo, componente y de la posición del dedo y tiempo desde que se inició el gesto hasta que se capturaron los datos. BoardApp1 almacena lo recolectado en un archivo .txt en la memoria del dispositivo, pero además al finalizar, le permite al usuario enviar dicho archivo por correo electrónico al interesado en él.

De esta manera, al reunir la cantidad de archivos .txt anteriormente mencionados, fue necesaria la ayuda de Microsoft Excel y su capacidad de ejecutar macros, así en un solo .xlsx se agruparon todos los datos. Sin embargo la información aún debe ser adecuada para propender por la *Invarianza*, comentada por Duda (1973), antes de ser usada para determinar el clasificador, así en la sección siguiente se explica el proceso de pre-procesamiento, y que constituye el inicio de la etapa 4 de la metodología (Tabla 1.)

Pre-procesamiento

Los datos extraídos pueden variar en cuanto al formato de pantalla, posición del dispositivo (vertical u horizontal), tamaño del gesto con relación al tamaño de la superficie táctil, entre otros, y pueden afectar negativamente el desempeño del clasificador que se seleccionara. Por lo que primero se hizo la separación de cada gesto en hojas de cálculo diferentes, y allí se procedió con la normalización de la información, todo ello con macros dado que el volumen de datos era muy grande para hacerse a mano.

De cuatro variables: *Dedo*, *Posición x* , *Posición y* , *Tiempo*, las influenciadas mayormente por los diferentes escenarios en los que se tomaron, son las posiciones x y y de cada dedo, dado que la primera es de tipo nominal (no se normaliza), mientras que el

tiempo se tiene en cuenta en otras características que más adelante se comentan. Así que se procedió con la normalización de ellas dos, tomando de cada componente el valor máximo: $x_{máx}$, $y_{máx}$, y mínimo: $x_{mín}$, $y_{mín}$, luego tomando la razón entre la diferencia de cada valor con el mínimo correspondiente, y el máximo correspondiente.

$$x_{i-Normalizado} = \frac{x_i - x_{mín}}{x_{máx}}$$

$$y_{i-Normalizado} = \frac{y_i - y_{mín}}{y_{máx}}$$

De esta manera los valores de estas dos variables quedan como si se hubiesen tomado en una única venta de dimensiones 1×1 .

Con ello hecho, se procedió a definir las características de los datos obtenidos, lo cual se logra de diversas formas, usando por ejemplo propiedades estadísticas y métodos como PCA (Jain, 2000), sin embargo se usaron algoritmos propios para obtener información a partir de los datos, es decir que se realizaría una *Selección de Características* y no una Extracción de ellas (Fernández, 2015). Así se estableció que se tomarían:

- Velocidad promedio por componente x y y separadamente (2 características) y por cada dedo (2 x 5 dedos = 10 características).
- Aceleración promedio por componente y dedo (2 x 5 dedos = 10 características).
- Velocidad resultante: $v = \sqrt{v_x^2 + v_y^2}$ por cada dedo (5 características).
- Aceleración resultante: $a = \sqrt{a_x^2 + a_y^2}$ por cada dedo (5 características).
- Cantidad de dedos usados (1 característica).
- Curvatura (Larson, 1999): $K = \frac{|a|}{(1+v^2)^{3/2}}$ por cada dedo (5 características).
- Cambios de dirección por cada componente y cada dedo (2 x 5 dedos = 10 características).
- Etiqueta (1 para gesto 1, 2 para gesto 2,...),

Con lo que se tiene un total de 47 características por cada ejemplo.

Obtener esta información y así generar su vector representativo de 47 componentes requirió de la implementación de macros, tanto para organizar como para calcular. Con esto se procedió a la etapa de Aprendizaje, que a continuación se describe.

Aprendizaje

Para la clasificación de patrones existen diversos algoritmos y metodologías (Jain, 2000), probar todas puede resultar un trabajo tedioso para los investigadores, pero permitiría obtener el clasificador más adecuado para el problema (Duda, 1973). En este sentido hay software que hace diferentes pruebas con varios tipos de clasificadores, se utilizó para este trabajo Matlab, el cuál incluye un módulo llamado Classification Learner. Éste permite

importar los datos desde un archivo de texto o una tabla de Excel, luego seleccionar las columnas que contienen descriptores y el resultado de cada ejemplo:

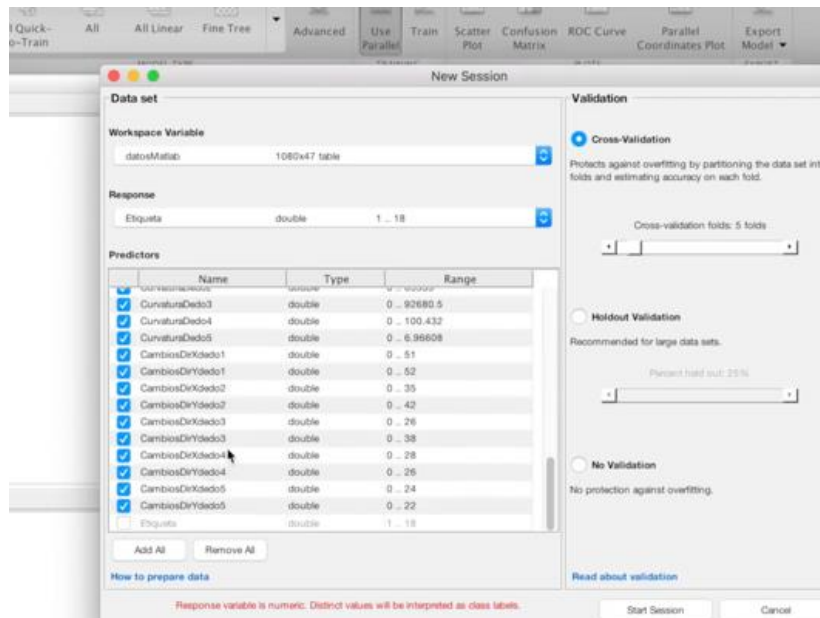


Ilustración 5. Importación de datos con Classification Learner de Matlab.

