



ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA
GRADO EN INGENIERÍA INFORMÁTICA

Curso académico 2015/2016

Trabajo Fin de Grado

**Adaptación de actividades educativas sobre mesas
multicontacto a pizarras digitales**

Autor: Alejandro Rubio Barrena
Tutores: Estefanía Martín Barroso, David Roldán Álvarez

RESUMEN

Las tecnologías multicontacto son superficies donde varios usuarios pueden estar interactuando a la vez de una forma natural a través de gestos realizados con las manos y/o con los dedos. Las posibilidades educativas de estas superficies son muy extensas. Estos dispositivos proveen un espacio físico — una mesa — y se manipulan — manos y dedos— de igual forma que los instrumentos tradicionales del aula. Ahora bien, con las ventajas que aportan los medios informáticos que potencian la creatividad, curiosidad e innovación del alumno, y flexibilizan el proceso educativo.

DEDOS es un software gratuito desarrollado por la Universidad Autónoma de Madrid, la Universidad Rey Juan Carlos y la Fundación Síndrome de Down de Madrid. Dicha aplicación permite crear distintos tipos de actividades educativas tanto individuales como colaborativas que pueden ser utilizadas sobre distintos dispositivos entre los cuáles están las mesas multicontacto. Sin embargo, esta aplicación no contaba con una versión especialmente adaptada a dispositivos muy frecuentes en las aulas actuales: las pizarras digitales interactivas. Parece que su disponibilidad en las aulas de clase, conjuntamente con unas sencillas orientaciones iniciales al profesorado, da lugar a una progresiva renovación de los procesos de enseñanza y aprendizaje que se va extendiendo a todo el profesorado, induciendo una notable renovación de las metodologías docentes y de los procesos de enseñanza y aprendizaje, incrementando la motivación de los estudiantes, revitalizando la autoestima profesional de los profesores y facilitando el logro de aprendizajes más significativos y acordes con la sociedad actual.

El objetivo de este Trabajo Fin de Grado es adaptar el software DEDOS-Player usado para la realización de actividades educativas en las aulas a una nueva versión, que sea capaz de ejecutarse en pizarras digitales, y que ofrezca las mismas posibilidades educativas en este nuevo dispositivo. El software está desarrollado en el entorno Adobe Flash, con el lenguaje de programación ActionScript, el lenguaje que utiliza dicho entorno. Ha de modificarse el código ya existente del software para implementar las nuevas funcionalidades para pizarras digitales.

A lo largo de esta memoria se hará un pequeño repaso a la tecnología de las mesas multicontacto, ya que es el punto inicial desde el que parte este proyecto. Asimismo, se analizará la pizarra digital interactiva como tecnología, así como los beneficios que tienen las TIC en las aulas de educación, y más específicamente las PDI en dicho contexto.

También se explicará con detalle el diseño que se ha adoptado para la nueva actualización de la herramienta DEDOS-Player, partiendo del prototipo inicial. Se detallará cómo se han implementado estos cambios y la posterior evaluación de las nuevas características implementadas enfocadas a las pizarras digitales interactivas.

Quisiera dar las gracias a mi tutora Estefanía Martín Barroso por su ayuda en la realización de este proyecto. Gracias también a David Roldán Álvarez por su ayuda en la parte técnica de la aplicación. Agradecimientos también a la Fundación Síndrome de Down de Madrid.

ÍNDICE DE CONTENIDOS

1.	MOTIVACIÓN.....	1
1.1.	MESAS MULTICONTACTO.....	1
1.2.	PIZARRAS DIGITALES INTERACTIVAS	4
1.3.	LAS PDIs COMO RECURSO EN EL AULA	7
1.4.	PROBLEMAS ACTUALES.....	11
1.5.	ESTRUCTURA DE LA MEMORIA	12
2.	OBJETIVO Y DISEÑO	13
2.1	MENÚ DE INICIO	14
2.2	REPRODUCTOR	16
3.	IMPLEMENTACIÓN.....	19
3.1	LENGUAJE UTILIZADO EN LA IMPLEMENTACIÓN.....	19
3.2	HERRAMIENTAS UTILIZADAS	20
3.3	ORGANIZACIÓN DE LA APLICACIÓN	21
3.4	ADAPTACIONES REALIZADAS.....	25
4.	EVALUACIÓN.....	37
4.1	EVALUACIÓN DE LA HERRAMIENTA EN LA FSDM.....	37
4.2	EVALUACIÓN DE LA HERRAMIENTA EN LA URJC	43
5.	CONCLUSIONES	51
5.1	LOGROS ALCANZADOS	51
5.2	TRABAJOS FUTUROS	53
	BIBLIOGRAFÍA	55
	ANEXO I: INSTALACIÓN DE ADOBE FLASH CS5.5	57
	ANEXO II: CREACIÓN E INSTALACIÓN DEL FICHERO .AIR.....	59

ÍNDICE DE TABLAS Y FIGURAS

Tabla 1 - Diferencias entre el aprendizaje tradicional y el aprendizaje que involucra las TIC.....	7
Tabla 2 - Tabla resumen de rotaciones para cada jugador según el número de jugadores y dispositivo.....	36
Tabla 3 - Estadísticas de W3Schools.com de las resoluciones que utilizan los usuarios de navegadores web.....	39
Ilustración 1- Vista horizontal y planta de la SUR40 o PixelSense de Microsoft-Samsung	2
Ilustración 2 – MultiTaction Cell MT550W7	3
Ilustración 3 - Atouch Interactive Table.....	3
Ilustración 4 - Ejemplos de una PDI donde los alumnos manipulan los objetos a través de un dispositivo externo como es el bolígrafo o con sus propias manos en el caso que sea una PDI táctil.....	4
Ilustración 5 - Esquema explicativo del accesorio eBeam para una PDIP	6
Ilustración 6 - Estado actual del menú de inicio de DEDOS-Player	14
Ilustración 7 - Prototipo inicial del menú de DEDOS para PDIs	16
Ilustración 8 - Ejemplo de actividad en DEDOS para mesas multicontacto	17
Ilustración 9 - Ejemplo de uso de DEDOS en una mesa multicontacto	18
Ilustración 10 - Aspecto de una actividad en DEDOS para dos jugadores en mesa multicontacto	18
Ilustración 11 - Prototipo inicial del reproductor DEDOS para PDIs con un ejemplo de actividad para dos jugadores	18
Ilustración 12 - Diagrama UML de clases de DEDOS-Player	22
Ilustración 13 – Ejemplo de actividad creada con el editor de DEDOS	23
Ilustración 14 - Diagrama UML del paquete OptionsMenu.....	24
Ilustración 15 - Tercer separador corta una de las opciones de la última columna.....	26
Ilustración 16 - Aspecto final del menú inicial.....	27
Ilustración 17 - Rotación de 90° a la zona común de jugadores para el caso de dos jugadores.....	28
Ilustración 18 - Sin rotación en la zona común de jugadores para el caso de tres jugadores.....	29
Ilustración 19 - Rotación del jugador 3 en el caso de cuatro jugadores jugando simultáneamente	30
Ilustración 20 - Esquema de la anchura y altura de los contenedores de juego y jugador en mesas multicontacto.....	31
Ilustración 21 - Esquema de la anchura y altura de los contenedores de juego y jugador en pizarras digitales	32
Ilustración 22 - Aspecto después de la implementación de la versión para PDIs para dos jugadores.....	33

Ilustración 23 - Aspecto después de la implementación de la versión para PDIs para tres jugadores.....	34
Ilustración 24 - Aspecto después de la implementación de la versión para PDIs para cuatro jugadores.....	35
Ilustración 25 - Relación de aspecto 4:3.....	38
Ilustración 26 - Relación de aspecto 16:9.....	38
Ilustración 27 - Estadísticas de uso de resoluciones de pantalla desde noviembre de 2013 a septiembre de 2014, según StatCounter.....	40
Ilustración 28 - Menú inicial de la versión 0.1 de DEDOS-Player para PDIs	41
Ilustración 29 - Actividad para dos jugadores en la versión 0.1 de DEDOS-Player para PDIs	42
Ilustración 30 - Actividad para tres jugadores en la versión 0.1 de DEDOS-Player para PDIs	42
Ilustración 31 - Actividad para dos jugadores en la versión 0.2 de DEDOS-Player para PDIs	44
Ilustración 32 - Actividad para tres jugadores en la versión 0.2 de DEDOS-Player para PDIs	45
Ilustración 33 - Actividad para cuatro jugadores en la versión 0.2 de DEDOS-Player para PDIs	45
Ilustración 34 - Actividad para dos jugadores en la versión 0.27 de DEDOS-Player para PDIs	47
Ilustración 35 - Actividad para tres jugadores en la versión 0.27 de DEDOS-Player para PDIs	47
Ilustración 36 - Actividad para cuatro jugadores en la versión 0.27 de DEDOS-Player para PDIs	48
Ilustración 37 - Prototipo alternativo para tres jugadores.....	48
Ilustración 38 - Prototipo alternativo para cuatro jugadores	49
Ilustración 39 - Actividad para cuatro jugadores aplicando el prototipo alternativo	49
Ilustración 40 - Pantalla inicial del instalador (Adobe Flash CS5.5)	57
Ilustración 41 - Opciones de instalación (Adobe Flash CS5.5).....	57
Ilustración 42 - Proceso de instalación (Adobe Flash CS5.5).....	58
Ilustración 43 - Pantalla inicial de Adobe Flash CS5.5.....	59
Ilustración 44 - Ventana de "Publish Settings" (Adobe Flash CS5.5).....	60
Ilustración 45 - Ventana de "AIR 2.6 Settings" (Adobe Flash CS5.5).....	60
Ilustración 46 - Elección de certificado (Adobe Flash CS5.5).....	61
Ilustración 47 - Instalación del fichero .air.....	62

1. MOTIVACIÓN

Este trabajo fin de grado se centra en la adaptación de un software educativo existente diseñado para mesas multicontacto hacia pizarras digitales interactivas, un dispositivo que es uno de los más utilizados en las aulas, según el último informe de la Unión Europea (European Union, 2013). Las mesas multicontacto están empezando a utilizarse en diferentes sectores, aunque siempre desde un marco de trabajo cooperativo donde varios usuarios podrán interactuar a la vez con los elementos mostrados en pantalla. Podemos encontrar estos dispositivos en aeropuertos, centros comerciales y grandes eventos, entre otros. En lo que respecta al entorno educativo, el informe Horizon 2014 (Horizon, 2014) destaca las ventajas de utilizar superficies táctiles dentro de las aulas, así como la creciente popularidad de los juegos y la gamificación de los contenidos, que fomentan el pensamiento crítico, la resolución de problemas de forma creativa y la colaboración entre los individuos. Por lo tanto, la unión de dispositivos táctiles con un software capaz de crear actividades educativas, formarían la unión perfecta para explotar las virtudes de ambos mundos, mejorando la experiencia educativa del alumnado.

En este primer capítulo se expondrán las características generales de las tecnologías de mesas multicontacto y de pizarras digitales interactivas, así como las posibilidades de uso que tienen estas últimas en el aula, ya que es el dispositivo sobre el cual se centra este trabajo fin de grado. Después se describirán aspectos a tener en cuenta cuando se usan pizarras digitales interactivas en las aulas y problemas relacionados. Para terminar, se mostrará la estructura de esta memoria.

1.1. MESAS MULTICONTACTO

La tecnología multitáctil concierne a una técnica de interacción persona-ordenador, así como al hardware que la emplea. Consiste en una pantalla táctil que reconoce múltiples puntos de contacto de manera simultánea, y el software que controla la pantalla, el cual interpreta las interacciones. El desarrollo de esta técnica tuvo sus inicios en la década de 1980, cuando la Universidad de Toronto presentó un prototipo¹

¹ SK. Lee, W. Buxton, K.C. Smith, University of Toronto – “A multi-touch three dimensional touch-sensitive tablet”, 1985: <http://www.billbuxton.com/leebuxtonsmith.pdf>

de tableta multitáctil. El teléfono iPhone fue uno de los primeros dispositivos en adoptar esta tecnología multitáctil².

Las mesas multitáctiles pueden utilizarse en diversos sectores, aunque siempre se usan desde un marco de trabajo cooperativo, en el que varios usuarios a la vez pueden interactuar con los elementos que se muestran en la pantalla. Algunos de los sectores en los cuales se utiliza este tipo de tecnología son: información aeroportuaria, información de grandes eventos (como congresos), aplicaciones y juegos médicos, museos, oficinas de turismo y ámbito de la educación, entre otros.

A continuación se detallan algunos ejemplos de mesas multitáctiles así como sus características técnicas, desarrolladas y comercializadas por distintas empresas:

- **Microsoft PixelSense³:** También conocida como SUR40, se trata de una mesa multitáctil realizada conjuntamente por Microsoft y Samsung. Reconoce formas y objetos, y detecta hasta 52 puntos de contacto simultáneos, haciendo posible que hasta cuatro personas la utilicen a la vez. Cuenta con una pantalla de 40 pulgadas con resolución 1929x1080 píxeles. Destaca su escaso grosor, de únicamente 4 centímetros. La Ilustración 1 presenta dos fotografías de esta mesa tanto en vista horizontal como desde arriba. Se enfoca a su uso en la empresa, facilitando tareas que se lleven a cabo en grupo. Su precio ronda los 7.900 €.



Ilustración 1- Vista horizontal y planta de la SUR40 o PixelSense de Microsoft-Samsung⁴

² Tecnología FingerWorks para iPhone: <https://en.wikipedia.org/wiki/FingerWorks>

³ Microsoft Pixelsense: <http://www.microsoft.com/en-us/pixelsense/default.aspx>

⁴ Fuente de la ilustración 1: <http://goo.gl/FZq0bS>

- **MultiTaction Cell MT550W7⁵** (véase la Ilustración 2): Desarrollada por la empresa *MultiTouch*, su característica más destacable es su tecnología especial que permite reconocer un número ilimitado de puntos de contacto. Cuenta con una pantalla de 55 pulgadas. Su futuro parece estar en museos o comercios con gran afluencia de público, para que el mayor número de personas pueda interactuar con ella al mismo tiempo.



Ilustración 2 – MultiTaction Cell MT550W7⁶

- **Atouch Interactive Table⁷** (véase la Ilustración 3 - Atouch Interactive Table): Se trata de una mesa interactiva desarrollada por la empresa española *VirtualWare Group*. Puede integrarse en múltiples ambientes, como educativo, corporativo, comercio, museos, etc. Dispone de una pantalla de 55” LED que, junto con la tecnología multitáctil de infrarrojos admite 32 toques simultáneos.



Ilustración 3 - Atouch Interactive Table⁸

⁵ *MultiTaction Cell MT550W7*: <http://goo.gl/lwsrir>

⁶ Fuente de la ilustración 2: <http://goo.gl/hZd1m9>

⁷ *Atouch Interactive Table*: <http://goo.gl/8lprnq>

⁸ Fuente de la ilustración 3: <http://goo.gl/uf9XY4>

1.2. PIZARRAS DIGITALES INTERACTIVAS

Una pizarra digital interactiva o PDI es simplemente un ordenador conectado a un videoprojector, el cual muestra la señal de dicho ordenador sobre una superficie desde la que se puede controlar el ordenador (véase la Ilustración 4). La principal función de la pizarra es interactuar con el ordenador a través de esta superficie. Esta interacción se puede realizar bien con un dispositivo externo como puede ser un bolígrafo (véase la parte izquierda de la Ilustración 4), las más nuevas incluso permiten que los alumnos interactúen directamente con sus propias manos sobre los objetos que se están proyectando (véase la parte derecha de la ilustración). Esto ofrece la posibilidad de interactuar directamente con los objetos multimedia que se están proyectando, y es la principal diferencia respecto a una pantalla digital, refiriéndonos a pantalla digital como un ordenador y un proyector trabajando conjuntamente. Por este motivo se dicen que son interactivas, debido a que los usuarios manipulan e interactúan directamente con los objetos mostrados en la PDI.



Ilustración 4 - Ejemplos de una PDI donde los alumnos manipulan los objetos a través de un dispositivo externo como es el bolígrafo o con sus propias manos en el caso que sea una PDI táctil⁹

Una PDI cuenta con los siguientes elementos:

- **Ordenador** (portátil o sobremesa): El ordenador es quien va a reproducir la información multimedia que tiene almacenada en el disco duro. El sistema operativo del ordenador tiene que ser compatible con el software del que disponga la pizarra.
- **Proyector**: El proyector habilita la visión de la imagen que se ve en el ordenador sobre la pizarra. Es conveniente colocar el proyector en el techo y a una cierta

⁹ Fuente de la ilustración 4: <http://goo.gl/JzSJI0>

distancia de la pizarra, de manera que se obtenga una mejor imagen y a un buen tamaño que se ajuste por completo a las dimensiones de la PDI.

- **Medio de conexión:** Comunica el ordenador y la pizarra. Existen conexiones a través de cable (USB, por ejemplo), *bluetooth*, o conexiones basadas en tecnologías de identificación por radiofrecuencia.
- **Pantalla interactiva:** Pantalla sobre la que es proyectada la imagen del ordenador, y que se controla mediante un puntero (p.e. un bolígrafo) o las propias manos del usuario dependiendo de si es táctil o no.
- **Software de la PDI:** Lo proporciona el fabricante o distribuidor. Otorga la funcionalidad destinada a gestionar las interacciones que los usuarios realizan sobre la pizarra digital interactiva.

Habitualmente, al adquirir una PDI solo se incluyen los siguientes: la pantalla, los elementos para interactuar con ella (si así lo requiere la pizarra, por ejemplo, un bolígrafo), el software correspondiente y el cableado necesario. A esto hay que añadir el proyector y el ordenador, así como los periféricos y accesorios que se consideren necesarios para su uso como un ratón o teclado.

También existen las PDIP (*Pizarra Digital Interactiva Portátil*), que son PDIs que pueden trasladarse fácilmente de un lugar a otro. Además, se puede utilizar cualquier superficie de proyección, por ejemplo una pantalla plegable o la pantalla gigante de un auditorio, por lo que no incluyen una pantalla. Un ejemplo de PDIP es eBeam Engage¹⁰. Se trata de una barra que se adhiere al lateral de la superficie que actúa como pizarra, y que, gracias al software de captura de movimientos y al lápiz que incluye, permite la manipulación de la imagen que se proyecta en la pantalla improvisada (véase la Ilustración 5).

Para resolver el problema de la interacción del usuario con los objetos que se muestran en las pizarras digitales interactivas, éstas pueden utilizar una de las siguientes tecnologías:

- **Infrarroja:** El movimiento del dedo o puntero es capturado en la pizarra por un receptor que recibe su interferencia con la luz infrarroja en la superficie de la pizarra. Cuando se presiona sobre un punto de la pizarra, un software triangula la

¹⁰ eBeam Engage: <http://goo.gl/s6eB30>

ubicación del dedo o lápiz, traduciendo la ubicación del punto (o los puntos) infrarrojos a coordenadas cartesianas, las cuales son usadas para ubicar el ratón. Esta tecnología no limita el área de proyección, pudiendo ser de varios metros cuadrados.

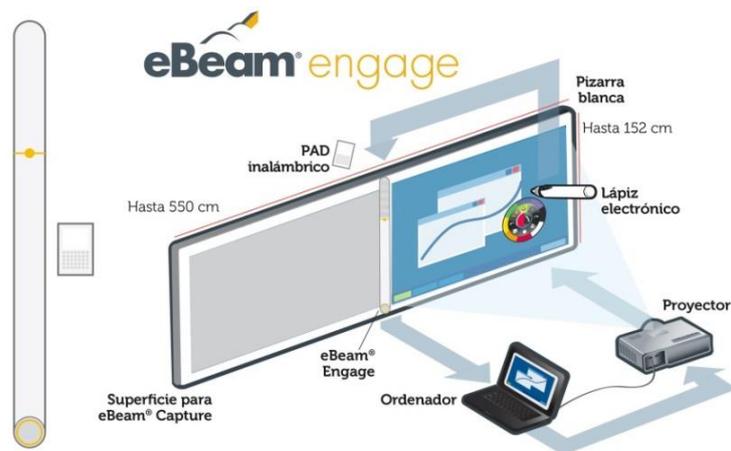


Ilustración 5 - Esquema explicativo del accesorio eBeam para una PDI¹¹

- **Electromagnética:** En una PDI electromagnética se utiliza un lápiz especial que actúa como puntero, que se combina con una malla ubicada en toda la superficie de proyección de la propia pizarra. Al detectar una pulsación en la pantalla con el puntero, esta malla detecta con precisión la ubicación de la señal que genera el lápiz en la pantalla, enviando un mensaje al ordenador cuando se detecta una pulsación. Por lo general, estos punteros no necesitan pilas u otra fuente de energía, ya que únicamente alteran las señales eléctricas producidas por la pizarra.
- **Ultrasonidos – Infrarroja:** Esta tecnología utiliza la luz infrarroja y un sistema de posicionamiento por ultrasonidos. Es la tecnología que utiliza el ejemplo visto anteriormente, eBeam Engage¹². El dispositivo escanea un área, recalibrando la imagen proyectada para ajustarse a las dimensiones de la pantalla.
- **Resistiva:** El panel de la pizarra está formado por dos capas separadas, deformándose la capa exterior bajo presión para hacer contacto con la segunda capa. Al realizarse el contacto entre las láminas exteriores e interiores, se provoca una variación de la resistencia eléctrica, la cual permite localizar el punto señalado.
- **Proyector interactivo:** Esta tecnología incluye una cámara CMOS (*Semiconductor Complementario de Óxido Metálico*) incorporada en el proyector, por lo que,

¹¹ Fuente de la ilustración 5: <http://goo.gl/xOkV2y>

¹² eBeam Engage: <http://goo.gl/kvUqHc>

además de producir la imagen, detecta la posición de un puntero al entrar en contacto con la superficie donde la imagen está proyectada. Se trata de una solución patentada en 2010 por la empresa americana Boxlight¹³.

- **Basada en Wiimote – infrarrojos:** Se trata de una tecnología que utiliza el Wiimote, el mando de la consola Wii. Esta tecnología la desarrolló Johnny Chung Lee en 2007¹⁴, aprovechando que el Wiimote puede rastrear un puntero que tenga una luz infrarroja. Es un enfoque interesante, debido a su precio económico, a las opciones de código abierto y ofertas comerciales de futuro¹⁵, y a que tiene una curva de aprendizaje suave, debido a que el sistema de juego es ya familiar para muchos.

1.3. LAS PDIs COMO RECURSO EN EL AULA

En los últimos años ha habido un alto interés en las Tecnologías de la Información y de la Comunicación (TIC) y sus usos educativos, como muestra tenemos la multitud de investigaciones que tienen por objeto de estudio las TIC en la educación y su uso educativo (Roschelle y otros, 2000; Passey, 2006; o Tondeur, Braak y Valcke, 2007).

Las nuevas tecnologías aplicadas al aula llevan a nuevas nociones del proceso de enseñanza-aprendizaje, que producen una mayor implicación del alumno en su propio proceso de aprendizaje, como vemos en la siguiente comparación¹⁶:

Tabla 1 - Diferencias entre el aprendizaje tradicional y el aprendizaje que involucra las TIC

AMBIENTE DE APRENDIZAJE TRADICIONAL	AMBIENTES DE APRENDIZAJE CON TIC
Instrucción dada por el docente	Aprendizaje enfocado en el estudiante
Trabajo individual	Trabajo colaborativo
Transmisión de información lineal	Hay intercambio de información
Aprendizaje pasivo	Aprendizaje activo, exploratorio, se basa en la indagación
Aprendizaje fáctico, se basa en la experiencia	Pensamiento crítico, toma de decisiones informadas

¹³ Patente proyector interactivo (Boxlight): US PATENT No: US 7,703,926 B2. April 27, 2010, CMOS cameras embedded inside of a projector: <http://patentimages.storage.googleapis.com/pdfs/US7703926.pdf>

¹⁴ Johnny Chung Lee, “Low-Cost Multi-point Interactive Whiteboards Using the Wiimote”: <http://johnnylee.net/projects/wii/>

¹⁵ Wiimote Whiteboard Concept: <https://goo.gl/axuvMv>

¹⁶ Comparación extraída de: Riascos-Eraza, Quintero-Calvache, Ávila-Fajardo. ‘Information Technology in the Classroom: The Views of University Professors’, 2009: <http://goo.gl/NpizHf>

Adoptar una nueva herramienta de trabajo en el aula lleva asociado, de manera inevitable, un aprendizaje de nuevas técnicas de trabajo que se adapten al nuevo contexto educativo. Podemos usar de ejemplo la inserción que tuvo en su día en el aula el vídeo o el proyector de diapositivas. En ambos casos era necesario el aprendizaje de una serie de procedimientos de uso. Según avanzamos en el tiempo, los dispositivos se han hecho más complejos, hasta llegar al ordenador. El nivel de complejidad que implica el uso de un ordenador ha superado claramente el de cualquier otra herramienta que se haya utilizado hasta ahora en educación. Sin embargo, el enorme potencial que llevan asociados los infinitos recursos que podemos encontrar en Internet ha empujado a los ordenadores a entrar en las aulas, y a utilizarlos como una herramienta más.

Podríamos decir que, con la PDI en el aula, se crea un salto tecnológico muy importante, similar al cambio que produjo el ordenador. Las PDIs fomentan la participación del alumnado, pues proporcionan un entorno en el que es fácil compartir contenidos e interactuar con ellos. Con las PDI suele ser más sencillo captar la atención de los alumnos, ya que hay disponible todo un abanico de recursos que permiten la adaptación a diferentes estilos de aprendizaje. Las herramientas asociadas a estos dispositivos hacen posible atender a la diversidad del alumnado, por ejemplo adaptando el tamaño de las fuentes utilizadas, o bien ampliando el tipo de actividades que se llevan a cabo en el aula. Además, la multitud de recursos creados ya en la red así como las herramientas de autor que permiten a los profesores crear sus propios contenidos, ofrecen una adaptación completa de los materiales educativos con los que se trabaja en el aula a las necesidades educativas de los alumnos, haciéndose especialmente importante en el ámbito de la Educación Especial.

Estas pizarras sirven especialmente en los primeros ciclos de Educación (Infantil y Primaria), y para niños con necesidades educativas especiales. Un alumno con necesidades educativas especiales presenta mayores dificultades que el resto de sus compañeros en el aprendizaje, y para compensarlas necesita ciertas adaptaciones de acceso que resultan significativas. Las necesidades educativas especiales son temporales o permanentes y a su vez, pueden ser debidas a causas físicas, psíquicas, sensoriales, afectivo – emocionales o inadaptación (cultural, lingüística, etc.). Desde hace unos años, ha podido comprobarse que para los niños con necesidades educativas especiales la PDI es un recurso motivador, y no discriminador, capaz de adaptarse a las

necesidades del alumnado¹⁷. Las TIC se adentran en la educación de manera exitosa debido a que cada alumno tiene necesidades diferentes, y con herramientas como la PDI pueden cubrirse más ampliamente estas necesidades. Además, las TIC son una realidad en las aulas docentes tal y como refleja el último informe de la Unión Europea donde se refleja que el equipamiento tecnológico de las aulas españolas frente a otros países europeos es elevado ya que sólo nos superan los países nórdicos y Dinamarca (European Union, 2013).

Las PDIs permiten realizar cualquier tipo de actividades educativas que el docente considere oportuno gracias al uso de recursos educativos cerrados creados por editoriales, por profesionales del ámbito materiales multimedia educativos, o bien actividades docentes preparadas por los propios profesores con alguna herramienta de autoría de contenidos, como JClíc¹⁸, Cuadernia¹⁹ o DEDOS²⁰, entre otros. Los siguientes son solo algunos ejemplos de actividades que se pueden realizar en estos dispositivos:

- Escribir palabras o frases.
- Hacer gráficos, esquemas o dibujos.
- Juegos de enlazar elementos de dos conjuntos.
- Resolver puzles con distintas imágenes.
- Realizar actividades de lectura y escritura.
- Juegos de buscar diferencias entre imágenes.
- Juegos de resolver laberintos.
- Ordenar o clasificar imágenes o palabras por distintos criterios como tamaño, o longitud.
- Resolver problemas de matemáticas u operaciones de manera visual.
- Exposición de trabajos individuales o colectivos con imágenes, sonidos, etc.

Éstas serían, por tanto, las ventajas más destacables que ofrece el uso de una PDI en el aula:

¹⁷ Cita de Lara Romero – Pedagoga del departamento de formación de StudyPLAN (Tecnología y Educación) <http://goo.gl/9fSZat>

¹⁸ JClíc: <http://goo.gl/u05SjC>

¹⁹ Cuadernia: <http://cuadernia.educa.jccm.es/>

²⁰ Proyecto DEDOS: <http://aprendecondedos.es/>

- **Captar y focalizar la atención del alumno:** Trabajar con herramientas específicas está dando muy buenos resultados en niños con trastornos, necesidades o déficit, como por ejemplo niños con trastorno de déficit de atención (TDAH).
- **Trabajar la independencia y el auto-concepto:** El niño puede manipular él solo la pizarra digital desde edades muy tempranas, algo que favorece y estimula su independencia y autoestima, a la vez que mejora su interacción con el entorno.
- **Motivación:** Al ser un elemento muy visual, proporciona una gran variedad de colores y formas que pueden ser manipulados. Esta manipulación afianza la confianza en el aprendizaje del niño, ayudándose de características como la paciencia (al realizar prueba y error), el refuerzo positivo (en los distintos pasos que el niño hace en la PDI) o la interactividad.
- **Adaptabilidad:** La PDI permite que los contenidos sean adaptados a las necesidades que el profesorado se va encontrando en el aula (desde zoom de objetos hasta sonidos o lenguajes icónicos que apoyan las explicaciones del docente). De esta manera, las clases son más abiertas al adaptar los contenidos y ser modificados durante su proceso.
- **Visibilidad de las acciones a toda la clase:** El uso de la PDI en el aula permite que todos los alumnos vean los objetos que se están proyectando y, además, tener diferentes dinámicas en el aula, donde el profesor puede estar interactuando con la PDI guiado por sus alumnos, un alumno puede estar resolviendo la actividad mientras otros ven cómo la realizan o bien un alumno gracias a la colaboración de sus compañeros va completando la tarea educativa en este dispositivo.

Como conclusión, podemos decir que la PDI se utiliza como instrumento para adquirir y afianzar los prerrequisitos del aprendizaje, tales como atención, percepción, o memoria. Se usa también como metodología en el desarrollo de los conocimientos en las áreas curriculares y en el desarrollo de las habilidades y conocimientos respecto al uso de la TIC y su habilidad en la vida diaria, y como recurso motivador y no discriminador para los alumnos con necesidades educativas especiales, porque se adapta a las necesidades específicas de cada alumno. La pizarra se usa como apoyo de las explicaciones del profesorado y facilita al profesor la posibilidad de presentar sus actividades y recursos de forma adaptada a las diferentes necesidades de los alumnos. Se usa también para la realización de trabajos colaborativos en clase; para la corrección colectiva de ejercicios en clase, para la búsqueda de respuestas a preguntas no previstas;

para el aprendizaje del manejo de programas informáticos o para lograr interacciones directas del alumno sobre su medio.

1.4. PROBLEMAS ACTUALES

Muchas son las ventajas que proporciona la PDI en las aulas de clase, pero su utilización por parte de los profesores y los estudiantes conlleva algunas exigencias:

- Ante todo, el centro debe contar con unas infraestructuras adecuadas. El uso de la pizarra digital por parte del profesorado debe resultar cómoda. Si puede ser, las pizarras digitales estarán fijas en las clases o situadas en aulas específicas que los profesores podrán reservar cuando les convenga.
- Todo el profesorado debe recibir una formación didáctico-tecnológica. Esta formación debe asegurar a cada docente la adquisición de las habilidades necesarias para el manejo del equipo, el conocimiento de algunos modelos de utilización y el conocimiento de algunos materiales multimedia de apoyo para su asignatura.
- Apoyo decidido de la dirección del centro a la implementación de este sistema didáctico-tecnológico.
- Voluntad de cambio y mejora de la práctica docente por parte del profesorado, que algunas veces deberá invertir tiempo en buscar recursos y elaborar nuevos materiales didácticos.

Por otra parte, la PDI también conlleva algunas problemáticas que es necesario considerar para reducir su impacto:

- Alto coste del equipo.
- Hay que prever una buena luminosidad (según iluminación del aula) y resolución suficiente (según prestaciones del ordenador) del proyector usado.
- Si no hay una buena instalación fija con proyector en el techo o pantalla retroproyectada pueden surgir problemas logísticos con el cableado, aparición de sombras en la pantalla que dificulten su visión, etc.
- Se puede perder la calibración de la PDI si el proyector es móvil.
- Se requiere para aprovechar las prestaciones del software asociado y formar a los profesores en su uso.

Además, no podemos olvidar que se trata de una tecnología que ha surgido hace poco tiempo, por lo que actualmente el catálogo de recursos software específicos para la realización de actividades o juegos educativos sigue creciendo diariamente.

1.5. ESTRUCTURA DE LA MEMORIA

La memoria de este trabajo fin de grado se ha dividido en los siguientes bloques:

- **Motivación:** En este primer capítulo se ha descrito la motivación del proyecto, con una introducción al estado del arte sobre el tipo de tecnología que se trata en este Trabajo Fin de Grado. También se trata el uso de dicha tecnología en el aula, así como los problemas actuales derivados de ella.
- **Objetivo y diseño:** El segundo capítulo concreta el objetivo abordado en este trabajo fin de grado, así como el diseño de las nuevas soluciones de diseño y cuáles son las diferencias respecto a la versión anterior.
- **Implementación:** En este capítulo se comentan las decisiones que se han tomado para la implementación de la aplicación. Se incluyen diagramas sobre la aplicación y aspectos relativos a la interfaz de usuario.
- **Evaluación:** Se describen los detalles de la evaluación realizada por usuarios finales de la aplicación.
- **Conclusiones:** Para terminar, se presentan las conclusiones finales, obtenidas después de haber realizado todo el proyecto.
- **Bibliografía:** Referencias bibliográficas utilizadas para la realización de este Trabajo Fin de Grado.

2. OBJETIVO Y DISEÑO

El proyecto DEDOS²¹ tiene como objetivo que los profesores puedan crear de una forma sencilla e intuitiva actividades educativas para distintos dispositivos. Fruto de este proyecto surgieron dos aplicaciones: DEDOS-Editor y DEDOS-Player. El editor permite crear distintos tipos de actividades y que sean ejecutadas en distintos dispositivos usando DEDOS-Player, programa que realiza la adaptación automática de los espacios de trabajo definidos en las actividades en función del número de alumnos que se encuentre interactuando con el dispositivo. Sin embargo, estos recursos necesitan de un tratamiento específico para ser mostrados en una pizarra digital ya que DEDOS-Player se diseñó para que en el caso que hubiera múltiples jugadores estuvieran en superficies horizontales y no verticales. Por tanto el diseño está afectado en cuanto a la disposición y orientación de los elementos mostrados cuando se cambia de dispositivos horizontales a verticales como es el caso de la PDI. Con la realización de este Trabajo Fin de Grado (en adelante TFG), se busca que un recurso generado por un profesor en el editor pueda ser mostrado indistintamente en una mesa multicontacto y en una pizarra digital gracias a esta evolución del software.

Para ello, se va a modificar el código fuente de la aplicación DEDOS-Player, añadiendo funcionalidad que permita la correcta y cómoda visualización e interacción de cualquier recurso generado en DEDOS en una pizarra digital, sin perder ninguna de las ventajas con las que cuenta la versión para mesas multicontacto.

En este capítulo se va a tratar cómo se ha abordado las diferencias en cuanto al diseño e interacciones de dispositivos horizontales como las mesas multicontacto a dispositivos verticales como la PDI. Para ello, se expondrá de forma detallada la base de la que partimos con DEDOS-Player, y el punto al que queremos llegar para conseguir la correcta visualización de recursos educativos generados con el editor tanto en dispositivos verticales como horizontales.

²¹ Proyecto DEDOS: <http://aprendecondedos.es/>

2.1 MENÚ DE INICIO

Al iniciar DEDOS-Player (el reproductor de DEDOS), se abre un menú en el cual pueden seleccionarse las opciones que la aplicación ofrece para adaptar el proyecto educativo a las necesidades del alumnado y a los objetivos docentes del profesor (véase la Ilustración 6). En la pantalla inicial aparecen tres opciones principales: número de jugadores que van a participar en la actividad siendo cuatro el número máximo debido a los laterales de la mesa multicontacto, control de respuesta y dinámica de respuesta.

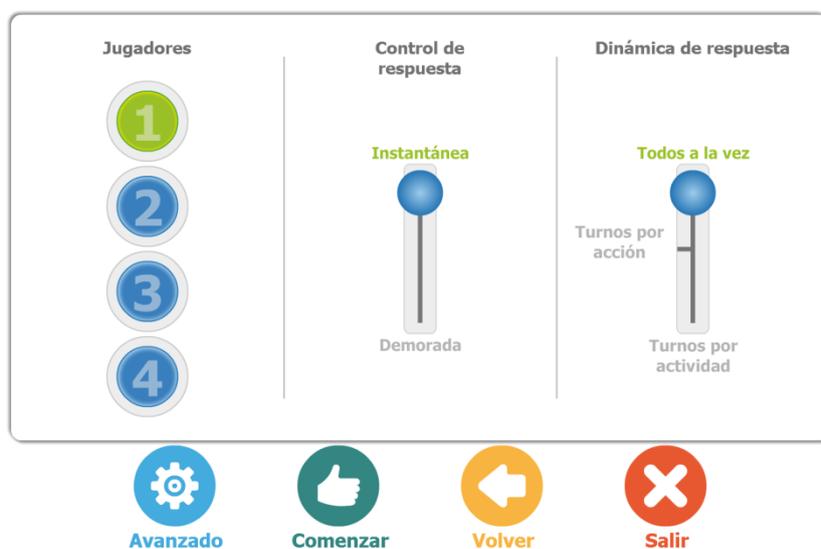


Ilustración 6 - Estado actual del menú de inicio de DEDOS-Player

En el control de la respuesta, el profesor puede seleccionar si se realizará de forma instantánea o demorada. Un control de respuesta instantánea comunica inmediatamente al jugador si su respuesta es correcta o no en cuanto selecciona la respuesta que él piensa que es correcta o realiza un emparejamiento. Si por el contrario elegimos un control de respuesta demorada, los jugadores conocerán si su respuesta es correcta cuando todos los jugadores hayan realizado la actividad y hayan confirmado que sus respuestas son definitivas.

La dinámica de respuesta ofrece tres opciones: todos a la vez, turnos por acción y turnos por actividad. Si se selecciona “*todos a la vez*”, todos los jugadores podrán participar simultáneamente en la actividad de forma independiente al resto de sus compañeros. Si se elige “*turnos por acción*”, el juego irá por turnos, es decir, todos los

jugadores a excepción del jugador que tenga el turno no podrán interactuar en la actividad. La diferencia entre esta opción y la siguiente, “*turnos por actividad*”, radica en que en la opción “*turnos por acción*” el turno va cambiando por cada una de las acciones que los alumnos tienen que realizar en la actividad y en la opción de “*turnos por actividad*” cada alumno hace una actividad completa y cuando termina el turno pasa al siguiente.

Aparte, debajo de estas opciones, DEDOS-Player ofrece una serie de botones que llevan a las opciones más avanzadas, al inicio de la actividad, permiten volver atrás para seleccionar otro archivo distinto de actividad, y ofrecen la posibilidad de cerrar la aplicación pulsando sobre el botón de salir.