Allí también se puede indicar la cantidad de ejemplos para entrenar los clasificadores y la cantidad que se usará para la validación. Se tomó 70% para entrenamiento y 30% para ser clasificados y validar el modelo.

Posterior a ello se permite seleccionar el clasificador deseado, o como se hizo para el proyecto, todos los clasificadores:

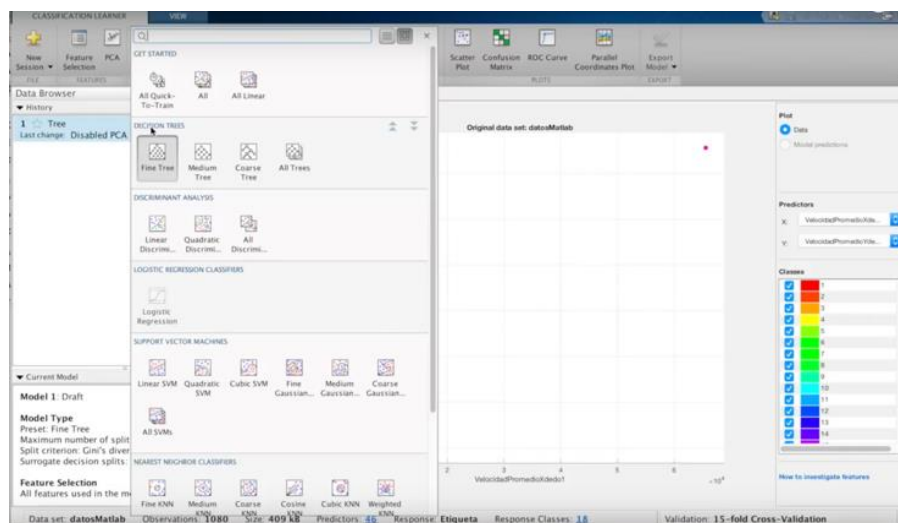


Ilustración 6. Selector de clasificadores en Classification Learner de Matlab.

Los resultados obtenidos se presentan al usuario y ello permitió escoger el tipo de clasificador con mejor desempeño para el proyecto, además se muestra información relevante como la matriz de confusión y la curva ROC⁴ por cada uno:

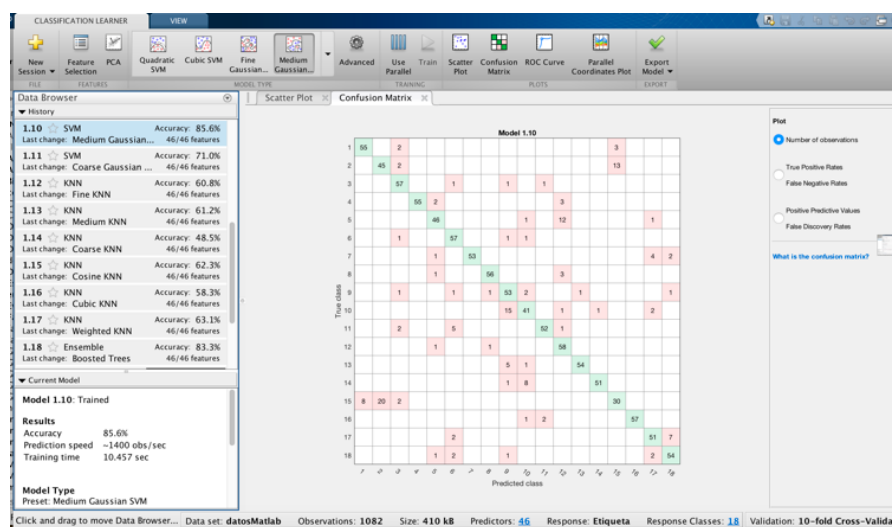


Ilustración 7. Classification Learner de Matlab mostrando resultados del entrenamiento de diversos clasificadores.

Como factor de decisión se tuvo en cuenta la exactitud de cada clasificador dada también por Matlab y cuyo cálculo se realiza de la siguiente manera:

$$\text{Exactitud} = \frac{\sum \text{Verdaderos Positivos} + \sum \text{Verdaderos Negativos}}{\sum \text{Población Total}}$$

Los resultados obtenidos con todo el conjunto de datos fueron los siguientes:

Clasificador	Algoritmo	Exactitud
Discriminantes	Discriminante Lineal	82,1%
	Discriminante Cuadrático	-
Árboles de Decisión	Árbol Simple	90,5%
	Árbol Medio	70,2%
	Árbol Complejo	27,6%
SVM	SVM Lineal	81,1%
	SVM Cuadrático	83,1%
	SVM Cúbico	82,2%
	SVM Gaussiano Fino	85,6%
	SVM Gaussiano Medio	85,6%

⁴ Cerda, Jaime, & Cifuentes, Lorena. (2012). Uso de curvas ROC en investigación clínica: Aspectos teórico-prácticos. *Revista chilena de infectología*, 29(2), 138-141. <https://dx.doi.org/10.4067/S0716-10182012000200003>

	SVM Gaussiano Grueso	71,0%
KNN	KNN Fino	60,8%
	KNN Medio	61,2%
	KNN Grueso	48,5%
	KNN Coseno	62,3%
	KNN Cúbico	58,3%
Compuestos	“Boosted Trees”	83,3%
	“Bagget Trees”	86,1%
	Subespacios Discriminantes	79,6%

Tabla 2. Resultados obtenidos por Classification Learner de Matlab.