Dentro de las opciones principales de esta pantalla inicial, sería necesario añadir una nueva columna que defina la orientación del dispositivo donde los alumnos van a realizar el proyecto educativo: horizontal (para mesas multicontacto, ya que la mesa se dispone en el eje X) o vertical (para pizarras digitales, ya que la disposición del dispositivo se encuentra en la pared o eje Y). Esta columna debe seguir el mismo “*look & feel*” existente de “*palancas*” que puede verse en todas las columnas. Deberá ser, por tanto, una palanca con dos opciones, semejante al “control de respuesta”. Además, dado que será una de las opciones más usadas, es conveniente colocar esta columna justo después de la opción de “*jugadores*”, desplazando las columnas “*control de respuesta*” y “*dinámica de respuesta*” hacia la derecha. Así, el espacio de opciones pasará de tener tres columnas a tener cuatro. También deberá añadirse una línea vertical más que actúe de separador entre columnas, tal y como lo están ahora.

Desde el punto de vista de usabilidad de la herramienta, como se puede ver en la Ilustración 6, para cada columna hay una opción que se encuentra predefinida en verde. En nuestro caso, para opción correspondiente a la orientación del dispositivo en el que trabajarán los alumnos, la opción que va a quedar predeterminada va a ser la opción vertical. El motivo de esta elección está basado en la frecuencia ya que actualmente la mayoría de los colegios sí disponen de PDIs, pero no disponen de mesas multicontacto. Por tanto, la orientación vertical será el valor por defecto de esta nueva variable.

De esta manera, se ha realizado un prototipo del aspecto final que tendría el menú inicial tal y como se muestra en la Ilustración 7. Se ha utilizado Photoshop para modificar el menú actual y añadir los nuevos cambios.

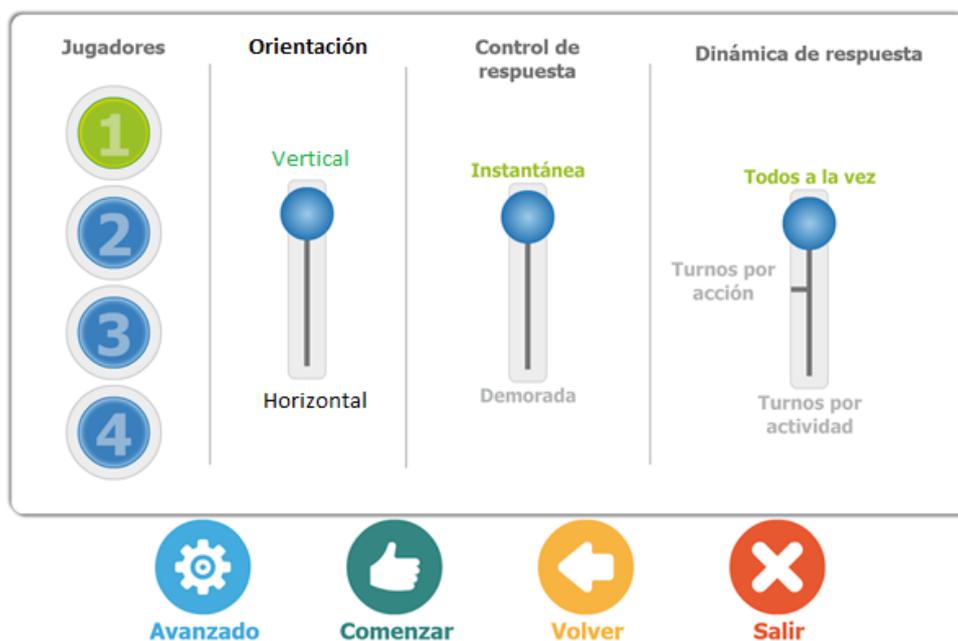


Ilustración 7 - Prototipo inicial del menú de DEDOS para PDIs

2.2 REPRODUCTOR

Una vez seleccionadas las opciones en el menú inicial, los alumnos pueden comenzar a realizar las actividades del proyecto educativo. La Ilustración 8 muestra un ejemplo de una actividad donde están interactuando cuatro jugadores en una mesa multicontacto donde se observan cuatro áreas, una para jugador, y que son las zonas de interacción de cada jugador. Tienen reflejado el número de jugador correspondiente en la esquina superior izquierda de cada tarjeta. En el centro podemos ver una pequeña área de interacción que sería aquella donde se situaran los elementos comunes a todos los jugadores (p.e. en el caso de que resuelvan correctamente la actividad, si ya se puede pasar a la actividad siguiente del proyecto educativo, tarjetas donde tengan que emparejar elementos, etc.).

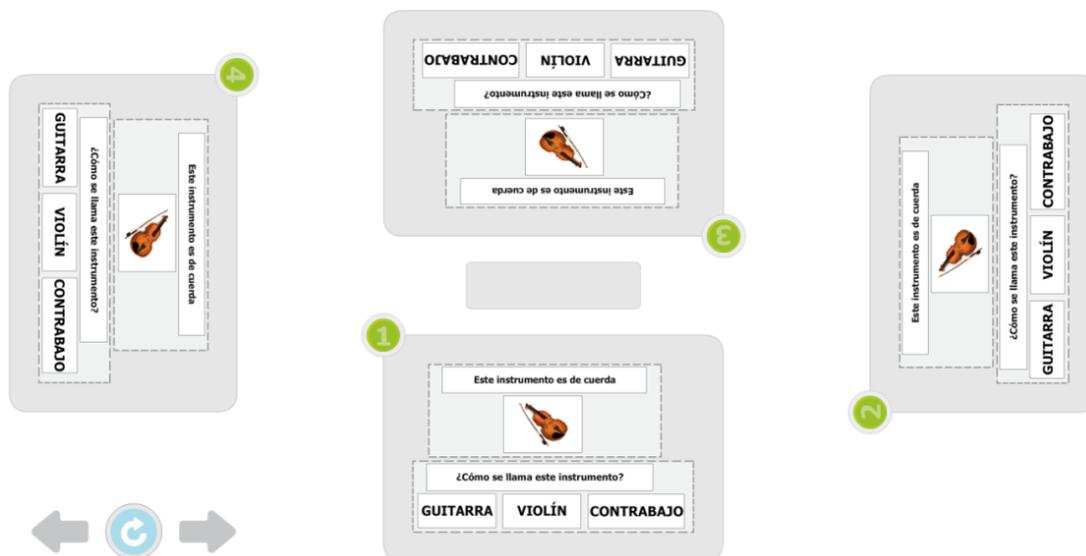


Ilustración 8 - Ejemplo de actividad en DEDOS para mesas multicontacto

El motivo de la ubicación de cada zona de jugador en la pantalla es para aprovechar las posibilidades que ofrece una mesa multicontacto. Así, cada jugador puede ocupar un lado de la mesa, siendo cuatro jugadores el máximo de jugadores que pueden jugar a DEDOS (véase la Ilustración 9).

Si se transportara esta imagen a una pizarra digital interactiva, resultaría inviable que cuatro jugadores jugaran a la vez, dado que el aspecto que presentaría sería el mismo que el de la Ilustración 8 pero en disposición vertical. Habría que girar la cabeza sólo para leer los textos que aparecerían al revés o rotados 90 grados. Por este motivo, es necesario hacer una rotación y recolocación de todas y cada una de las zonas para que queden en una posición horizontal, y dispuestas uniformemente a lo largo del eje X que permita la lectura de todos los textos que se muestran en pantalla.

Por tanto, si tuviéramos el proyecto educativo mostrado en la Ilustración 10 cuya vista se corresponde con una mesa multicontacto, sería necesario adaptarlo tal y como se representa en el prototipo de la Ilustración 11, el cual refleja la adaptación a PDI. Dicha adaptación tiene que tener en cuenta tanto la rotación de zonas como la recolocación de las mismas. Se trata de un ejemplo para una actividad en la cual participan únicamente dos jugadores, pero la idea que se tiene que realizar para el caso de tres y cuatro jugadores es análoga.

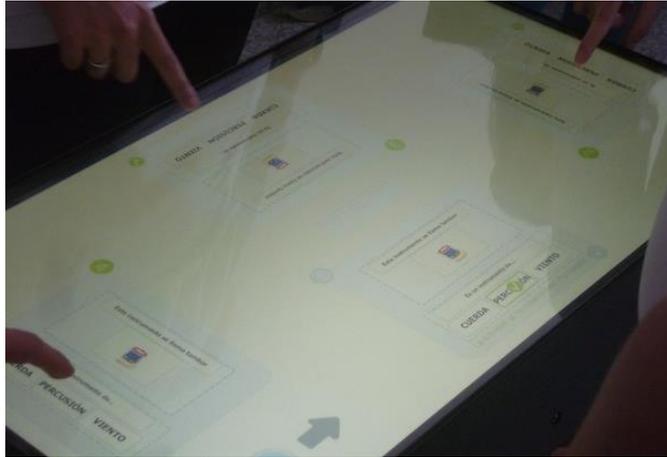


Ilustración 9 - Ejemplo de uso de DEDOS en una mesa multicontacto



Ilustración 10 - Aspecto de una actividad en DEDOS para dos jugadores en mesa multicontacto



Ilustración 11 - Prototipo inicial del reproductor DEDOS para PDIs con un ejemplo de actividad para dos jugadores

3. IMPLEMENTACIÓN

A lo largo de este capítulo se explicarán los pasos que se han seguido en el proceso de implementación a nivel de código de la actualización de la herramienta DEDOS-Player. Se explicarán las herramientas que han sido necesarias para desarrollar este TFG, las modificaciones que se han realizado al código fuente, así como un esquema orientativo acerca de cómo están jerarquizadas las clases que forman DEDOS-Player y su dependencia entre sí.

3.1 LENGUAJE UTILIZADO EN LA IMPLEMENTACIÓN

El lenguaje de programación con el que está implementada la herramienta DEDOS-Player es ActionScript, un lenguaje de programación orientado a objetos inventado por la empresa Adobe y empleado para la plataforma Adobe Flash. Surgió en 1977 para que los desarrolladores programasen de una manera más interactiva. La programación con ActionScript permite más eficiencia en las aplicaciones de la plataforma Flash para construir animaciones de todo tipo, pudiendo ser simples o complejas, manejando una buena cantidad de datos o interfaces interactivas.

La versión más extendida actualmente es ActionScript 3.0, que significó una mejora en el manejo de programación orientada a objetos al ajustarse mejor al estándar ECMA-262. Es utilizada en las últimas versiones de Adobe Flash y Flex, así como en anteriores versiones de Flex. ActionScript 3.0 mejora su rendimiento en comparación con sus antecesores e incluye nuevas características, como el uso de expresiones regulares y nuevas formas de empaquetar las clases. Aunque no se ha estudiado durante la carrera de Grado en Ingeniería Informática, sí resulta familiar a los desarrolladores con conocimientos sobre programación orientada a objetos. Algunas de las principales funciones de ActionScript 3.0 son:

- Una máquina virtual ActionScript, denominada AVM2, que utiliza un nuevo conjunto de instrucciones de código de bytes y proporciona importantes mejoras de rendimiento.
- Una base de código de compilador más moderna, que se ajusta mejor al estándar ECMAScript (ECMA 262) y que realiza mejores optimizaciones que las versiones anteriores del compilador.

- Una interfaz de programación de aplicaciones (API) ampliada y mejorada, con un control de bajo nivel de los objetos y un modelo orientado a objetos.
- Una API XML basada en la especificación de ECMAScript para XML (E4X) (ECMA-357 edición 2). E4X es una extensión del lenguaje ECMAScript que añade XML como un tipo de datos nativo del lenguaje.

Flash está compuesto por objetos, con su respectiva ruta dentro del swf (*Small Web Format* - formato de archivo de gráficos vectoriales creado por la empresa Macromedia, ahora propiedad de Adobe Systems). Cada uno de estos objetos en ActionScript pertenece a una clase (MovieClip, Botones, Vectores, etc.), la cual contiene propiedades y métodos. Las propiedades se encuentran dentro del archivo raíz de la clase declaradas como variables (*alpha, useHandCursor, length,...*). Los métodos están dentro del archivo raíz de la clase declaradas como funciones (*stop(), gotoAndPlay(), getURL(),...*).

3.2 HERRAMIENTAS UTILIZADAS

En el apartado anterior se ha detallado el lenguaje de programación en el que estaba programada la herramienta DEDOS-Player, y, por tanto, el lenguaje en el que se debe programar la nueva funcionalidad de la herramienta. En este nuevo apartado se especifican las herramientas que se han utilizado para la implementación de este proyecto:

- **Adobe Flash CS5.5:** Se trata de uno de los programas más famosos de Adobe, junto con Illustrator y Photoshop. Es una aplicación de creación y manipulación de gráficos vectoriales con posibilidades de manejo de código mediante el lenguaje ActionScript, especificado anteriormente. Flash es el entorno de desarrollo, y Flash Player es el reproductor utilizado para visualizar los archivos generados con Flash. En otras palabras, Adobe Flash crea y edita las animaciones o archivos multimedia y Adobe Flash Player las reproduce. Se ha utilizado la versión CS5.5 al ser la versión con la que estaba implementada la anterior versión de DEDOS-Player, causando ciertas incompatibilidades al compilar el código con una versión distinta a esta.
- **Notepad++:** Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Se parece al bloc de notas de Windows en cuanto

al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que son útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la Licencia Pública General de GNU. A pesar de que Adobe Flash CS5.5 ya incluye dentro del propio programa un editor de texto donde escribir el código con una interfaz bastante amigable, se ha elegido Notepad++ por una cuestión de costumbre y rutina a la hora de realizar las prácticas de la carrera. Se trata, pues, de una herramienta de la que se puede prescindir, pero que al conocer bastante bien, me ha permitido desarrollar la implementación con más fluidez.

- **Adobe AIR (*Adobe Integrated Runtime*):** Entorno de ejecución multiplataforma para la creación de aplicaciones RIA (*Rich Internet Applications*) que utilizan Adobe Flash, Adobe Flex, HTML y AJAX, para usarse como aplicaciones de escritorio. Adobe lo define como un entorno de ejecución que no necesita navegador para traer al escritorio las RIA, más que como una infraestructura corriente (*framework*) de aplicaciones. Una aplicación implementada en Adobe AIR requiere empaquetamiento, firma digital e instalación en el sistema del usuario. Se ha utilizado la versión de AIR 2.6, al ser la versión más reciente que ofrece Adobe Flash CS5.5.

3.3 ORGANIZACIÓN DE LA APLICACIÓN

Antes de especificar las adaptaciones que se han realizado a nivel de código, es conveniente hacer un esquema de cómo estaba desarrollada la aplicación DEDOS-Player, con la jerarquía de clases que forman la herramienta y las dependencias entre sí.

El primer reto que surgió al empezar la implementación fue ordenar las ideas en cuanto a la organización del código, con el objetivo de saber las clases concretas donde realizar las modificaciones pertinentes para lograr la nueva funcionalidad deseada.

La carpeta que contenía el código fuente de la aplicación contaba con 1.467 archivos, ordenados en 47 carpetas. Dada la complejidad a la hora de realizar un esquema, teniendo en cuenta todos estos archivos y dependencias, se pidió ayuda a uno de los autores iniciales de la aplicación DEDOS-Player, David Roldán Álvarez, el cual

facilitó su memoria de Trabajo Fin de Máster, en la cual venían especificadas todas las clases que forman la aplicación, así como las dependencias entre las mismas.

En la Ilustración 12 se puede observar un diagrama UML que especifica cómo están organizadas las clases de DEDOS-Player.

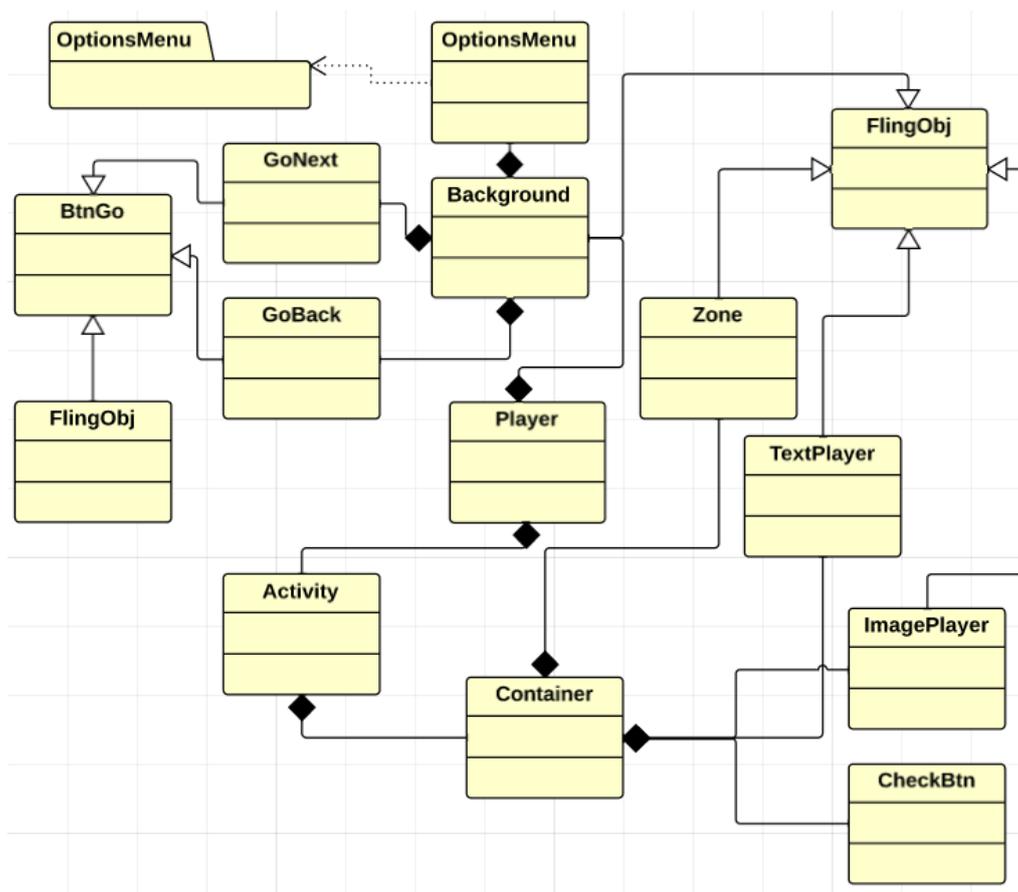


Ilustración 12 - Diagrama UML de clases de DEDOS-Player

Player es la clase principal, que contiene el *Background* y las actividades (*Activity*). Desde esta clase Player se van a gestionar todas y cada una de las acciones que se ejecuten en el reproductor en sí. Cuando se carga un proyecto que ha sido creado a través de DEDOS-Editor, se hace a través de la clase Player, almacenando las actividades definidas en el proyecto educativo.

La clase *Background* sirve para, a la hora de hacer alguna redimensión global de la aplicación, todos los elementos que aparezcan en el reproductor se redimensionen al mismo porcentaje. Esta clase contiene un menú de opciones (*OptionsMenu*) y los objetos de las clases *GoBack* y *GoNext*, los cuales sirven para avanzar o retroceder en la

lista de actividades del proyecto. Sin embargo, estos botones para avanzar o retroceder estarán activos o inactivos según las opciones que se hayan elegido en el menú inicial del reproductor.

Por otro lado, la clase *Activity* contiene la información relativa a cada una de las actividades que forman el proyecto. También verifica si las actividades se han resuelto correcta o incorrectamente. Cada una de estas actividades está formada por uno o más *Container* (contenedores), siendo estos contenedores quienes representan el espacio físico de cada jugador. Dependen, por tanto, del número de jugadores que esté usando DEDOS-Player.

Estos contenedores son los encargados de agrupar y mostrar cada elemento que se ha definido con DEDOS-Editor. En el editor, únicamente se diseña la actividad para un solo jugador (ver Ilustración 13), siendo el reproductor el que replica de manera automática la actividad tantas veces como estudiantes se elijan en las opciones del menú inicial.