Se seleccionó por sus resultados y sencillez de implementación el clasificador “Árbol Simple”, cuya exactitud es del 90,5% en la validación. Matlab también ofrece la opción de exportar el código para la aplicación:

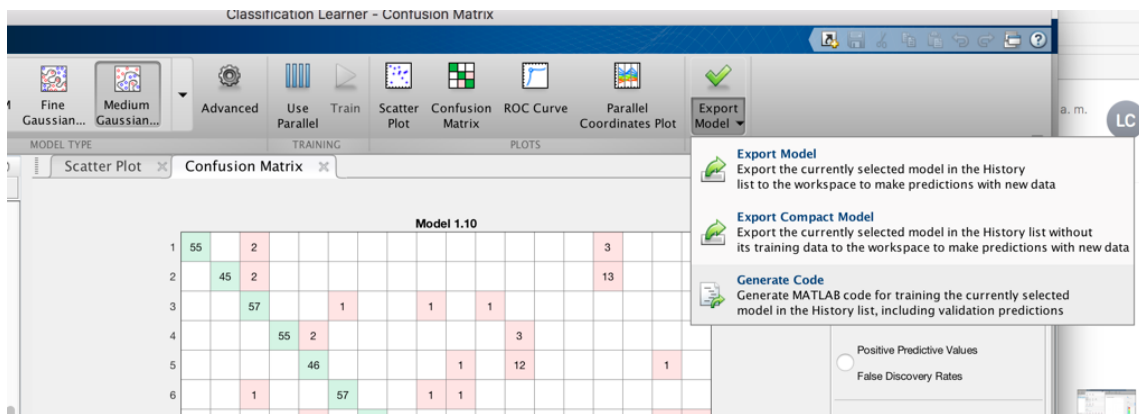


Ilustración 8.. Opciones de exportación del modelo del clasificador seleccionado en Matlab.

Aún así el árbol exportado se debió traducir a JAVA para poder ser implementado en Android. El modelo dado por Matlab se presenta en el anexo 2 y obtenerlo fue una parte fundamental del trabajo. En la sección siguiente se describe el desarrollo de las etapas 5 y 6 de la metodología (Tabla 1).

Diseño y desarrollo de la aplicación

BoardApp se diseñó usando el paradigma de programación orientada a objetos. Cada clase representa un tipo de objeto bien definido y su instanciación cumple unas funciones particulares para todo el sistema. A continuación, se presenta el diagrama de clases con la estructura del proyecto:

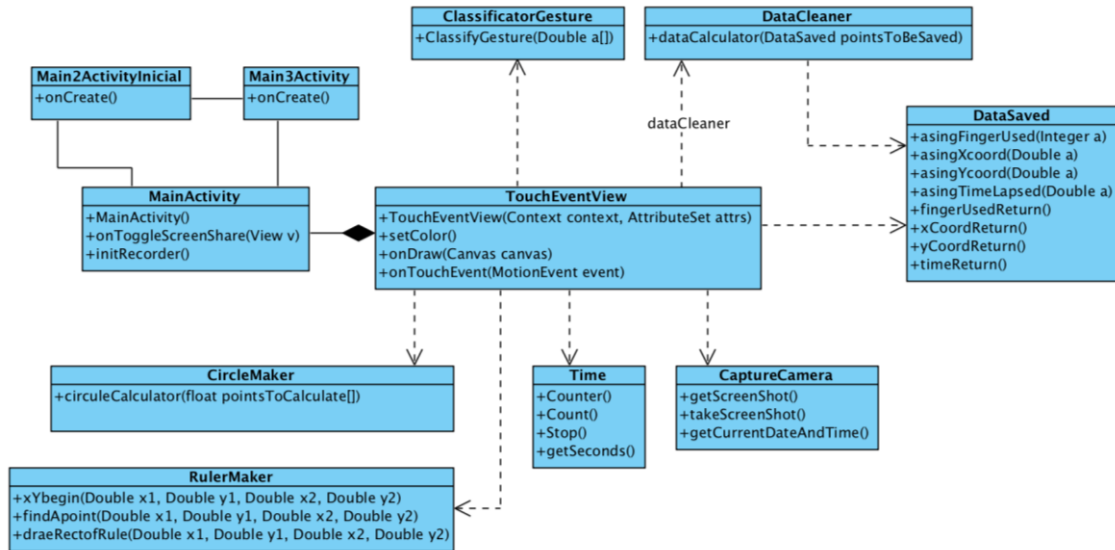


Ilustración 9. Diagrama de clases para BoardApp,

Dando respuesta a los requerimientos, se planteó que la “pizarra” de dibujo fuera un Canvas y con ello se permitiera el trazo sobre la misma, la creación de figuras y su exportación como dibujo (opción de screenshot). Sin embargo, esto representó dificultades a la hora de manipular colores, guardar lo dibujado como objetos independientes y deshacer o rehacer trazos realizados. Lo primero fue desarrollar el código necesario para el reconocimiento de gestos (Clase “ClassifierGesture”), su instanciación se hace en la clase “TouchEventView” que se encarga de toda la parte gráfica y de captura de la posición de cada dedo sobre la pantalla.

Luego, se hizo el desarrollo para que cada gesto tuviera su correspondiente reacción. Lo primero en este caso fue reconocer el gesto de escritura con dos dedos, el cual inicialmente solo podía realizarse en color negro. Al detectarse el gesto se almacenan las coordenadas del primer dedo puesto sobre la pantalla, en un arreglo de puntos llamado *Path*, de tal manera que cada vez que se actualiza refresca la pantalla y anexa los nuevos gráficos a los ya existentes.

Posteriormente se desarrollaron los algoritmos y códigos correspondientes a la regla (Clase RulerMaker) y al trazo de circunferencias (Clase CircleMaker). Para la primera fue necesario generar un rectángulo de tamaño suficientemente grande para ser visible por el usuario, a pesar de éste tener dispuestos sobre la pantalla tres dedos. Se calculó primero la posición de los dos dedos más lejanos entre sí de los tres, a partir de sus coordenadas *x*, luego se calculó la pendiente de la recta que los une, así se podrían encontrar las pendientes de las rectas paralelas y perpendiculares que formarían los lados del rectángulo (regla).