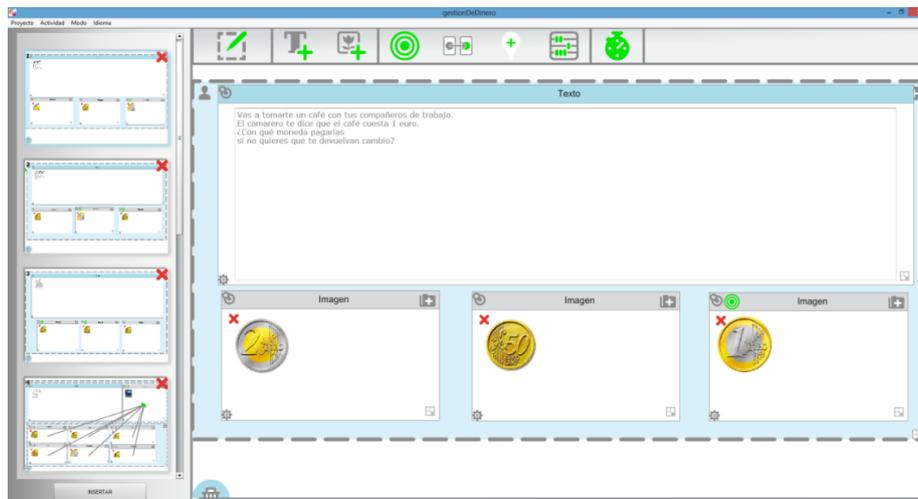


Ilustración 13 – Ejemplo de actividad creada con el editor de DEDOS

Si al crear una actividad en DEDOS-Editor se especificó que el modo de mostrar los mensajes de retroalimentación es manual, se añade una instancia de la clase *CheckBtn* a los contenedores de jugador que se crean en DEDOS-Player. Esto hace que no se verifique si la actividad se ha resuelto satisfactoriamente hasta que el jugador pulse sobre dicho botón.

Las clases *Zone*, *TextPlayer* e *ImagePlayer* representan a las zonas, textos e imágenes que se han definido en DEDOS-Editor. Las instancias de la clase *Zone* permite al jugador mover las tarjetas en el reproductor. Por otro lado, las instancias de la clase *TextPlayer* representan texto descriptivo con el que se informa al jugador sobre lo que debe hacer en la actividad. Esto se corresponde con las tarjetas de texto de la herramienta de edición de DEDOS, DEDOS-Editor. Los elementos *ImagePlayer* serán los elementos con los que se realizará interacción para resolver las actividades, y representan las tarjetas de tipo imagen.

El paquete *OptionsMenu* contiene la implementación del menú inicial del reproductor. Estas opciones se han descrito anteriormente en el apartado 2.1, y son el número de jugadores, dinámica de respuesta, control de respuesta, etc. La Ilustración 14 muestra un nuevo diagrama UML que indica la jerarquía de clases relacionada con las opciones.

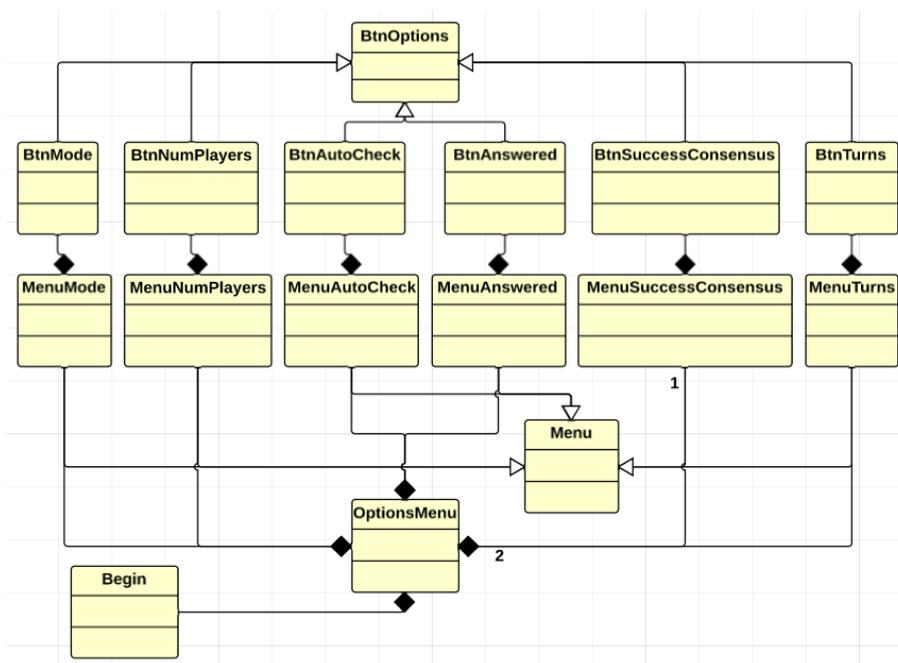


Ilustración 14 - Diagrama UML del paquete OptionsMenu

En la Ilustración 14 puede observarse como todas las opciones heredan de la clase *BtnOptions*. Cada una de estas clases hijas representan los botones que hay en los diferentes submenús de la pantalla inicial, donde se eligen las opciones que se desean. Además, todos los menús heredan de la clase *Menu*. Cada instancia de la clase *OptionsMenu* está compuesta por un submenú de cada tipo y representa las diferentes opciones del reproductor.

Los diagramas UML expuestos anteriormente para describir el funcionamiento de la herramienta DEDOS-Player han sido generados por la herramienta web *LucidChart*, la cual permite generar distintos tipos de diagramas directamente online²², sin necesidad de instalar ningún programa.

3.4 ADAPTACIONES REALIZADAS

Una vez descrito el funcionamiento de la aplicación, así como la organización de las clases y archivos, se describirán en este apartado las modificaciones realizadas a nivel de código para conseguir el objetivo de este TFG.

Para empezar, y como se describió en el apartado 2.1, se necesita modificar el menú inicial para implementar una nueva columna que permita elegir entre una orientación vertical (para PDIs) y otra horizontal (para mesas multicontacto).

Lo primero que se ha hecho ha sido declarar una nueva variable de tipo *Menu2GearStick*, clase que genera la “palanca” de dos posibles opciones, que es la que se necesita para no romper con el estilo de palancas que ya está establecido en el menú y seguir el mismo estilo en la apariencia que tenía la aplicación. Definimos el título de la columna, “*Orientación*”, y también sus dos posibles valores: “*Vertical*” y “*Horizontal*” en nuevas variables en el archivo *Constants.as*, archivo en el que se almacenan todas las constantes. Además, en el archivo “*OptionsMenu.as*” añadiremos una línea vertical que actúe de separador entre la columna de jugadores y la nueva columna de orientación.

Modificando el orden de los elementos en el archivo “*OptionsMenu.as*” para colocarlos en el orden correcto, se desplazaron las dos últimas columnas a la derecha, encajando la nueva columna después de la columna de jugadores. Así, se encontró el primer error que trastoca los planes que estaban establecidos respecto al diseño del prototipo inicial.

Como puede verse en la Ilustración 15, se ha añadido la nueva columna “*Orientación*”, con las dos opciones que puede adquirir, siendo “*Vertical*” la opción predefinida, como se especificó en el diseño del prototipo. Antes el espacio del menú estaba repartido en tres columnas, pero ahora son necesarias cuatro columnas. Los separadores están colocados a 1/4, 2/4 y 3/4 del eje X respecto a la ventana de las

²² *LucidChart*: <https://www.lucidchart.com>

opciones. Debido a que ahora cada columna cuenta con menos espacio que antes de tener la nueva funcionalidad, el último separador corta el texto de la opción “*Turnos por acción*” de la columna “*Dinámica de respuesta*”.



Ilustración 15 - Tercer separador corta una de las opciones de la última columna

Además, el texto de “*Dinámica de respuesta*” también cortaba con el separador al estar definido todo en una línea. Pudo repararse de manera sencilla, ya que pudo aplicarse al String del texto un valor verdadero a su atributo booleano `.wordWrap`, encargado del *line-break*, es decir, del salto de línea. De esta manera el texto deja de definirse en una única línea de texto.

Para la opción de la palanca fue algo más complejo aplicar ese line-break, debido a que se trataba de un objeto `BtnSelectableText`, y no un String directamente como en el caso del anterior. Se pensó en varias opciones, como aplicar saltos de línea en el texto, quedando cada palabra de la frase “*Turnos por acción*” en una línea distinta. Sin embargo, había demasiado espacio entre cada línea. Finalmente se modificó el ancho predefinido en el archivo “*Constants.as*” al texto de la opción, haciendo el line-break automáticamente, quedando como figura en la Ilustración 16, logrando el objetivo del prototipo inicial, aunque con los cambios descritos.



Ilustración 16 - Aspecto final del menú inicial

Además, también en el archivo “OptionsMenu.as”, se ha escrito un método nuevo, llamado *getOrientation()*, el cual recoge el valor que tome la palanca. Devuelve un valor booleano, siendo “*Vertical*” verdadero y “*Horizontal*” falso. Se ha tomado esta decisión a aspectos prácticos, ya que únicamente hay dos valores posibles que puede tomar la orientación, al igual que una variable booleana.

Al realizar unas sencillas pruebas de depuración con el método *trace()*, el cual permite mostrar por la consola de salida de Adobe Flash el valor de una variable, se pudo constatar que el valor de la nueva variable booleana era el esperado según se eligieran los dos distintos valores de la palanca “*Orientación*”.

Teniendo el menú inicial acabado, se pasó al siguiente paso: definir las variables necesarias para transportar este valor booleano a lo largo de las clases que fueran determinantes a la hora de hacer la rotación de las zonas.

En la clase *Player* se definió la variable booleana “*orientation*”, que será utilizada después en el método *beginProject()*. Este método, como su nombre indica, es el encargado de comenzar con el proyecto, teniendo en cuenta todas las opciones que se hayan escogido en el menú inicial. Es por esto que era importante guardar el valor de la orientación, junto con el número de jugadores, tipo de control de respuesta, tipo de dinámica de respuesta, etc. Además dentro de dicho método, para cada actividad se llama al método *.setupActivity()*, al cual se pasaban por parámetros las variables

descritas. Por tanto, debíamos añadir a la lista de parámetros la nueva variable que define la orientación de la actividad.

La clase *Activity* es la clase en la que más se trabajó, siendo en esta clase donde produciríamos la rotación de las tarjetas. Como se hizo en la clase *Player*, se definió una nueva variable booleana “*orientation*”. Al haber modificado la salida de parámetros del método *setupActivity()* en la clase *Player*, se modificó la entrada de los mismos en la clase *Activity*, lugar donde está implementado el método, de manera que le llegará el valor de la variable “*orientation*”.

El método *getPlayerRotation()* especifica la rotación de cada tarjeta del jugador en función del número de jugadores que haya. Así, la zona común a todos los jugadores, en el caso de que hubiera dos jugadores, debía rotarse 90°, como puede verse en la Ilustración 17. En otro caso (para uno, tres o cuatro jugadores) no debía tener rotación, ya que los elementos comunes se encuentran correctamente orientados (ver Ilustración 18).

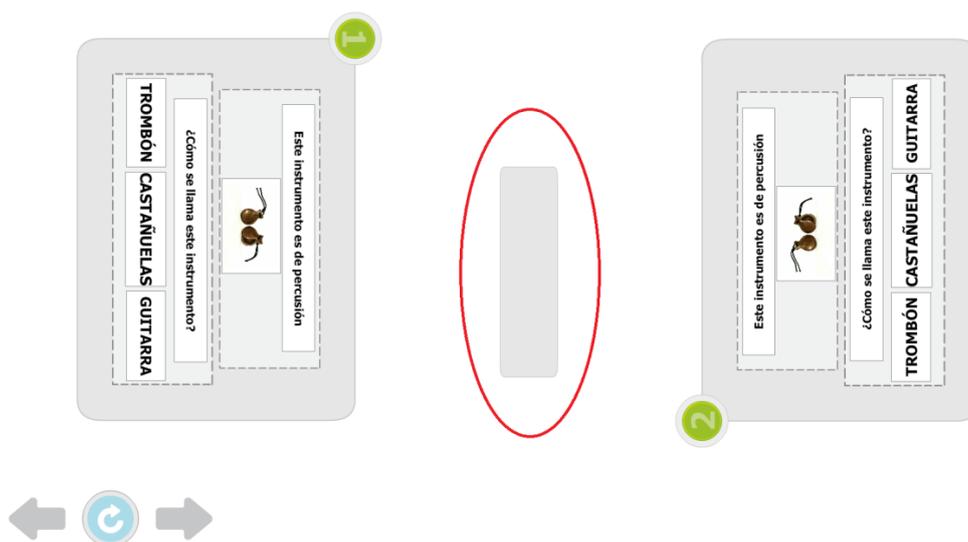


Ilustración 17 - Rotación de 90° a la zona común de jugadores para el caso de dos jugadores

En el caso de que estén jugando dos jugadores, los elementos individuales de cada jugador deben sufrir una rotación, como puede verse en la Ilustración 17. El jugador número uno debe rotar 270°, mientras que el jugador número dos debe rotar 90°.

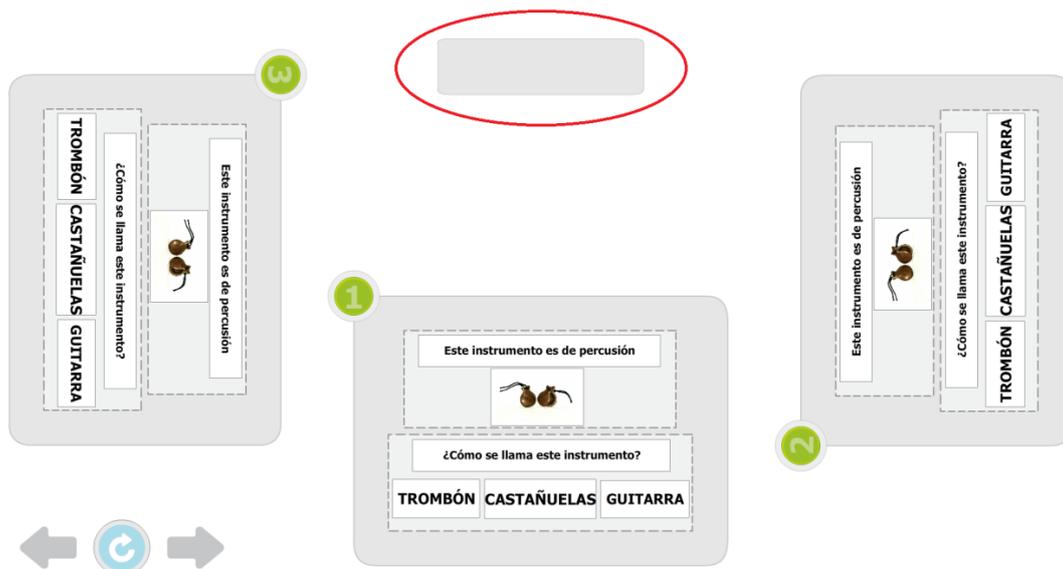


Ilustración 18 - Sin rotación en la zona común de jugadores para el caso de tres jugadores

En el caso de tres o cuatro jugadores, las adaptaciones que hay que realizar de los elementos incluidos en las áreas individuales de cada uno son las siguientes:

- El espacio de trabajo del estudiante número uno no sufre rotación, ni cuando interactúan tres jugadores ni cuando interactúan cuatro jugadores.
- El espacio de trabajo del estudiante número dos debe rotar 90°, tanto para tres como para cuatro jugadores.
- En el caso del tercer espacio de trabajo, su rotación depende del número de jugadores: se rotará 270° en el caso de tres alumnos interactuando para resolver la actividad, y 180° cuando fueran cuatro alumnos interactuando (véase la Ilustración 18 y la **¡Error! No se encuentra el origen de la referencia.**).
- Por último, el espacio de trabajo del cuarto alumno, será rotado 270° (ver **¡Error! No se encuentra el origen de la referencia.**), en el caso que exista.

Es necesario reseñar que, en el código de la aplicación, el estado normal de una tarjeta de jugador es el que se refleja en un proyecto con un único jugador, es decir, no tiene rotación (0°). Las tarjetas no rotan en PDI, cambian su disposición únicamente en mesas multicontacto. Cuando la variable “*orientation*” toma valor verdadero, es decir, el dispositivo que se está utilizando es una PDI, todas las tarjetas tienen una rotación de 0°. Es cuando adquiere valor falso cuando se aplican las rotaciones, algo que no se ha modificado, ya que ya estaba contemplado en el código para su correcto funcionamiento en mesas multicontacto antes de la realización de este proyecto.

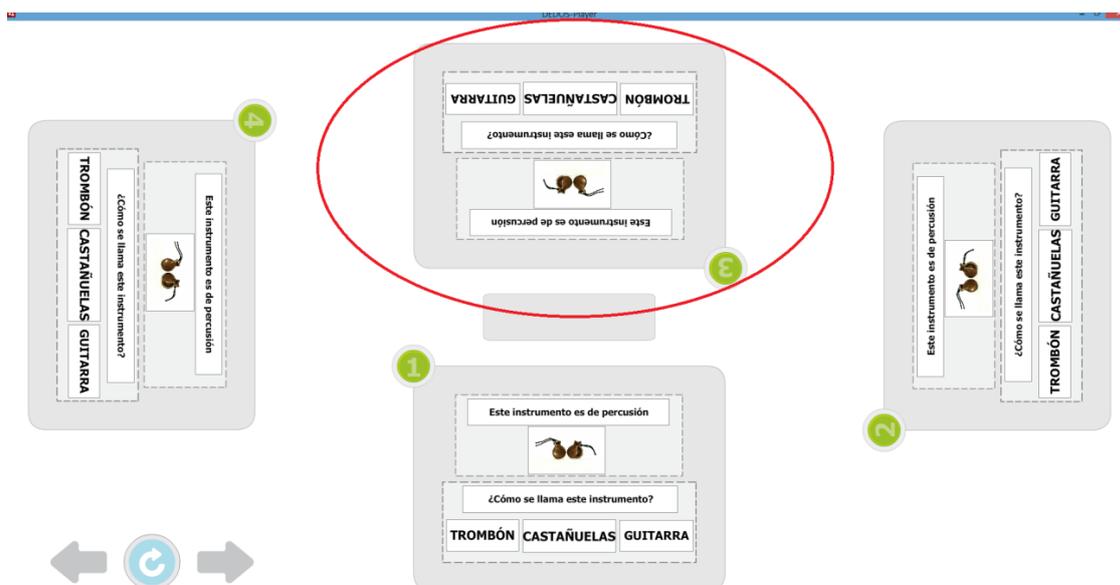


Ilustración 19 - Rotación del jugador 3 en el caso de cuatro jugadores jugando simultáneamente

Por tanto, se ha modificado este método *getPlayerRotation()*, de manera que para cada jugador debemos tener en cuenta la nueva variable definida “*orientation*”. Si toma valor falso (orientación horizontal) debe mantener la rotación ya definida, mientras que si toma valor verdadero (orientación vertical), la rotación debe ser siempre de 0°, quedando tal y como el profesor lo diseñó con el editor para que el texto se pueda leer de forma correcta y sin dificultades de orientación para los alumnos.

El siguiente método que debe modificarse es el método *getSetupInfo()*, también implementada en la clase *Activity.as*. Este método es el que define la posición sobre los ejes X e Y de las tarjetas de jugador, el escalado del tamaño de la tarjeta teniendo en cuenta la resolución de la pantalla en la que se está ejecutando el proyecto, así como la rotación definida para cada jugador en el método descrito anteriormente, *getPlayerRotation()*.

Este método contempla individualmente los casos en los que juegan un solo jugador, dos jugadores, tres jugadores y cuatro jugadores. Para cada caso, establece para cada jugador las propiedades antes descritas. En el caso de un único alumno, no sería necesario ninguna adaptación de los espacios de trabajo ya que tanto en mesas multicontacto como en PDIs la representación visual sería la misma.

Aparte de la rotación es necesario reescalar el tamaño de los elementos que se presentan a los alumnos en los espacios de trabajo. En el caso de dos jugadores, se

calculan los píxeles de la altura, teniendo en cuenta *gameC.height* (la altura del contenedor de juego), y *playerC.height* (la altura de la tarjeta del jugador) multiplicada por dos, ya que son dos jugadores los que están jugando. De igual manera se calculan los píxeles de la anchura, comprobando si *gameC.width* (el ancho del contenedor de juego) es mayor que *playerC.width* (el ancho de la tarjeta del jugador), quedándose con el mayor de ambos. La Ilustración 20 muestra un esquema orientativo sobre la localización de los valores de *gameC* y *playerC* en mesas multicontacto.

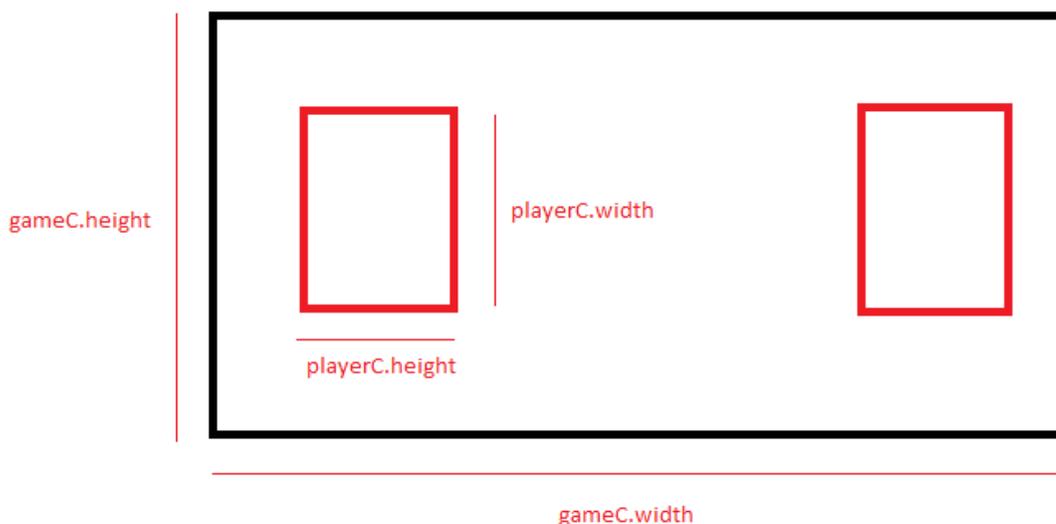


Ilustración 20 - Esquema de la anchura y altura de los contenedores de juego y jugador en mesas multicontacto

Teniendo en cuenta los píxeles horizontales y verticales calculados, se calcula una escala en la que el aspecto de las tarjetas de jugador encaje bien en el contenedor de juego, con un tamaño adecuado, de manera que las tarjetas no se salgan de la pantalla aunque las dimensiones de pantalla de distintas mesas multicontacto varíen.

Como en el caso de PDIs las tarjetas de jugador van a estar rotadas, es necesario modificar el valor de los píxeles horizontales y verticales cuando la variable “*orientation*” tome valor verdadero. Para los píxeles verticales, es necesario tener en cuenta *gameC.height* (la altura del contenedor de juego) y *playerC.width* (el ancho de la tarjeta de jugador) multiplicado por dos. Para los píxeles horizontales, se debe comparar *gameC.width* (el ancho del contenedor de juego) con *playerC.height* (la altura de la tarjeta del jugador), quedándose con el mayor de ambos. En la Ilustración 21 puede observarse un esquema similar al de la Ilustración 20, pero para PDIs.

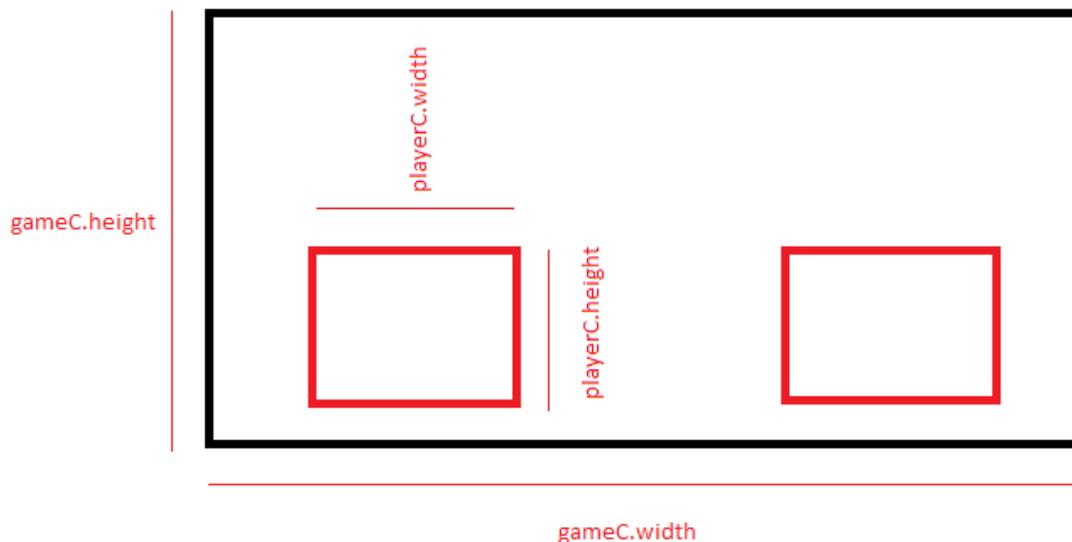


Ilustración 21 - Esquema de la anchura y altura de los contenedores de juego y jugador en pizarras digitales

Esta información se guarda en un vector de 4 posiciones. La primera posición la ocupa esta relación de escalado. En la segunda posición se encuentra la localización de la tarjeta en el eje X, mientras que en la tercera posición se encuentra la localización en el eje Y. Por último, en la cuarta posición se guarda el valor de la rotación de la tarjeta.

Una vez guardada la relación de escalado en la primera posición del vector, se debe calcular la posición en el eje X de la tarjeta. Esta posición se calcula de manera particular para cada tarjeta.

El espacio reservado para la agrupación de los elementos comunes a todos los jugadores siempre estará en la misma posición, tanto en el eje X como en el eje Y. Se ha dispuesto que se encuentre centrada respecto al eje X, y con un cierto margen respecto a la parte de arriba de la pantalla, de manera que no solape con las tarjetas de jugador, pero que no se encuentre muy cercana a la parte de arriba del contenedor de juego. Para este margen se ha utilizado una constante que ya se empleaba en el código, “*MARGIN_2P*”, y que se define como el 10% de la resolución de la pantalla en el eje X.

Adaptaciones de posición y escalado – Un jugador:

Para un único jugador, en principio, no debería haber ningún problema en dejar la misma posición y fórmula de escalado que en la mesa multicontacto, aunque solo podrá comprobarse si se necesita modificarse probando la nueva versión de la herramienta en la PDI.

Adaptaciones de posición y escalado – Dos jugadores:

En el caso en el que estén jugando únicamente dos jugadores, la posición en el eje X del jugador número uno se ha definido teniendo en cuenta el ancho de la pantalla. Dividiendo la pantalla en tres secciones, el jugador número uno quedaría en la posición 1/3, mientras que el jugador número dos ocupará la posición 2/3. Así conseguimos que los dos jugadores estén centrados respecto al eje X. A continuación, se multiplica por el valor del escalado.

En cuanto al eje Y, en cualquier caso las tarjetas de jugador van a encontrarse en la misma posición: teniendo en cuenta la altura del contenedor de juego, debemos restar la altura del margen inferior en el que están dispuestos los botones de ir hacia atrás, ir hacia adelante y actualizar actividad. Después, a esa cantidad se le restará una constante que definirá una posición en el eje Y que se encuentre por debajo del centro de la pantalla respecto a dicho eje Y. A continuación, al igual que con el eje X, se multiplica por el valor del escalado.

De esta manera, ya están definidos todos los valores en el vector: escalado, posición en eje X, posición en eje Y, y rotación de la tarjeta, cuyo valor se obtiene llamando al método *getPlayerRotation()*.

Así, ya se puede ejecutar la aplicación con orientación vertical para pizarras digitales para dos jugadores, teniendo este resultado (ver Ilustración 22), el cual se acerca bastante al prototipo inicial (ver Ilustración 11).



Ilustración 22 - Aspecto después de la implementación de la versión para PDIs para dos jugadores

Adaptaciones de posición y escalado – Tres jugadores:

En el caso de tres jugadores, la zona común para todos los jugadores, como se ha especificado anteriormente, va a permanecer igual que para el caso con dos jugadores, y permanecerá también igual para el caso de cuatro jugadores.

En cuanto al escalado, se va a tener en cuenta para los píxeles horizontales, el valor *gameC.width* (el ancho del contenedor de juego) sumado a *playerC.width* (el ancho de la tarjeta del jugador) multiplicado por tres, ya que hay tres jugadores.

Para los píxeles verticales se tendrán en cuenta dos valores. El primer valor será *gameC.height* (el alto del contenedor de juego) sumado a *playerC.height* (el alto de la tarjeta de jugador). El segundo valor será *playerC.width* (el ancho de la tarjeta de jugador). Comparando ambos valores, se tomará como valor final el mayor de los dos, y se almacenará en la primera posición del vector.

Para las posiciones en ejes X e Y, serán muy similares al caso de dos jugadores. La posición en el eje Y se quiere que sea la misma, para que en cualquier caso los jugadores se ubiquen siempre a la misma altura. En cuanto al eje X, si en el caso con dos jugadores se dividió el ancho de la pantalla en tres secciones, ahora se dividirá en cuatro, ubicando al jugador uno a 1/4 del eje X, al jugador dos a 2/4 y al jugador tres a 3/4; multiplicándolos después por el escalado.

En la Ilustración 23 puede verse el resultado de la implementación en la pantalla de 21'' de la que dispongo.

Aun así, dado que no dispongo de una pizarra digital para probar lo que se va implementando poco a poco, habrá que esperar a la evaluación de la aplicación en una pizarra digital para evaluar la disposición de las tarjetas, comprobando que la fórmula del escalado es la correcta.

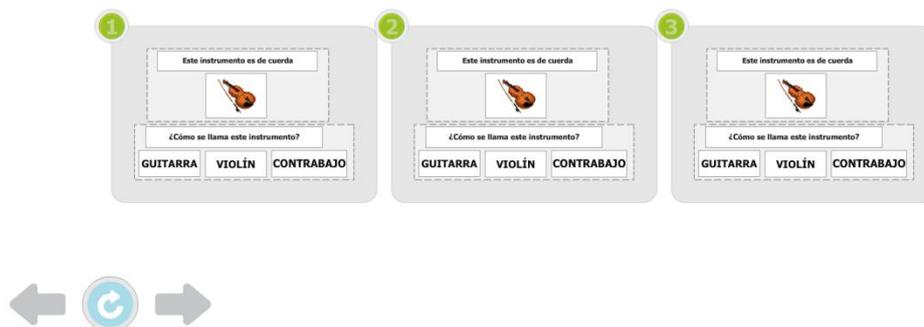


Ilustración 23 - Aspecto después de la implementación de la versión para PDIs para tres jugadores

Adaptaciones de posición y escalado – Cuatro jugadores:

El caso para cuatro jugadores es muy similar al de tres jugadores. En cuanto al escalado, se van a tener en cuenta para los píxeles horizontales *gameC.width* (el ancho del contenedor de juego) sumando a *playerC.width* (el ancho de la tarjeta del jugador) multiplicado por cuatro, pues son cuatro los jugadores que están jugando. Para los píxeles verticales, se van a tener en cuenta los mismos valores que para el caso de tres jugadores.

Para las posiciones de las tarjetas en el eje X, vamos a dividir el eje X en cinco secciones, colocando la primera tarjeta en la posición $1/5$, la tarjeta del segundo jugador en la posición $2/5$, el tercer jugador a $3/5$, y el cuarto a $4/5$. La posición en el eje Y será la misma que en los anteriores casos.

En cuanto a la rotación, como en todos los casos, será llamando al método *getPlayerRotation()*, que, para una orientación vertical, siempre devolverá 0° .

Así, quedarían rellenas las cuatro posiciones del vector, pudiendo ejecutarse un juego para cuatro jugadores para pizarras digitales, quedando como resultado final la Ilustración 24.



Ilustración 24 - Aspecto después de la implementación de la versión para PDIs para cuatro jugadores

Como puede observarse en la Ilustración 24, las tarjetas aparecen más pequeñas debido al poco tamaño de mi pantalla de ordenador, y, al igual que en el anterior caso, no conoceremos el aspecto final de la aplicación en una pizarra digital hasta el bloque de evaluación de la herramienta en una pizarra digital real.

A modo de resumen, a continuación se muestra una tabla que detalla las rotaciones en grados para cada jugador, dependiendo del número de jugadores que participen y del dispositivo que se esté usando:

Tabla 2 - Tabla resumen de rotaciones para cada jugador según el número de jugadores y dispositivo

	PDI					MESA				
	Jugador número:					Jugador número:				
	0 (zona común)	1	2	3	4	0 (zona común)	1	2	3	4
1 jugador	0	0				0	0			
2 jugadores	0	0	0			90	90	270		
3 jugadores	0	0	0	0		0	0	270	90	
4 jugadores	0	0	0	0	0	0	0	270	180	90

Además de las modificaciones descritas, se han modificado otros pequeños cambios que no influyen en la rotación de las tarjetas. Por ejemplo, en la clase LogManager.as, el método *writeProjectData()* escribe en un fichero .xml entre etiquetas el valor de las opciones elegidas en el menú inicial. Se ha añadido el valor de la nueva opción de orientación entre las etiquetas *<Orientacion>* *</Orientacion>*.

4. EVALUACIÓN

La evaluación de la herramienta sobre pizarras digitales interactivas se ha llevado a cabo en la FSDM (Fundación Síndrome de Down de Madrid²³) y en los laboratorios de la Universidad Rey Juan Carlos en el campus de Móstoles.

Para llevar a cabo la ejecución de la aplicación, debe compilarse la nueva versión para PDIs desde Adobe Flash Professional CS5.5, programa utilizado para todas las compilaciones del proyecto, tanto de prueba como final. La publicación desde Adobe resulta en un fichero .air, que puede instalarse como aplicación de escritorio en el ordenador que controla la PDI.

4.1 EVALUACIÓN DE LA HERRAMIENTA EN LA FSDM

La primera evaluación de la nueva versión de DEDOS-Player para PDIs tuvo lugar en la Fundación Síndrome de Down de Madrid, en una de las pizarras digitales que tienen instaladas en sus aulas. Al no conocer nada sobre cómo sería la pizarra (marca, modelo o dimensiones), al llegar se pudo constatar que, dadas las dimensiones de la pizarra así como su resolución, las tarjetas de jugadores se verían demasiado pequeñas. La pizarra digital tiene una relación de aspecto similar a los antiguos monitores de ordenador, es decir, pantallas con una forma más cuadrada (con una relación de aspecto 4:3) que las actuales, con una forma más rectangular (relación de aspecto 16:9). Mientras que la resolución más habitual de un monitor 4:3 es de 1024x768 píxeles, la de una de 16:9 es de 1920x1080 píxeles. Al hablar de la resolución de una pantalla, debe tenerse en cuenta que se trata del número de píxeles que pueden ser mostrados en ésta. Esto viene dado por la relación de aspecto, que es la proporción entre el ancho y el alto de la pantalla. Se calcula dividiendo el ancho de la imagen entre la altura (ver Ilustración 25 e Ilustración 26).

Si se tiene una resolución de 1024x768 píxeles, la primera cifra se refiere al ancho de pantalla (1024), y la segunda hace referencia al alto (768), medido en píxeles. Dividiendo 1024 entre 768, obtenemos 1.3, mismo resultado que dividir 4 entre 3, es por eso que la relación de aspecto de esta resolución es 4:3.

²³ Fundación Síndrome de Down de Madrid: <http://www.downmadrid.org/>



Ilustración 25 - Relación de aspecto 4:3²⁴



Ilustración 26 - Relación de aspecto 16:9²⁵

No se esperaba una resolución de 4:3 en una PDI, sino una de 16:9, que es la más habitual en monitores. Según un estudio de W3Schools.com²⁶ (página con información para desarrolladores web) que recopila las resoluciones de pantalla de usuarios de navegadores web (ver Tabla 3), en 2004 los monitores con resolución 1024x768 eran los más usados para utilizar navegadores de internet, con un 47%. Durante cuatro años esta tendencia se mantiene, pero en 2009 la resolución 1024x768 queda en un uso del 36%, mientras que otras resoluciones más altas ya alcanzan un porcentaje de uso del 57%. Con el paso de los años, la resolución 1024x768, que fue la más usada durante cinco años, ha ido disminuyendo a gran velocidad, hasta quedarse en la actualidad (Enero 2015) con un porcentaje de uso del 4%, mientras que la relación de aspecto 16:9 es ahora la predominante, siendo la más usada la resolución 1366x768, con un 33% de uso. Esta resolución es habitual en ordenadores portátiles.

²⁴ Fuente de imagen: <https://goo.gl/BMcf1o>

²⁵ Fuente de imagen: <https://goo.gl/36uBg7>

²⁶ W3Schools.com: <http://www.w3schools.com/>

Tabla 3 - Estadísticas de W3Schools.com de las resoluciones que utilizan los usuarios de navegadores web²⁷

Fecha	Más altas	1920x1080	1366x768	1280x1024	1280x800	1024x768	800x600	Más bajas
Enero 2015	32.7%	16%	33%	7%	5%	4%	0.3%	2%
Enero 2014	34%	13%	31%	8%	7%	6%	0.5%	0.5%
Enero 2013	36%	11%	25%	10%	8%	9%	0.5%	0.5%
Enero 2012	35%	8%	19%	12%	11%	13%	1%	1%
Enero 2011	50%	6%	-	15%	14%	14%	0%	1%
Enero 2010	39%	2%	-	18%	17%	20%	1%	3%
Enero 2009	57%	-	-	-	-	36%	4%	3%
Enero 2008	38%	-	-	-	-	48%	8%	6%
Enero 2007	26%	-	-	-	-	54%	14%	6%
Enero 2006	17%	-	-	-	-	57%	20%	6%
Enero 2005	12%	-	-	-	-	53%	30%	5%
Enero 2004	10%	-	-	-	-	47%	37%	6%
Enero 2003	6%	-	-	-	-	40%	47%	7%
Enero 2002	6%	-	-	-	-	34%	52%	8%
Enero 2001	5%	-	-	-	-	29%	55%	11%
Enero 2000	4%	-	-	-	-	25%	56%	15%

En este otro estudio de StatCounter²⁸ (ver Ilustración 27) se indica que la resolución más utilizada del mundo de 2013 a 2014 es también la de 1366x768, como muestra el estudio de W3Schools, algo indicativo también de que la gran mayoría de usuarios de ordenador utilizan un ordenador portátil. Sin embargo, esto no ocurría en PDI.