Para la circunferencia se hizo un proceso similar: se escogieron las posiciones de las parejas de dedos más lejanas entre sí, con la finalidad de que visualmente se apreciara una circunferencia hecha a partir de 5 puntos. Dadas esas posiciones se calculó la distancia entre ellas y el radio de la circunferencia, además del punto medio que serviría como centro de la misma.

Reconocidos los demás gestos se requería tan solo de algunas líneas de código que ejecutaran el borrado completo de pantalla o el cambio de color, lo que no requiere de algún algoritmo que merezca su explicación aquí. Por otro lado, los gestos: “encerrar objetos con dos dedos” (Anexo 1, gesto 6) y dibujar arco con dos dedos (Anexo 1, gesto 3) no fueron incluidos en la versión final de BoardApp dado su bajo desempeño de reconocimiento en la práctica. La matriz de confusión pese a que en el entrenamiento arrojó para estos gestos un alto índice de exactitud (Ilustración 9), en su implementación no se obtuvieron los resultados esperados, generando una exactitud máxima de apenas el 40%.

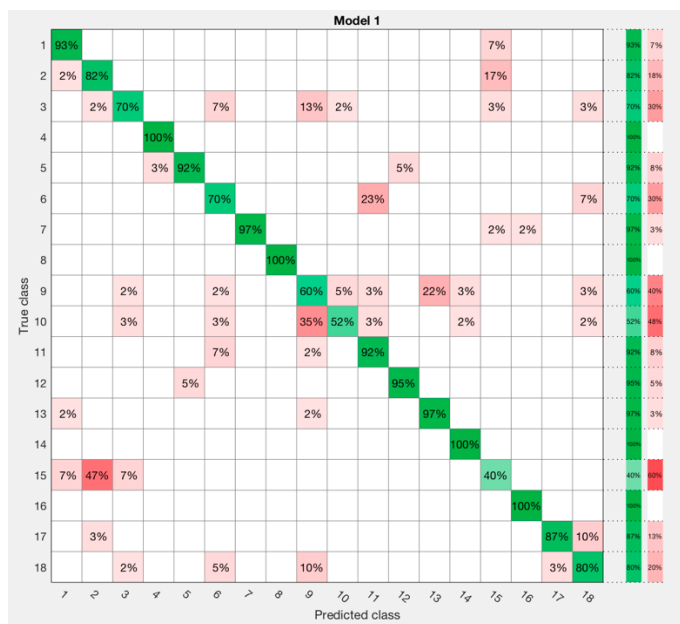



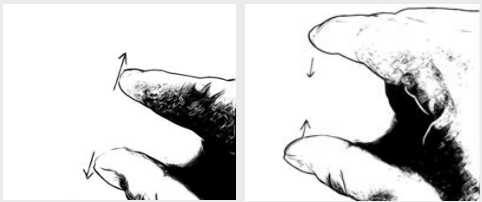
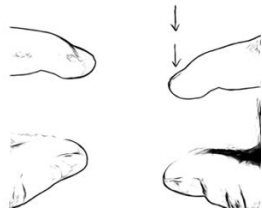


Ilustración 10. Matriz de confusión dada por Classification Learner de Matlab.

Dadas las limitaciones comentadas al inicio de esta sección sobre el atributo *path*, gestos como cambiar color de escritura y hacer zoom (Gestos 1, 9, 10, 11 y 13 del Anexo 1) tuvieron cambios en el desarrollo final de BoarApp. Esto también se evidenció en pruebas de prototipos que se fueron desarrollando (Etapa 7 de la metodología). Para algunos de ellos, su funcionalidad se modificó, otros fueron fusionados, quedando establecidos los siguientes, después de las modificaciones:

	Gesto	Descripción
1		Escribir con dos dedos: estilo bolígrafo. Después de iniciado el gesto si el usuario desea puede cambiar a un dedo. Éste estará activo hasta que se detecte otro gesto.

2		<p>Regla: Pulsar con tres dedos sin soltar, movimiento. Aparecerá un rectángulo emulando una regla sobre la pizarra, para que el usuario pueda dibujar con un cuarto dedo.</p>
3		<p>Borrador: Con cuatro dedos y mover. Por las zonas de la pantalla en donde se encuentre los cuadro dedos se “borrará” lo que haya dibujado.</p>
4		<p>Circunferencia: tocar con 5 dedos separados. Abrir o cerrar para modificar el tamaño de la circunferencia, mover para cambiar su posición.</p>
5		<p>Cambio de color: Abrir o cerrar con dos dedos. Automáticamente el color de los objetos cambiará.</p>
6		<p>Captura de pantalla: Con 4 dedos, se desliza hacia abajo el índice derecho (Obturador cámara) para tomar pantallazo. Se genera una imagen .JPEG en el directorio: Almacenamiento Local -> Images/Clases/</p>


7		<p>Borrar pizarra por completo: Con dos dedos hacer cualquiera de estos gestos: abrir y cerrar en zig-zag, con dos dedos juntos subir o bajar despacio.</p>
---	---	---

Tabla 3. Gestos reconocidos por BoardApp

En conjunto, los gestos implementados buscan, en primera instancia, cumplir con los requisitos planteados para el proyecto (aunque algunos se replantearon). En segundo lugar, proveer al usuario final herramientas de fácil uso y memorización. También se incluyó la grabación de audio y video, lo que permite cumplir con uno de los principales objetivos de la aplicación, y es que funcione como una herramienta educativa. Se crearon clases para ello y para la captura de pantalla, con la finalidad de seguir el planteamiento hecho sobre la programación orientada a objetos que se mencionó antes.