²⁷ Estadísticas de resoluciones que utilizan usuarios de navegadores, según W3Schools.com: <http://goo.gl/6DFZS2>

²⁸ StatCounter, GlobalStats: <http://gs.statcounter.com/>

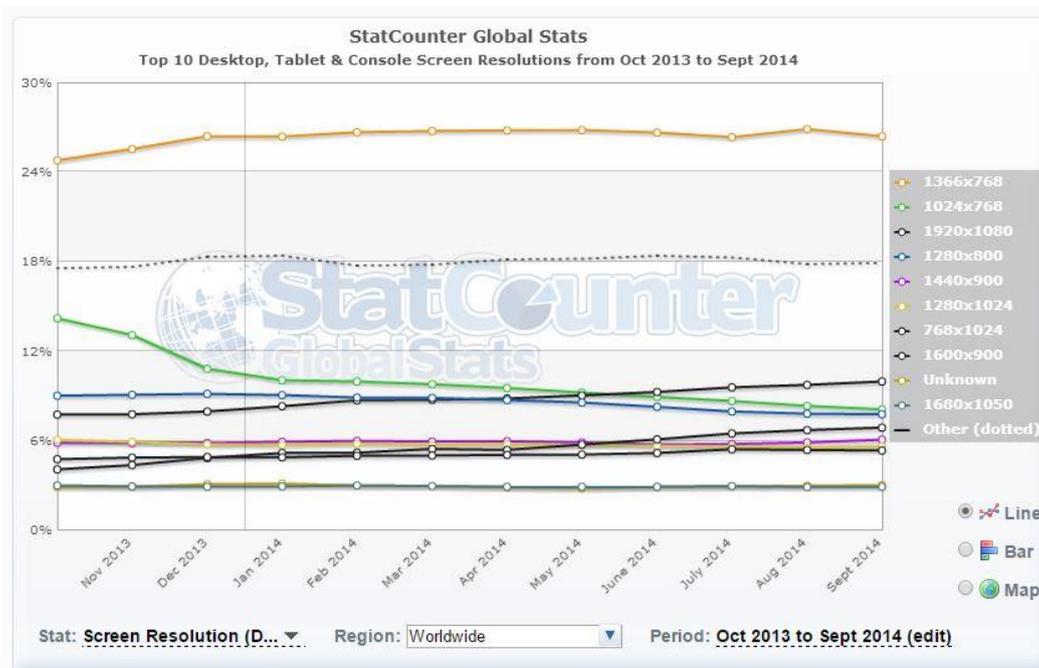


Ilustración 27 - Estadísticas de uso de resoluciones de pantalla desde noviembre de 2013 a septiembre de 2014, según StatCounter

La PDI en la que se realizó la evaluación de la versión 0.1 de DEDOS-Player era un modelo Board²⁹ de la marca SMART Tech³⁰. Así, una vez instalada en el ordenador de la Fundación Síndrome de Down de Madrid la primera versión (0.1) de la herramienta DEDOS-Player para PDIs, se comprobó que la modificación del menú inicial se veía correctamente, a excepción del texto de la última columna, que recortaba las palabras al haber añadido una columna más al menú y disponer de menos espacio (ver Ilustración 28). Era un fallo con una solución sencilla, había que dotar a esos campos de texto un poco más de espacio.

Al abrir un proyecto y probar cómo se verían las tarjetas para los jugadores, se pudo comprobar que la rotación de las tarjetas era correcta, para dos, tres y cuatro jugadores, todas se veían correctamente en disposición horizontal y centradas, con el mismo margen a derecha e izquierda. Pero la fórmula de escalado que se había pensado había que modificarla, ya que tenía mucho en cuenta el eje X, y al ser mucho más pequeño ese eje en la pizarra de lo que se esperaba debido a la relación de aspecto, todo se veía excesivamente pequeño (ver Ilustración 29 e Ilustración 30), siendo el texto muy poco legible, tanto de cerca como de lejos.

²⁹ Especificaciones de SMART Board: <http://goo.gl/NwlyY8>

³⁰ SMART Tech: <http://home.smarttech.com/es-es>

Al no disponer de mucho tiempo en el aula y tener que volver a pensar en la fórmula de escalado de las tarjetas, no fue posible modificar el código en ese momento, solo probar algunos proyectos más para constatar que ese fallo se daba en todos los casos, tanto para dos, tres y cuatro jugadores.

También se pudo observar que, dadas las dimensiones de la pizarra, pensar en una actividad para cuatro jugadores resultaría muy complicado, dado que el espacio físico horizontal que ocupa la PDI no abarca el suficiente como para que puedan interactuar cuatro personas a la vez. Tendrían que estar muy juntas entre sí, demasiado para una interacción cómoda, o puede que ni siquiera cupiesen en ese espacio.

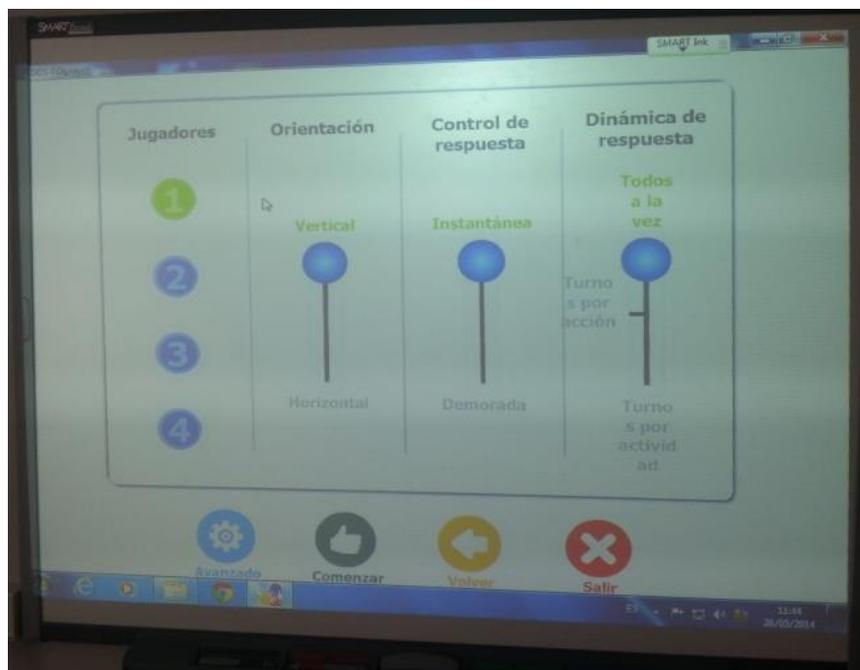


Ilustración 28 - Menú inicial de la versión 0.1 de DEDOS-Player para PDIs

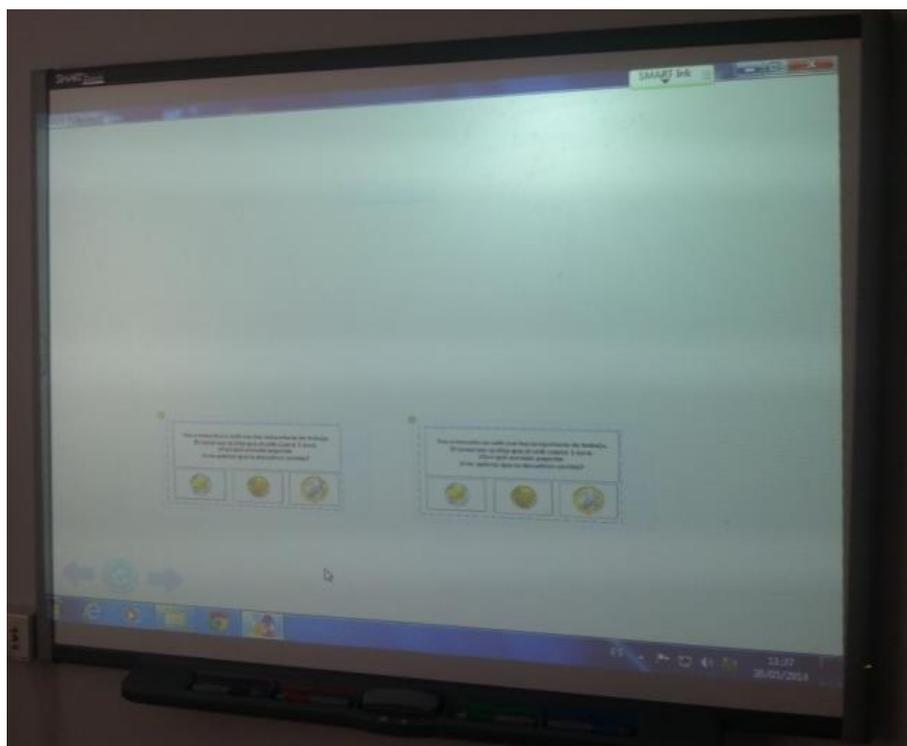


Ilustración 29 - Actividad para dos jugadores en la versión 0.1 de DEDOS-Player para PDIs

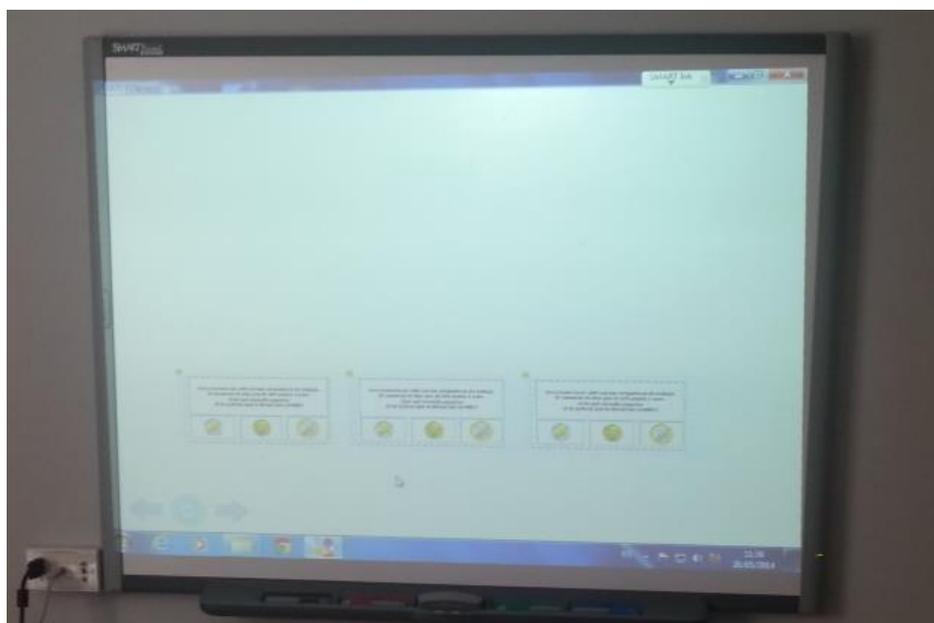


Ilustración 30 - Actividad para tres jugadores en la versión 0.1 de DEDOS-Player para PDIs

4.2 EVALUACIÓN DE LA HERRAMIENTA EN LA URJC

Una vez apuntados los fallos que debían corregirse de la versión 0.1 de la herramienta DEDOS-Player para PDIs, se comenzó a desarrollar la versión 0.2, para dar solución a dichos errores. El principal fallo y más importante era la función de escalado, la cual daba tamaño a las tarjetas de jugadores.

Se modificó la fórmula de escalado para dotar de más dimensiones a las tarjetas, pero dado que no dispongo de una PDI a mano para ver cómo varían los tamaños de las distintas compilaciones cuando se cambia el código, no es posible saber cómo será el resultado final hasta que se haga una nueva evaluación de la herramienta en una PDI.

La segunda evaluación de la herramienta DEDOS-Player se llevó a cabo en un aula de los laboratorios de la Universidad Rey Juan Carlos en el campus de Móstoles, el cual cuenta con una pizarra digital de la marca InterWrite, modelo SchoolBoard³¹. Esta PDI, al igual que la SMART Board de la Fundación Síndrome de Down, tiene una relación de aspecto 4:3. Echando un vistazo a modelos recientes de ciertos fabricantes, por ejemplo, SMART Tech, puede verse que éstos ya disponen de una relación de aspecto de 16:9³².

Haciendo una búsqueda rápida en Google Shopping³³ (servicio de Google que ofrece la posibilidad de comparar precios de productos en varias tiendas online) de “pizarras digitales”³⁴, pueden encontrarse distintas PDIs de distintos fabricantes, pudiendo apreciarse que las más caras son las que disponen de una relación de aspecto mayor y una mejor resolución, superando ampliamente de precio a las que no disponen de estas características (por ejemplo, una PDI Hitachi StarBoard FX89³⁵, lanzada al mercado en 2012, con relación de aspecto 16:10, tiene un precio de 992.20€ en el portal DeskIdea.com³⁶; mientras que una PDI ActivBoard 100Range³⁷, lanzada al mercado en 2009, con relación de aspecto 4:3, tiene un precio de 518.80€ en el portal 101gigas.com³⁸).

³¹ InterWrite SchoolBoard: <http://stg-inc.com/schoolboard.shtm>

³² Modelos recientes de SMART Board: <http://goo.gl/sGxaJg>

³³ Acerca de Google Shopping: <https://goo.gl/z1J2Zv>

³⁴ Búsqueda en Google Shopping de “pizarras digitales” (fecha de consulta 11/12/2015):

<https://goo.gl/eIOUKA>

³⁵ Hitachi StarBoard FX89: <http://goo.gl/aHw4zd>

³⁶ Hitachi StarBoard FX89 en DeskIdea.com (fecha de consulta 11/12/2015): <http://goo.gl/7Dyu1U>

³⁷ ActivBoard 100Range: <http://goo.gl/OE0yWT>

³⁸ ActivBoard 100Range en 101gigas.com (fecha de consulta 11/12/2015): <http://goo.gl/a6kAZn>

Esto hace pensar que las PDIs puede que estén siguiendo, guardando las distancias, una tendencia similar a la de los monitores de ordenador en cuanto a la relación de aspecto y resolución (ver Tabla 3 e Ilustración 27), por la similitud en su formato. Dentro de unos años se podrá estudiar con más detalle esta tendencia entre PDIs y monitores.

Al instalar la versión 0.2 de DEDOS-Player y abrir un proyecto, se pudo observar que, aunque las tarjetas habían aumentado su tamaño, no era suficiente. Seguía viéndose todo demasiado pequeño, con un texto muy poco legible para actividades para dos, tres y cuatro jugadores, sin aprovechar al máximo el tamaño que ofrece la pizarra (ver Ilustración 31, Ilustración 32 e Ilustración 33).

Se tuvo, pues, que volver a modificar la función de escalado. También, al igual que con la pizarra utilizada en la primera evaluación en la Fundación Síndrome de Down, pudo observarse que cuatro personas interactuando a la vez en la pizarra sería muy incómodo, ya que no hay prácticamente espacio físico para cuatro personas en la pizarra.

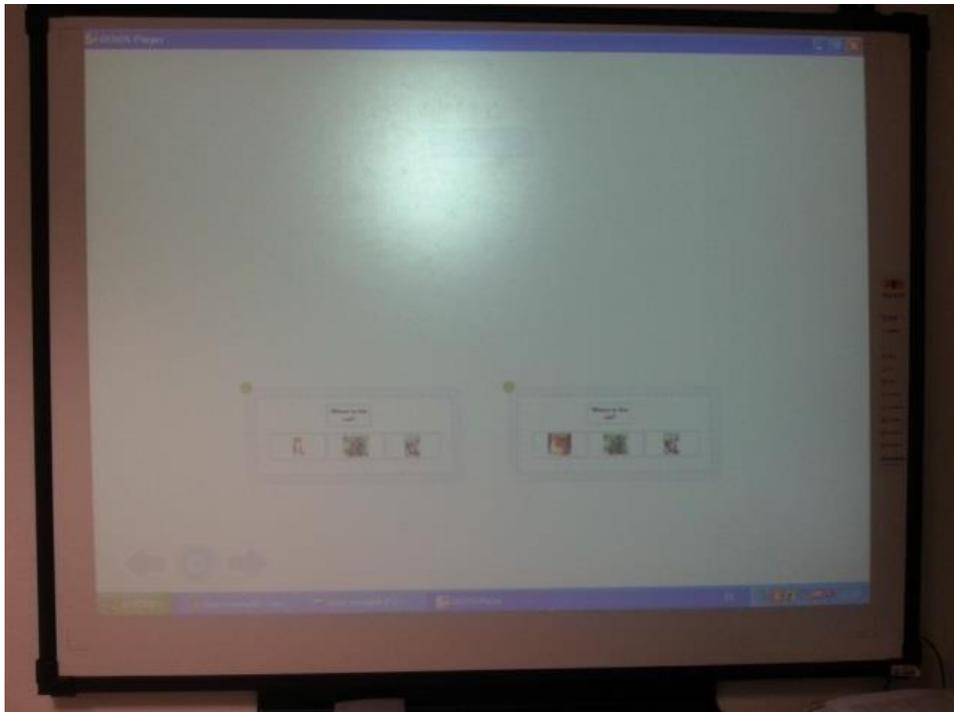


Ilustración 31 - Actividad para dos jugadores en la versión 0.2 de DEDOS-Player para PDIs

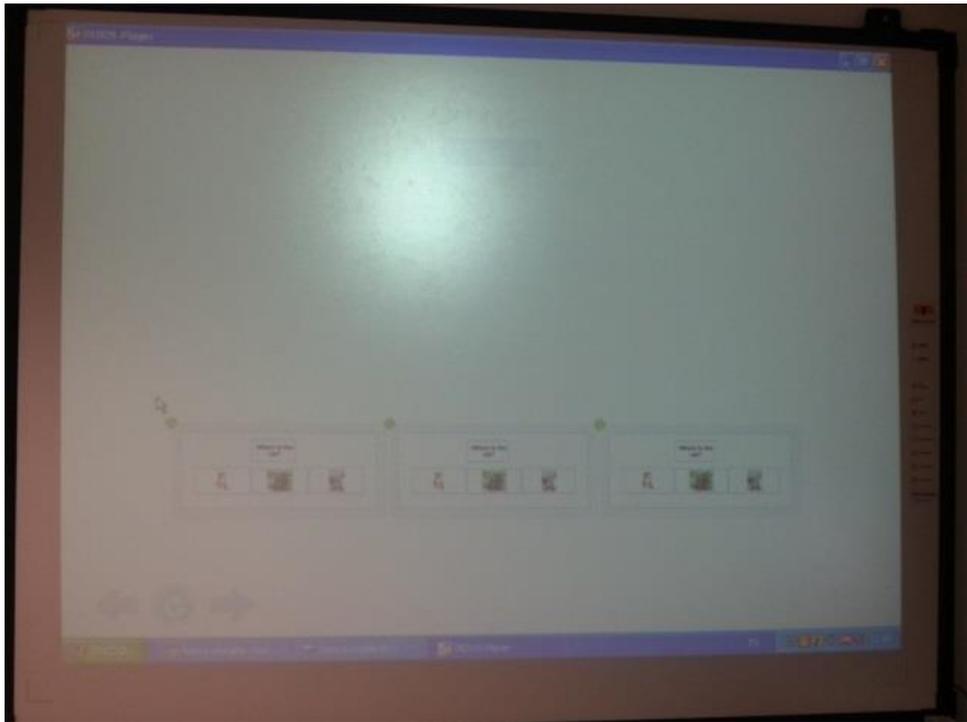


Ilustración 32 - Actividad para tres jugadores en la versión 0.2 de DEDOS-Player para PDIs

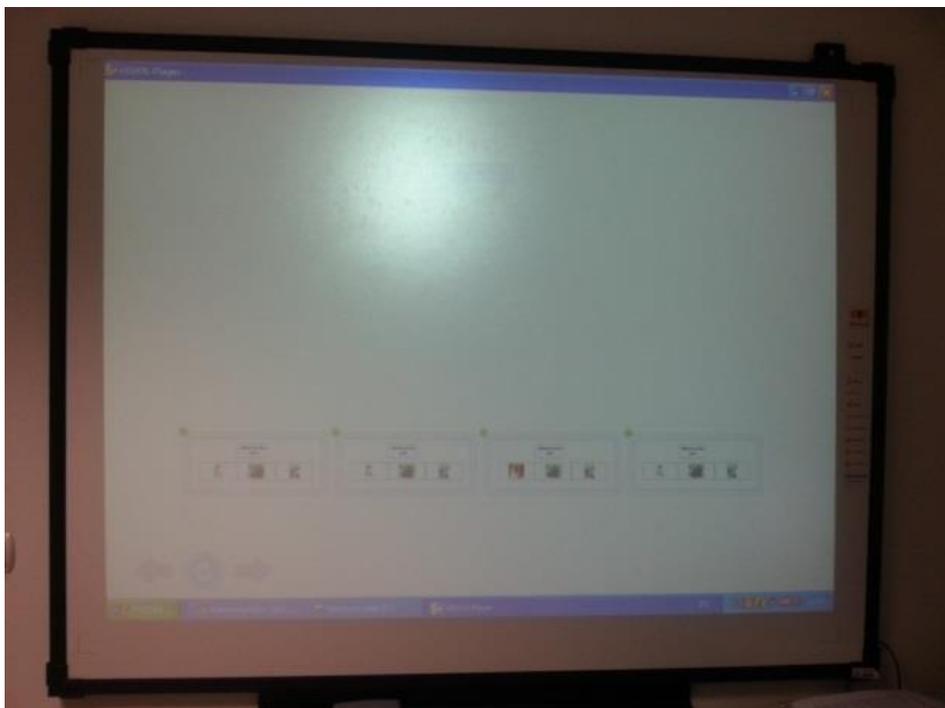


Ilustración 33 - Actividad para cuatro jugadores en la versión 0.2 de DEDOS-Player para PDIs

Quizá en una PDI con relación de aspecto más alta sea posible abarcar más espacio. Por ejemplo, la PDI de Hitachi serie StarBoard FX cuenta con dos modelos, la 79E1 y la 89WE1³⁹. Ambas cuentan con muchas similitudes, de hecho su hoja de especificaciones es prácticamente la misma, excepto que la 79E1 tiene una relación de aspecto de 4:3 y la 89WE1 tiene una de 16:10. Esto se traduce en distintas dimensiones: la de 4:3 tiene un ancho de 1'852 metros, mientras que la de 16:9 tiene mucho más ancho, 2'300 metros. Se trata de una diferencia de 448 cm, casi medio metro. Es, por tanto, muy probable que en una PDI con relación de aspecto más alta sea mucho más cómodo y efectivo utilizar la herramienta DEDOS-Player para cuatro jugadores.