Descripción de la aplicación final

BoardApp consta de 3 *Activity* diferentes, la primera (Ilustración 10), que siempre se mostrará al usuario, lo invita a ver una “introducción” al manejo de la app, allí se presenta una animación que muestra los gestos que es capaz de reconocer el software. La segunda *Activity* (Ilustración 11) es la introducción y la tercera (Ilustración 12) es la pizarra como tal.



Ilustración 11. Activity 1 BoardApp.



Ilustración 12. Activity 2 BoardApp.



Ilustración 13. Activity 3 BoardApp.

El Activity 3 posee un botón que le permite al usuario indicar si se inicia o no la grabación de audio y video. El estado “OFF” indica que no se está grabando, el “ON” que se ha comenzado a crear el video. Al oprimir “ON” de nuevo para cambiar su estado a “OFF” se detiene la grabación y el video se almacena en: Almacenamiento Local -> Images/, en formato mp4. Allí, el usuario podrá buscar el video y compartirlo por el medio que él desee.

BoardApp reconoce cuando se inicia un gesto, allí toma el tiempo y los datos de la posición de cada dedo para luego de pasado un segundo, envíe ello a un objeto de la clase ClassifierGesture y que éste devuelva el gesto reconocido. El tiempo permite que el usuario pueda tomar una decisión de si continúa o no con el gesto. Por el momento la aplicación corre con gran fluidez (aunque también depende de las características del dispositivo). Sin embargo, hay gestos cuyo reconocimiento pueden ser mejorados.

RESULTADOS

La utilización del software Matlab fue de gran ayuda para la definición y construcción del clasificador que se implementó en BoardApp. La herramienta *Classification Learner* de este software les permite a investigadores de diversas áreas del conocimiento, realizar estudios que se basan en el reconocimiento de patrones. Dadas sus funciones, que permiten contrastar entre diferentes clasificadores por medio de su desempeño, mostrar gráficos de relaciones entre variables, calcular la matriz de confusión, exportar el modelo seleccionado a código, entre muchas otras, se decidió utilizarlo en este trabajo.

Así se logró construir el clasificador que permite realizar el reconocimiento de gestos en BoardApp. El algoritmo construido se basó en Árboles de decisión, dado el desempeño que tuvo respecto a otros clasificadores en las pruebas, además de su sencilla implementación a partir de los datos de los que se disponía. Con respecto a ellos fue necesario un trabajo considerable para normalizarlos, organizarlos y filtrarlos. Se requirió en primera instancia una app que permitiera tomar las muestras y seguidamente de macros en Microsoft Excel para calcular sobre ellos.

Dada la *Exactitud* de cada clasificador como indicador para ser seleccionado o no, y la matriz de confusión como sustento, se logró el diseño e implementación de un algoritmo para el reconocimiento de gestos táctiles.

Posterior a ello comenzó el proceso de desarrollo de BoarApp, el cual fue hecho para sistemas operativos Android 4.4 o superior, en tabletas o *smartphones*. La selección de la versión se hizo con base en dos aspectos: a) Abarcar la mayor cantidad de dispositivos en el mercado, b) Permitir el funcionamiento de todas sus características. Por ejemplo, el grabador de video solo está disponible para las versiones comentadas de Android.

Se desarrollaron algoritmos para que la aplicación actuara basada en el gesto reconocido por la clase correspondiente. Gestos como dibujar una circunferencia y usar una regla en pantalla, requirieron del manejo de coordenadas en el plano, distancias entre puntos, ecuaciones de rectas paralelas y perpendiculares, y algunas subrutinas de ordenamiento de datos. Otros gestos resultaron más sencillos de implementar dado que solo manipulaban propiedades del *Canvas* como el color, o el almacenamiento de imágenes y su eliminación de la pantalla.

Por su parte los resultados obtenidos en la implementación final exigieron que se reevaluara la pertinencia de la inclusión de algunos gestos, dado que su similitud dificultaba a BoardApp el pasar de uno a otro sin quedarse en el primero. Los porcentajes dados por la matriz de confusión en la práctica, podían crecer al dársele un uso continuo a la aplicación. Ello posiblemente porque la toma de datos fue hecha en condiciones controladas, donde no se hacían los gestos aleatoriamente sino como estaba programada la aplicación para la recolección de datos.

Esto, sin embargo, dada la gran cantidad de gestos planeados originalmente, no afectó que la aplicación reconociera los fundamentales para el trabajo del usuario final. De esta forma, se logró desarrollar una aplicación que emula un tablero convencional con herramientas de dibujo y escritura para el usuario.

Por último, BoardApp al ejecutarse en una sesión de clase, por ejemplo, desde una tableta o *Smartphone* cuya pantalla está siendo proyectada, permite grabar lo que en ella se dibuje, a la vez que el audio producido usando los micrófonos del dispositivo. Esto gracias a que Android pone a disposición de los desarrolladores (desde su versión 4.4) la posibilidad de implementar métodos en sus aplicaciones que realicen este tipo de tareas. El archivo resultante es un .mp4 que puede ser reproducido en el mismo aparato o compartido, sea por correo, redes sociales, etc.

Con esto se logró desarrollar una herramienta tecnológica que permite grabar el contenido consignado en una pizarra (BoardApp) y el audio, para su posterior distribución y/o reproducción.

DISCUSIÓN Y CONCLUSIONES

BoardApp como aplicación es asequible a todas las personas con un dispositivo Android y sistema operativo con versión superior o igual a la 4.4, lo que la hace una herramienta que potencialmente podrá ser usada por muchos a nivel global, cumpliendo así una de las tareas de la tecnología que es llegar a todo individuo que la requiera (UNESCO, 2004). Pero no solo se limita a estar disponible, sino que es software que busca llegar a usuarios de todo nivel, en lo referente al uso de dispositivos con pantallas táctiles. El hecho de presentar una interfaz sencilla y reconocer gestos, muy similares a los que se realizan en un tablero tradicional, busca ante todo una relación amigable entre el usuario (docente) y la aplicación, apoyado ello en los planteamientos de Rautaray et al (2012), sobre la necesidad de sistemas que interactúen con los humanos de una forma más natural.

Una de las grandes ventajas de la aplicación y que la diferencia de otras, es su capacidad de almacenar una clase (Audio y video), permitiéndole a docentes usar herramientas a la par, como presentaciones, páginas web, videos y siendo todo esto registrado, siempre con el apoyo de la superficie para escribir, que para todo docente o expositor le será de utilidad. Efectivamente se dará una mejor interacción entre docentes y estudiantes (Marzano, 2009).

Los docentes que se adentren en la preparación interactiva de sus clases con BoardApp, notarán grandes beneficios tanto en la respuesta de sus estudiantes (Ahedo, Danvila, n.d.) como en su propia salud. Como se comentó en la justificación de este documento, son permanentes los reportes de afectaciones a la fisionomía de docentes por el uso excesivo de tableros convencionales, debido a los movimientos repetitivos y poco naturales (como levantar un brazo por escribir a una altura diferente) que deben realizar al momento de dictar sus clases. Realmente una aplicación como la construida tiene aplicaciones ilimitadas en el aula de clases, los docentes con su creatividad e innovación son los que con su uso pueden lograr la consecución de las metas educativas sobre sus estudiantes.

En este punto se evidenció el cumplimiento de los objetivos de este trabajo, BoardApp es una aplicación que puede evolucionar y admitir muchas funciones, pero aún debe ser probada por docentes y de acuerdo a sus reacciones encaminar la misma. Próximos trabajos podrían medir el impacto de la inclusión de estas tecnologías en el aula de clases, y no solo para los estudiantes, sino cómo afecta en la vida profesional al docente.

BoardApp se fundamentó desde las TIC como una herramienta digital que aportaría en el proceso de enseñanza, y desde el reconocimiento de patrones, un área que se expande cada vez más dadas sus numerosas aplicaciones y beneficios. Su implementación requiere también tener en cuenta que existen metodologías de desarrollo de software que ayudan a la consecución de una aplicación como la construida aquí.

Desde la ingeniería este documento sirve como ejemplo de la implementación de árboles de decisión para el reconocimiento de gestos. Se mostraron las etapas y requisitos necesarios para el desarrollo de un proyecto de este estilo, y se deja al lector la inquietud sobre el campo de estudio del *Reconocimiento de Patrones*, sus aplicaciones, alcances y beneficios.

REFERENCIAS

[1] Abdulhakim A., Bt A., Manaf A., Khan S. (2016) A Study of Touching Behavior for Authentication in Touch Screen Smart Devices.

[2] Ahedo J., Danvila del Valle I. (n.d.) Las nuevas tecnologías como herramientas que facilitan la educación formativa en la educación

[3] Baudel T. and Beaudouin-Lafon M. (1993) Charade: Remote control of objects using free-hand gestures. 28, 35.

[4] Becta, I. Advice (2004) (British Educational Communications and Technology Agency), "Getting the most from your Interactive Whiteboard" A guide for secondary schools. 7.

[5] Chacón A., (n.d.) La tecnología educativa en el marco de la didáctica. 23.

[6] Coll, C. (2004). Psicología de la educación y prácticas educativas mediadas por las tecnologías de la información y la comunicación Una mirada constructivista. Revista Electrónica Sinéctica, 25, 1–24.

Recuperado de: <http://www.redalyc.org/src/inicio/ArtPdfRed.jsp?iCve=99815899016>

[7] Dittmar, T., Krull, C., & Horton, G. (2015). A new approach for touch gesture recognition: Conversive Hidden non-Markovian Models. Journal of Computational Science, 10, 66–76. <http://doi.org/10.1016/j.jocs.2015.03.002>

[8] Duda, R. O., Hart, P. E., & Stork, D. G. (1973). Pattern classification (pp. 3-17). Wiley, New York.

[9] Farooq, A., Evreinov, G., Raisamo, R., & Majeed, A. A. (2014). Haptic user interface enhancement system for touchscreen based interaction: A novel system for multimodal interaction with touchscreen interfaces. ICOSST 2014 - 2014 International Conference on Open Source Systems and Technologies, Proceedings, 25–31.

<http://doi.org/10.1109/ICOSST.2014.7029316>

[10] Fernández, A., Musé, P., Ramírez, I., Canela P. & Carvajal, G., "Selección de Características". 2015. Recuperado de: <https://eva.fing.edu.uy/course/view.php?id=514>, Marzo de 2016

[11] Fernández, V. (2014), Evaluación de la salud laboral docente: estudio psicométrico del cuestionario de salud docente, Barcelona.

[12] Fukuda, H., Sakamoto, K., Lin, C. L., Li, C. S., Chang, Y. M., Lin, T. C., ... Kissane, B. (2014). Touch-panel interface system which can recognize who touched the screen and where was pointed. IEEE/OSA Journal of Display Technology, 9(1), 1–4. <http://doi.org/10.1109/INEC.2008.4585597>