Para la siguiente evaluación se llevaron preparadas varias versiones de DEDOS-Player con distintas fórmulas de escalado, ya que es complicado modificar el código en el propio laboratorio de la URJC. Como no se dispone de mucho tiempo en el laboratorio y el proceso de cambiar cualquier detalle tarda varios minutos, se optó por llevar preparados cinco ficheros distintos .air de casa, cada uno con distintas fórmulas de escalado tanto para dos, tres como para cuatro jugadores. De esta manera se ahorraba tiempo y podía verse fácilmente qué versión obtenía un mejor escalado.

La tercera evaluación de la herramienta volvió a llevarse a cabo en el mismo laboratorio de la URJC en Móstoles donde se hizo la segunda evaluación, es decir, con la misma PDI, InterWrite SchoolBoard. Se instalaron las cinco versiones que se llevaban preparadas (versiones 0.21, 0.22, 0.23, 0.24 y 0.25), y se tomaron las respectivas notas sobre cada uno de los tamaños de las tarjetas en la pizarra.

A pesar de que ninguno de los resultados fue óptimo, sí se pudo observar que las fórmulas de escalado que en un principio se habían pensado que derivarían en un tamaño quizá excesivo de las tarjetas aun podían aumentarse un poco más para aprovechar más espacio en la pizarra. Así, dado que se había ahorrado bastante tiempo al llevar varias versiones, se pudo modificar el código en el propio laboratorio, llevando a cabo un par de versiones extra (0.26 y 0.27), resultando la versión 0.27 la que aprovechaba prácticamente todo el espacio de la pizarra para dos, tres y cuatro jugadores, especialmente para estos dos últimos, que eran los más conflictivos, ya que agrupan más tarjetas en el eje horizontal (ver Ilustración 34, Ilustración 35 e Ilustración 36).

³⁹ Hitachi StarBoard FX-79E1 y FX-89WE1: <http://goo.gl/vGf1V2>

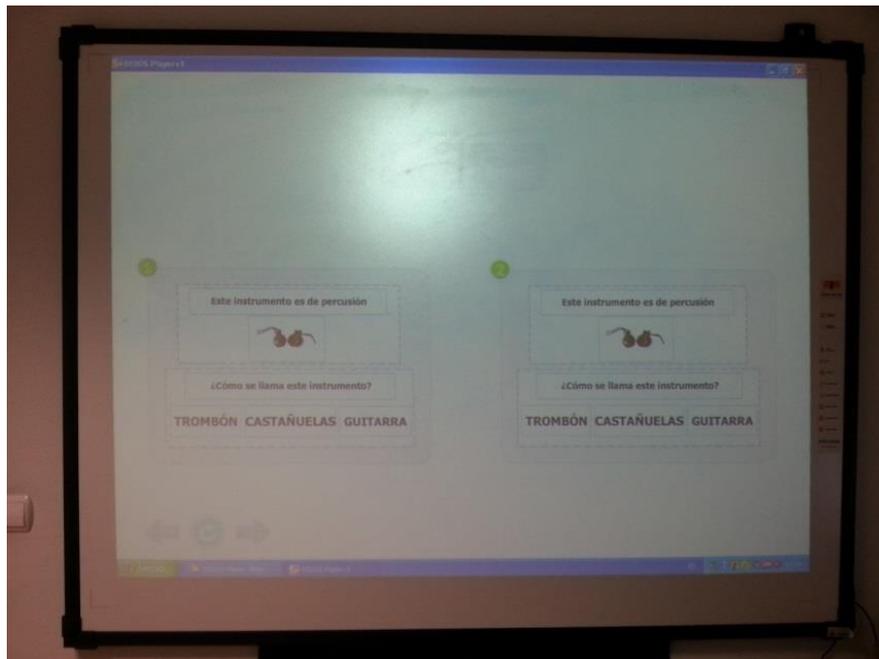


Ilustración 34 - Actividad para dos jugadores en la versión 0.27 de DEDOS-Player para PDIs

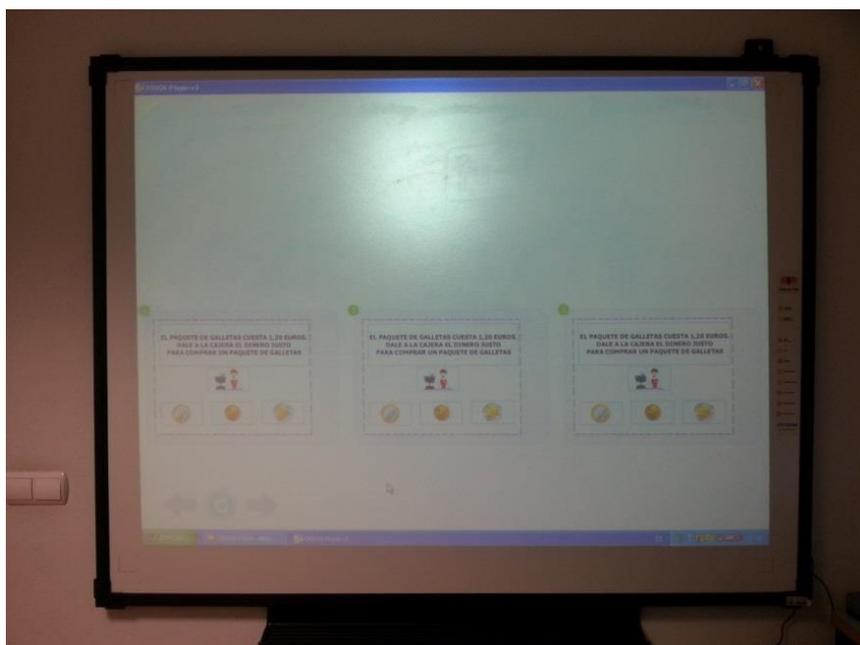


Ilustración 35 - Actividad para tres jugadores en la versión 0.27 de DEDOS-Player para PDIs

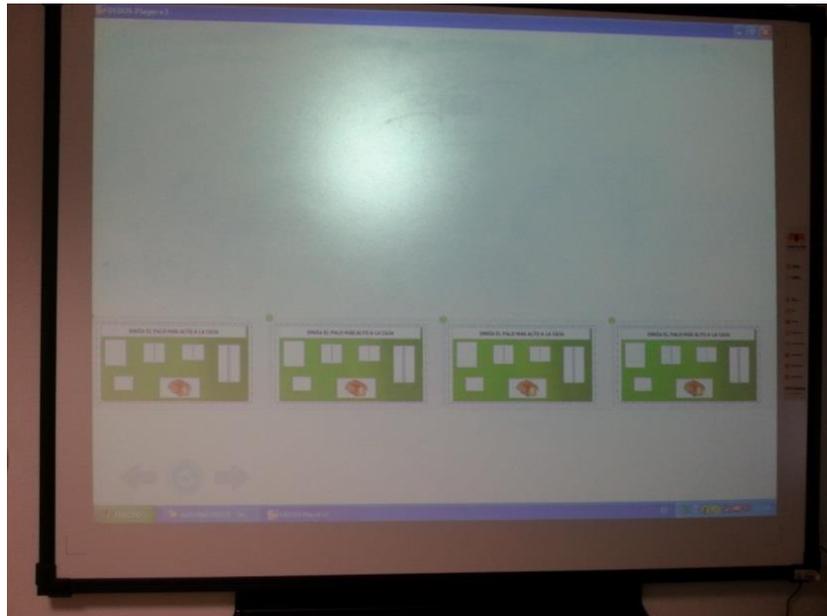


Ilustración 36 - Actividad para cuatro jugadores en la versión 0.27 de DEDOS-Player para PDIs

Tanto para tres como para cuatro jugadores puede observarse como las tarjetas están prácticamente pegadas a los extremos de la pizarra, por lo que es el grado máximo de tamaño que puede obtenerse antes de que las tarjetas se salgan del límite.

Dado que para tres y cuatro jugadores la interacción puede ser incómoda, se tuvieron en cuenta otras posibilidades, como otra disposición de las tarjetas, con el objetivo de hacer más cómoda la interacción (ver Ilustración 37 e Ilustración 38).

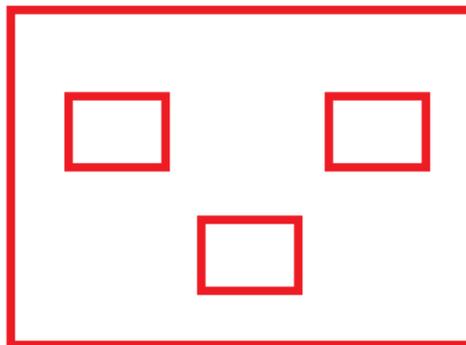


Ilustración 37 - Prototipo alternativo para tres jugadores

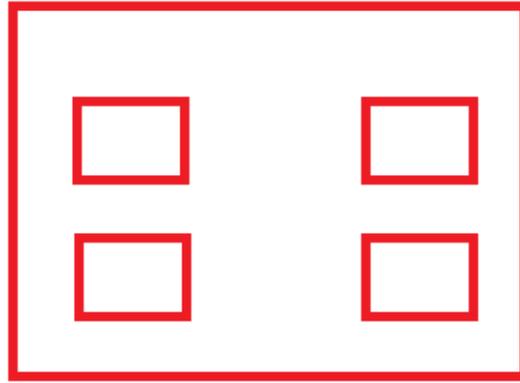


Ilustración 38 - Prototipo alternativo para cuatro jugadores

El prototipo alternativo para cuatro jugadores se descartó rápidamente, ya que una formación en columnas molestaría mucho tanto al jugador de arriba como al de abajo, ya que se disputan el mismo espacio físico en el eje X de la pizarra. Se optó por probar un prototipo más acercado al de tres jugadores, pero con dos jugadores en la parte baja en vez de uno. El resultado puede verse en la Ilustración 39.

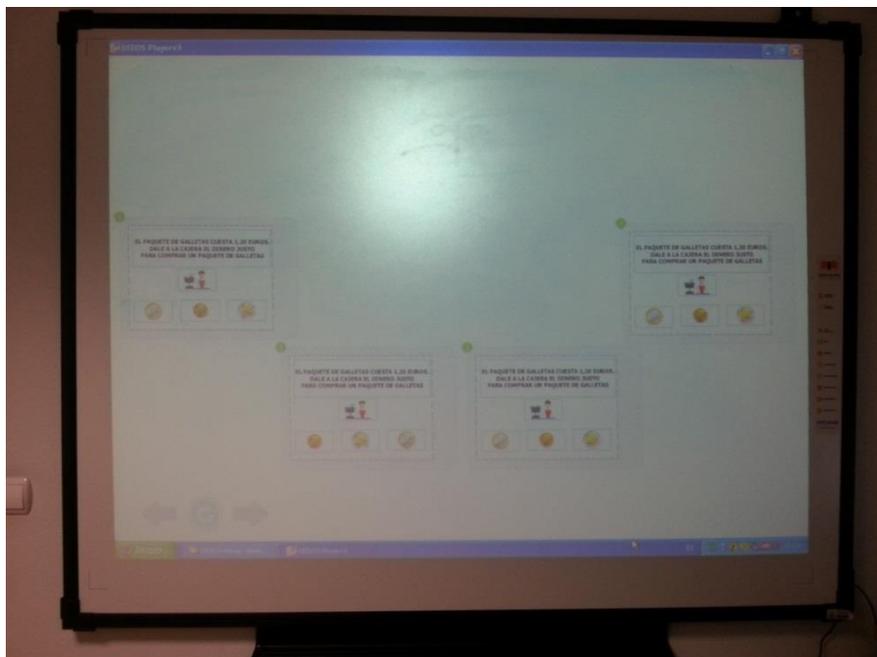


Ilustración 39 - Actividad para cuatro jugadores aplicando el prototipo alternativo

Con este diseño, las tarjetas pueden hacerse un poco más grandes, pero implica que las tarjetas de arriba estén por encima del límite superior de las tarjetas dispuestas en la parte inferior de la pizarra. Dado que la herramienta DEDOS está pensada para ser utilizada mayoritariamente por niños, la altura es un obstáculo para llevar a cabo este prototipo. Se probó si un adulto puesto de rodillas podría llegar a manipular las tarjetas de arriba, lo cual resultó estar muy al límite, haciendo la interacción muy incómoda. Por tanto, tras discutirlo con los tutores de este proyecto, este prototipo alternativo también se descartó, tanto para tres como para cuatro jugadores, ya que la dificultad para ambos era la misma.

Así, descartando estos prototipos alternativos, la versión 0.27 resultó ser la que mejor aprovechaba el espacio de la pizarra adaptándose a las necesidades del usuario medio, resolviendo finalmente el problema del escalado, y resultando la versión final de la nueva herramienta.

5. CONCLUSIONES

5.1 LOGROS ALCANZADOS

Este trabajo fin de grado ha dado la posibilidad de actualizar una herramienta ya existente a una nueva versión capaz de operar en un dispositivo distinto. Tal y como se ha expuesto en el bloque anterior, la última versión desarrollada de la herramienta, la 0.27, cumple con los requisitos iniciales de este proyecto, ya que es capaz de ejecutarse en pizarras digitales, y a su vez ofrece las mismas posibilidades educativas que en el dispositivo inicial, la mesa multicontacto.

El proceso de diseño del prototipo no fue la tarea más costosa, ya que solo había que rotar las zonas de cada jugador de la herramienta DEDOS-Player, de tal manera que la interacción con la PDI fuese lo más cómoda posible, distribuyendo estas zonas de forma lógica y horizontal a lo largo del eje X de la pizarra. El nuevo diseño del menú para incluir la opción de seleccionar si la herramienta se estaba utilizando en una PDI o en una mesa multicontacto tampoco fue un problema, ya que se siguieron las líneas y el diseño que ya tenía el menú anteriormente, sin provocar grandes cambios que desorientasen al usuario al utilizar esta nueva versión.

A la hora de implementar los cambios, ha habido la relativa dificultad de enfrentarse a un lenguaje de programación nuevo, así como el desafío de actualizar una herramienta ya existente, siendo necesario estructurar las clases y sus dependencias entre sí en diagramas UML para facilitar el entendimiento de cómo había sido programada la herramienta. Una vez superado este paso, fue sencillo implementar los cambios necesarios, ya que se habían localizado los métodos dentro de las clases en los que había que cambiar código sin alterar el buen funcionamiento de la herramienta DEDOS-Player.

A medida que se fue desarrollando y evaluando la aplicación, el mayor problema que ha surgido ha sido el del escalado de las distintas zonas de jugadores. El tamaño de estas zonas resultaba ser más pequeño del esperado en un principio, haciendo que la interacción no fuese fácil, con textos e imágenes muy poco legibles.

Una vez solucionado este problema, en una nueva evaluación de la herramienta pudo comprobarse que, para cuatro jugadores la interacción con la PDI no sería la ideal, ya que no hay espacio físico suficiente en el eje X de la pizarra como para que cuatro personas puedan manejar de una manera cómoda e independiente su zona de jugador, sin invadir el espacio del jugador de al lado. Además, el formato de aspecto de las PDIs en las que se evaluó la herramienta (4:3) tampoco ayudaba a una buena interacción para cuatro jugadores, ya que la distancia en el eje X de una PDI con relación de aspecto 16:9 es mayor que la de una con relación de aspecto 4:3. De esta manera, y con la fórmula de escalado que mejor aprovechaba el espacio en el eje X para las tarjetas de los jugadores, la legibilidad de textos e imágenes para cuatro jugadores en mi opinión no es óptima. Debido a esto, se barajaron otros prototipos que cambiaban la disposición de las tarjetas en la pantalla, siendo todos ellos descartados debido a que presentaban de una u otra manera más incomodidades a la hora de interactuar con el dispositivo que el prototipo inicial.

Se ha conseguido ofrecer una herramienta que puede tener una utilidad real en el aula, gracias a la incursión de las TIC en la docencia. La PDI es un dispositivo que puede servir para la realización de trabajos colaborativos en clase, así como de apoyo a las explicaciones del profesor, mostrando de forma gráfica actividades y recursos de una manera personalizada dependiendo de las necesidades del alumnado.

Desde el punto de vista personal, gracias a este proyecto se ha logrado conocer de primera mano cómo es el proceso de desarrollar la adaptación de una herramienta ya existente a través de nuevas tecnologías, en este caso de un dispositivo como es una mesa multicontacto a otro como una pizarra digital. Ha servido para adquirir conocimientos en un nuevo lenguaje de programación, así como para conocer el funcionamiento e implementación de la herramienta DEDOS.

Ha sido muy satisfactorio el hecho de superar obstáculos que, a pesar de haber estado en situaciones similares a lo largo del itinerario de formación del Grado en Ingeniería Informática, en este caso tienen un verdadero impacto en el mundo real.

5.2 TRABAJOS FUTUROS

Considero que este trabajo fin de grado aun ofrece posibilidades para el futuro. Sería muy conveniente evaluar la herramienta en PDI's con una mayor resolución y relación de aspecto. De esta manera, la interacción para cuatro jugadores sería mejor, ya que ofrecería más espacio entre jugador y jugador para interactuar, además de poder aumentar el tamaño de textos e imágenes, para una mejor visualización a más distancia de la PDI. Podría incluso plantearse, dependiendo del espacio disponible, la posibilidad de aumentar el número de jugadores que pueden interactuar con la pizarra, aumentando a su vez las posibilidades de la herramienta DEDOS en términos de nuevas actividades.

Soy consciente de que uno de los mayores problemas actuales de la pizarra digital es su alto coste, y de que no todos los centros escolares pueden permitirse este tipo de tecnología, pero confío en que en el futuro la buena acogida de estos dispositivos lo lleve a un mejor mercado, y de esta manera baje su precio, haciéndolas accesibles para centros de aprendizaje de todo tipo.