- [13] Galvis, A. (1998), Ambientes de Enseñanza – Aprendizaje Enriquecidos con Computador. Boletín de Informática Educativa. UNIANDES-LIDIE Vol. 1(2), pp. 117-145.
- [14] Garcia, A. (2003), Medios Informáticos. Recuperado: Diciembre 27 de 2017, Disponible en: <http://www.web.usal.es/anagy/arti5.htm>
- [15] Guarneri, I., Capra, A., Farinella, G. M., Cristaldi, F., & Battiato, S. (2013). Multi Touch Shape Recognition for Projected Capacitive Touch Screen.
- [16] Harrison, C., & Hudson, S. E. (n.d.). TouchTools : Leveraging Familiarity and Skill with Physical Tools to Augment Touch Interaction.
- [17] Hsu S., Cheng H., Yu C. (2014) Classification of holding hand for dynamic interface adjustment on mobile devices. 112.
- [18] Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence, 22(1), 4-37.
- [19] Karray, F., Alemzadeh, M., Saleh, J. A., & Arab, M. N. (2008). Human-computer interaction: Overview on state of the art.
- [20] Kumar D. (2015) Recognizing Hand Gestures for Human Computer Interaction, 379.
- [21] Larson, R. E. H., Robert, P., Edwards, B. H., & Abellanas Rapún, L. (1999). Cálculo y geometría analítica.
- [22] Ma K. , Li J. , Han J., Hu Y. & Xu W. (n.d.) Enhancing touchscreen input via finger identification.
- [23] Marcelo C. (2013) Las tecnologías para la innovación y la práctica docente. Revista Brasileira de Educação v. 18, 36.
- [24] Marzano, R. J. (2009). Teaching with Interactive Whiteboards. Educational Leadership. <http://doi.org/Article>
- [25] MEN, Ministerio de Educación Nacional (2015) decreto 1655, 20 de Agosto de 2015. Seguridad y Salud en el Trabajo para los educadores afiliados al Fondo Nacional de Prestaciones Sociales del Magisterio. 8-9.
- [26] Myers, Brad. (2001). A Brief History of Human Computer Interaction Technology. Interactions. 46-49.

[27] Organización Mundial de la Salud, OMS. (2004), Prevención de trastornos musculoesqueléticos en el lugar de trabajo, pg. 29

[28] Palma N. (2011) La tecnología de la información y la educación. 48.

[29] Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.

[30] Rautaray S., & Agrawal A. (2012) Vision based hand gesture recognition for human computer interaction: a survey.

[31] Reddy, R. (1996) “To dream the possible dream.” Turing Award Lecture. Communications of the ACM 39, 5. 105–112.

[32] Rodríguez A.V. (2014). Consideraciones y lineamientos para fomentar el uso de los tableros digitales interactivos en los ambientes de aprendizajes educativos en los municipios de Medellín e Itagüí. 13,18.

[33] Sabogal A., Velásquez C. (2014) Implementación de un Tablero Digital de Bajo Costo. 63, 64

[34] Salinas, J. (2004). Cambios metodológicos con las TIC . Estrategias didácticas y entornos virtuales de enseñanza-aprendizaje. Bordón, 58, 3–4. <http://doi.org/469-481>

[35] Sato D., Morimura T., Katsuki T., Toyota Y. , Kato T., Takagi H. (2016), Automated Help System for Novice Older Users from Touchscreen Gestures.

[36] Shahzad M., Liu A., & Samuel A., (2017) Behavior Based Human Authentication on Touch Screen Devices Using Gestures and Signatures. 26.

[37] Shin H., Lim J., Oh C., Kim M., Jeong H. & Son J. (2015) Performance Comparison of Tap Gestures on Small-Screen Touch Devices. 120-121




[38] UNESCO (2004) Las tecnologías de la información y la comunicación en la formación docente. 42.





[39] UNESCO, (2005), Estándares UNESCO de competencia en TIC para docentes. Paris.




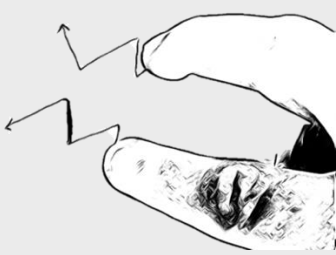
[40] Xuesong, G., Xin, W., & Qin, S. A. (2015). Teaching Application Research of Electronic White Board Used in Design Disciplines. 2015 Sixth International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), 794–797. <http://doi.org/10.1109/ISDEA.2015.200>

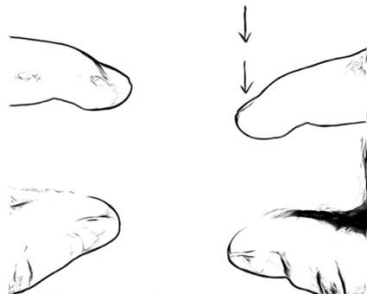

ANEXOS

Anexo 1: Tabla que contiene los gestos a considerar

	Gesto	Descripción
1		Resaltador: Dibujar en línea recta con dos dedos.
2		Escribir con dos dedos: estilo bolígrafo.
3		Arco de circunferencia: Girar con dos dedos abiertos.
4		Regla: Pulsar con tres dedos sin soltar, movimiento.

		
5		Borrador: Con cuatro dedos y mover.
6		Seleccionar objetos: Encerrar objetos con dos dedos.
7		Clic con tres dedos: Un solo clic con tres dedos para desplegar opciones de un(os) objeto(s) seleccionado(s).
8		Circunferencia: tocar con 5 dedos separados y mover.

		
9		<p>Zoom In: Abrir con dos dedos.</p>
10		<p>Zoom Out: Cerrar (pellizcar). Con dos dedos.</p>
11		<p>Tijeras: Cortar con dos dedos, movimiento en zig-zag tipo tijeras.</p>

12		<p>Captura de pantalla: Con 4 dedos, se desliza hacia abajo el índice derecho (Obturador cámara) para tomar pantallazo.</p>
13		<p>Cambio de color de escritura: Doble clic con los dedos en posición de escritura.</p>

Anexo 2: Código del árbol de decisión implementado en BoardApp

```
if Cantidaddedos<2.5
  if VelocidadPromedioXdedo2<-1.04695
    outputArg1 = 16
  else
    if Cantidaddedos<1.5
      outputArg1 = 7
    else
      if VelocidadPromedioXdedo1<0.538683
        if VelocidadDedo2<0.000481334
          if CambiosDirXdedo1<1.5
            outputArg1 = 17
          else
            outputArg1 = 18
          end
        else
          if VelocidadPromedioYdedo1<0.835651
            if VelocidadPromedioYdedo2<-1.19554
              if AceleracinPromedioYdedo1<-6.55254
                if VelocidadPromedioYdedo1<-1.18885
                  outputArg1 = 13
                else
                  outputArg1 = 9
                end
              else
                outputArg1 = 9
              end
            else
              if VelocidadPromedioXdedo1<-0.0807819
                if VelocidadPromedioXdedo2<-0.295168
                  if AceleracinPromedioYdedo2<1.14224
                    if CambiosDirYdedo2<3.5
                      outputArg1 = 6
                    else
                      outputArg1 = 11
                    end
                  else
                    outputArg1 = 6
                  end
                else
                  if CambiosDirXdedo2<13.5
```



```

if VelocidadPromedioYdedo1<-0.301874
    outputArg1 = 10
else
    if AceleracinPromedioXdedo1<11.6062
        if VelocidadPromedioYdedo2<-0.26612
            outputArg1 = 11
        else
            outputArg1 = 6
        end
    else
        outputArg1 = 18
    end
end
else
    outputArg1 = 11
end
end
else
    if VelocidadPromedioXdedo2<0.362668
        if VelocidadDedo2<0.249085
            if CambiosDirXdedo1<3
                outputArg1 = 18
            else
                if VelocidadPromedioYdedo2<-0.0432051
                    outputArg1 = 6
                else
                    if VelocidadDedo2<0.125628
                        outputArg1 = 18
                    else
                        outputArg1 = 3
                    end
                end
            end
        end
    end
end
else
    if VelocidadPromedioXdedo1<0.102254
        if VelocidadDedo2 <1.02317
            if VelocidadPromedioYdedo1<0.536721
                if CurvaturaDedo2<2.26011
                    outputArg1 = 10
                else
                    if VelocidadPromedioXdedo2<-0.3221
                        outputArg1 = 3
                    else
                        if AceleracinPromedioXdedo1<3.91882
                            if AceleracinPromedioYdedo2<-27.035
                                outputArg1 = 18
                            else
                                else

```

```

        outputArg1 = 9
    end
    else
        outputArg1 = 6
    end
end
end
else
    outputArg1 = 10
end
else
    if VelocidadPromedioYdedo1<-1.7103
        outputArg1 = 9
    else
        outputArg1 = 10
    end
end
else
    if AcelerainPromedioXdedo2<4.38384
        if VelocidadDedo1<0.440528
            outputArg1 = 3
        else
            outputArg1 = 9
        end
    else
        outputArg1 = 18
    end
end
end
else
    if VelocidadPromedioYdedo2<-0.660901
        outputArg1 = 9
    else
        outputArg1 = 3
    end
end
end
else
    if VelocidadPromedioYdedo2<1.06615
        if VelocidadPromedioYdedo2<0.377797
            outputArg1 = 10
        else
            outputArg1 = 9
        end
    else
        if VelocidadPromedioXdedo2<32767.8

```

```

        outputArg1 = 14
    else
        outputArg1 = 10
    end
end
end
end
else
if VelocidadPromedioXdedo2<1.08058
if CambiosDirXdedo1<2.5
if AceleracinPromedioXdedo2<4.98705
outputArg1 = 3
else
if CurvaturaDedo1<3.64171
outputArg1 = 2
else
outputArg1 = 1
end
end
else
if VelocidadPromedioYdedo1<1.74309
if AceleracinDedo2<1.66032
outputArg1 = 15
else
outputArg1 = 3
end
else
outputArg1 = 10
end
end
else
if VelocidadPromedioXdedo1<1.59708
if AceleracinPromedioYdedo1<-23.7019
outputArg1 = 2
else
if CambiosDirYdedo2<2.5
outputArg1 = 1
else
if CambiosDirXdedo1<4.5
outputArg1 = 15
else
outputArg1 = 2
end
end
end
else
if VelocidadPromedioXdedo1<32769.3

```

```

if VelocidadPromedioXdedo1<2.66201
  if CambiosDirYdedo1<2.5
    outputArg1 = 15
  else
    if CambiosDirYdedo2<8.5
      if CurvaturaDedo2<0.0252587
        outputArg1 = 15
      else
        if AceleracinDedo2<0.701065
          outputArg1 = 2
        else
          if VelocidadPromedioXdedo1<1.70906
            outputArg1 = 15
          else
            if VelocidadPromedioYdedo1<-0.121906
              if VelocidadPromedioYdedo1<-0.938644
                outputArg1 = 15
              else
                if CambiosDirYdedo2<1.5
                  outputArg1 = 15
                else
                  outputArg1 = 2
                end
              end
            end
          else
            if VelocidadPromedioYdedo2<0.201714
              if VelocidadPromedioYdedo2<-0.0840729
                outputArg1 = 2
              else
                if VelocidadPromedioXdedo1<1.78965
                  outputArg1 = 2
                else
                  outputArg1 = 15
                end
              end
            end
          else
            outputArg1 = 2
          end
        end
      end
    end
  end
end
end
end
end
else
  outputArg1 = 2
end
end
else

```

```

        outputArg1 = 2
    end
else
    outputArg1 = 17
end
end
end
end
end
end
else
    if Cantidaddedos<4.5
        if Cantidaddedos<3.5
            if CurvaturaDedo3<9.21352
                outputArg1 = 4
            else
                outputArg1 = 5
            end
        else
            if VelocidadPromedioXdedo4<0.0533998
                if VelocidadPromedioXdedo1<0.162686
                    if AceleracinPromedioYdedo3<89.179
                        if AcelerainPromedioXdedo4<-12.9043
                            outputArg1 = 5
                        else
                            outputArg1 = 12
                        end
                    else
                        outputArg1 = 5
                    end
                else
                    if VelocidadPromedioXdedo1<1.46481
                        outputArg1 = 5
                    else
                        outputArg1 = 12
                    end
                end
            else
                outputArg1 = 5
            end
        end
    else
        outputArg1 = 8
    end
end
end

```