BIBLIOGRAFÍA

- Becther, C. L. (2009). *The Interactive Whiteboard Revolution*. ACER Press.
- European Union. (2013). *Survey of schools: ICT in education. Benchmarking access, use and attitudes to technology in Europe's schools*. Recuperado el 31 de mayo de 2014, de <https://goo.gl/56vb8l>
- Hernández Martín, A. O. (2012). *Metodologías de aprendizaje colaborativo a través de las tecnologías*. Ediciones Universidad Salamanca.
- Kopp, K. L. (2012). *Using Interactive Whiteboards in the Classroom*. Shell Education.
- Matthews, J. (2009). *Interactive Whiteboards*. Cherry Lake.
- Mehta, N. (1982). *A flexible machine interface*, M.A.Sc. Thesis. Department of Electrical Engineering, University of Toronto.
- New Media Consortium, EDUCAUSE Learning Initiative. (2014). *The NMC Horizon Report > 2014 Higher Education Edition*. Recuperado el 31 de mayo de 2014, de <http://goo.gl/3KW5Ip>
- Passey, D. (2006). *Technology enhancing learning: analysing uses of information and communication technologies by primary and secondary school pupils with learning frameworks*. Recuperado el 23 de octubre de 2014, de <http://goo.gl/oZCe1h>
- Pitts, K., Hurst, N., & David Sarnoff Res. Center, P. N. (1989). *How much do people prefer widescreen (16x9) to standard NTSC (4x3)?* Recuperado el 10 de diciembre de 2015, de <http://goo.gl/e3mLBL>
- Roschelle, J. M., Pea, R. D., Hoadley, C. M., & Gordin, D. N. (2000). *Changing how and what children learn in school with computer-based technologies*. Recuperado el 23 de octubre de 2014, de <http://goo.gl/Laie81>
- Schubin, M. (1995). *The History of the Perfect Aspect Ratio*. Recuperado el 10 de diciembre de 2015, de <http://goo.gl/TgY2Mj>
- Thomas, M. C. (2010). *Interactive Whiteboards for Education: Theory, Research and Practice*. Information Science Reference.
- Tondeur, J., & Braak, J. &. (2007). *Towards a typology of computer use in primary education*. Journal of Computer Assisted Learning.

Tondeur, J., Hermans, R., & Valcke, M. &. (2008). *Exploring the link between teachers' educational belief profiles and different types of computer use in the classroom.*

Recuperado el 21 de octubre de 2014, de <http://goo.gl/8d6RgU>

Younie, S. L. (2013). *Teaching with technologies: The essential guide.* Open University Press.

ANEXO I: INSTALACIÓN DE ADOBE FLASH CS5.5

Para instalar en un ordenador el programa Adobe Flash CS5.5, es necesario ejecutar el instalador de dicho programa. Aparecerá la ventana principal de instalación, en la que habrá que aceptar el acuerdo de licencia del software de Adobe (ver Ilustración 40).

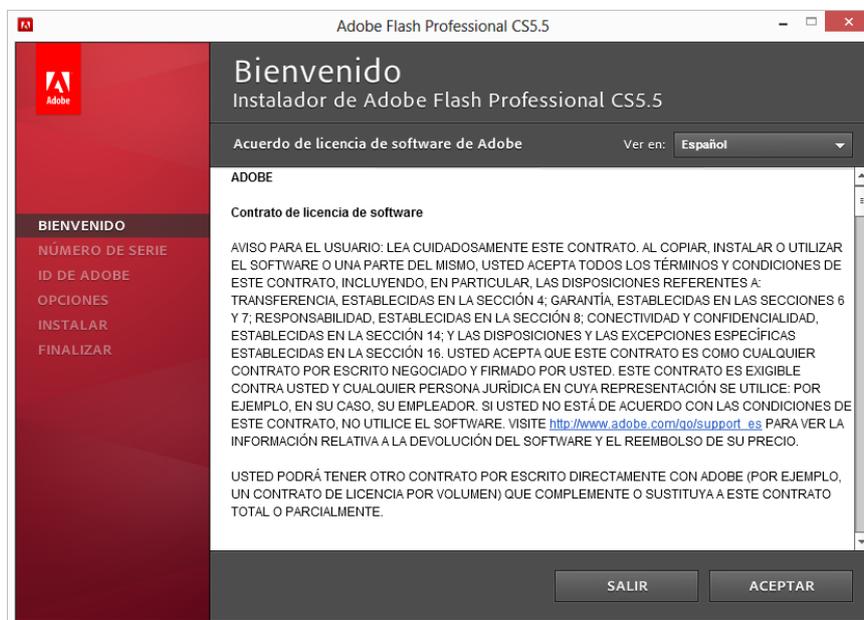


Ilustración 40 - Pantalla inicial del instalador (Adobe Flash CS5.5)

Después de aceptar el acuerdo de licencia, será necesario introducir el número de serie del producto, elegir el idioma en el que será instalado, así como seleccionar los componentes de la instalación y la ruta del equipo en la que se instalarán (ver Ilustración 41).

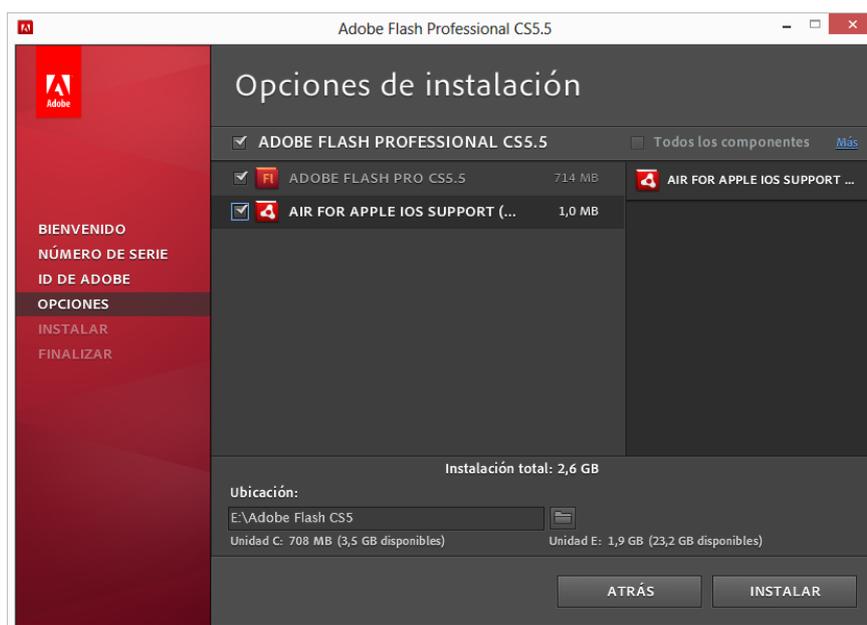


Ilustración 41 - Opciones de instalación (Adobe Flash CS5.5)

Una vez seleccionadas estas opciones, se instalará Adobe Flash CS5.5 en el ordenador, mostrando una barra de progreso (ver Ilustración 42). Una vez haya terminado, habrá terminado la instalación.

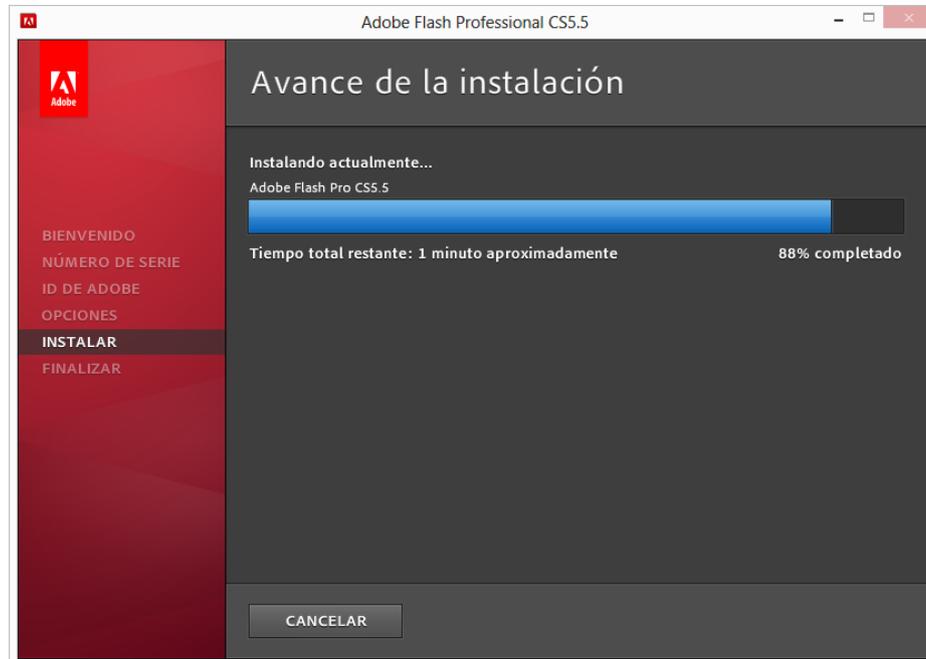


Ilustración 42 - Proceso de instalación (Adobe Flash CS5.5)

ANEXO II: CREACIÓN E INSTALACIÓN DEL FICHERO .AIR

Para poder instalar la aplicación en varios equipos es necesario generar un archivo .air. Este archivo será el instalador de la aplicación, y no será necesario contar con Adobe Flash CS5.5 en los ordenadores en los que vaya a instalarse esta aplicación. Sin embargo, sí es necesario disponer de Adobe Flash CS5.5 en el ordenador en el que va a generarse el archivo .air, ya que es el programa desde el cual va a crearse el instalador.

El archivo principal de una aplicación basada en Adobe Flash es el archivo con extensión .fla, el cual contiene el código necesario para compilar y comprimir después el fichero con extensión .swf. Este archivo .swf puede ejecutarse en el ordenador, pero no es el instalador de la aplicación.

Partiendo de la pantalla inicial de Adobe Flash CS5.5 (ver Ilustración 43), y una vez se tenga terminado y abierto el archivo .fla, haciendo clic en el menú *File* seleccionaremos la opción *Publish Settings*. Se abrirá una pantalla en la que podrán elegirse los ajustes para la creación del archivo .air (ver Ilustración 44).

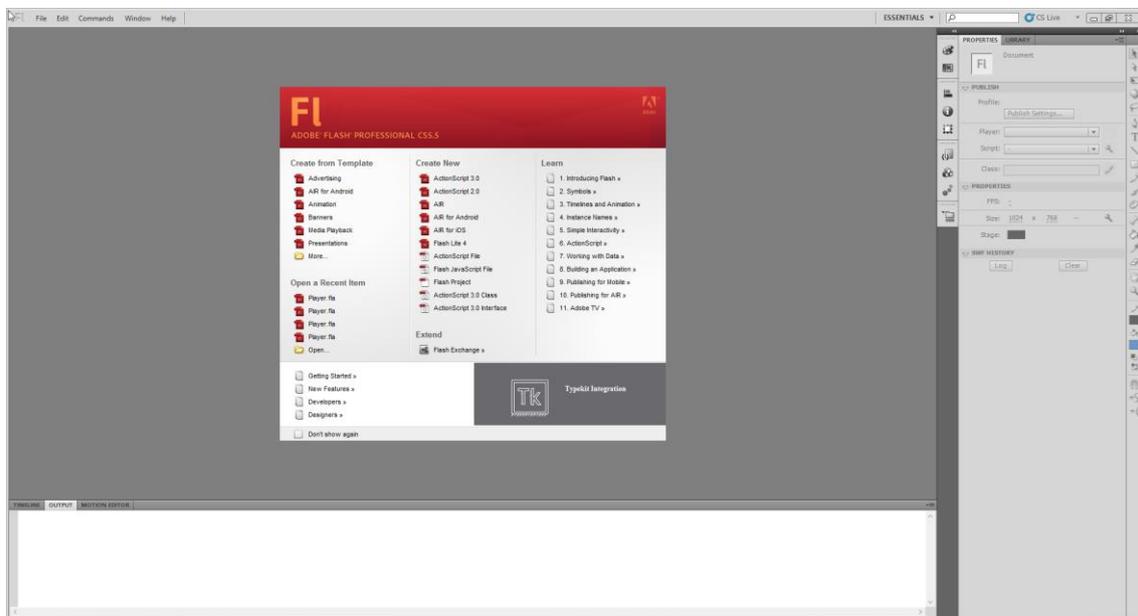


Ilustración 43 - Pantalla inicial de Adobe Flash CS5.5

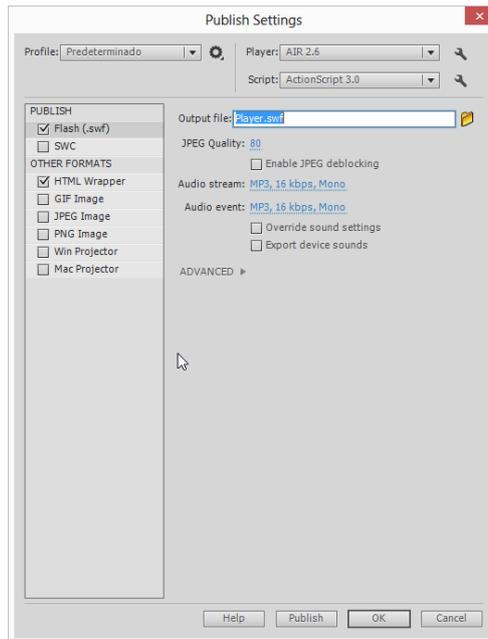


Ilustración 44 - Ventana de "Publish Settings" (Adobe Flash CS5.5)

Es importante seleccionar la opción *AIR 2.6* en el desplegable *Player*, y *ActionScript 3.0* en el desplegable *Script*. Una vez seleccionadas, hay que cerrar la pantalla pulsando en *OK*.

Ahora, en la pantalla principal de Adobe Flash CS5.5 (ver Ilustración 43), haciendo clic en *File* y después en *Air 2.6 Settings...*, aparecerá una pantalla en la cual se podrán elegir los ajustes para crear el archivo .air (ver Ilustración 45).

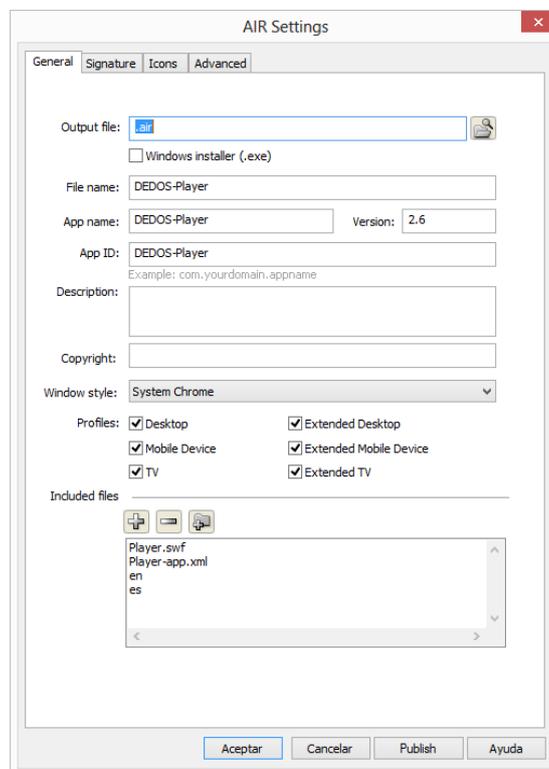


Ilustración 45 - Ventana de "AIR 2.6 Settings" (Adobe Flash CS5.5)

En esta pantalla hay que elegir el nombre de la aplicación, así como el nombre que tendrá el archivo .air. También es posible añadir una imagen para el icono del archivo en la pestaña *Icons*. Es indispensable elegir un certificado para proporcionar los datos del creador de la aplicación. Esto se hace en la pestaña *Signature* (ver Ilustración 46).

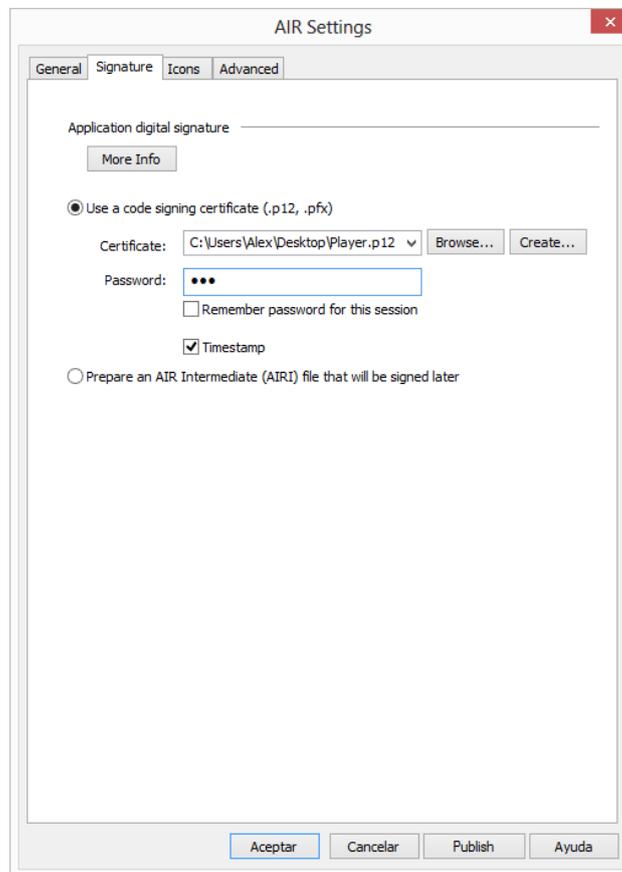


Ilustración 46 - Elección de certificado (Adobe Flash CS5.5)

Haciendo clic en *Create...* se creará un nuevo certificado. Aparecerá una ventana en la que podrá rellenarse el nombre de la persona que publica la aplicación, la organización, etc. Lo más importante en esta ventana será elegir una contraseña para el certificado. Una vez rellenados los datos, hay que hacer clic en *OK*, habiéndose creado el certificado. Una vez de vuelta a la pantalla de *Signature*, se deberá seleccionar el certificado anteriormente creado y escribir la contraseña. Una vez escrita, hay que hacer clic en *Publish*, y se creará el fichero .air en la ruta que se haya especificado.

Para instalar este archivo .air, es necesario que en los equipos en los que se vaya a instalar tengan instalada la aplicación Adobe AIR. Se puede descargar gratuitamente desde <https://get.adobe.com/es/air/>.

Una vez se haya instalado Adobe AIR en el equipo, puede ejecutarse el archivo .air, que instalará la aplicación generada en Adobe Flash CS5.5.

Al instarla, aparecerá una ventana que pide una confirmación (ver Ilustración 47). Dependiendo de los campos que se hayan rellenado al generar el certificado, podrán identificarse en la ventana algunos aspectos relacionados con la seguridad, como la identidad del editor.

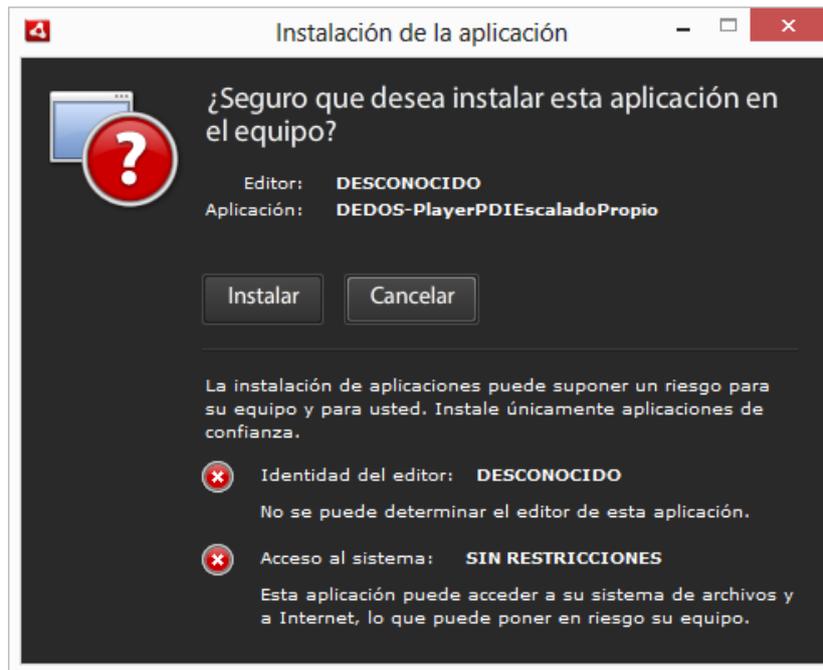


Ilustración 47 - Instalación del fichero .air

Al hacer clic en *Instalar*, aparecerá otra ventana que pedirá la ruta en la que debe instalarse la aplicación. Una vez elegido, pulsando en *Continuar* se instalará la aplicación en el ordenador.