

ED-IT: una herramienta de autor para la creación de actividades educativas basadas en interacción tangible

Verónica Artola^{1,3}, Patricia Pesado¹, Cecilia Sanz^{1,2}

¹Instituto de Investigación en Informática III-LIDI (Centro Asociado CIC) - Facultad de Informática, Universidad Nacional de La Plata

²Investigador Asociado de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

³Becaria CONICET

Resumen

La creación de aplicaciones basadas en interacción tangible (IT), en particular sobre *tabletops*, es un área que se encuentra en evolución. En este artículo se presenta un aporte en esta temática, ya que se describe el desarrollo de una herramienta de autor para la creación de actividades educativas basadas en interacción tangible (IT) sobre *tabletops*. El desarrollo se fundamenta en una investigación previa de diferentes herramientas que posibilitan la creación de aplicaciones IT, sus principales características y el análisis de sus posibilidades para el escenario educativo, considerando puntos fuertes y los puntos de vacancias. Se comienza por la presentación de los resultados de este análisis, para luego describir el desarrollo de Ed-IT (**ED**itor para aplicaciones educativas **IT**) que es la herramienta de autor diseñada por los autores y foco de este trabajo. Se describe la propuesta para llevar adelante las primeras evaluaciones de ED-IT y las líneas de trabajo futuro.

Palabras clave: interacción tangible, *tabletop*, herramientas de autor, actividades educativas

1. Introducción

Desde sus inicios, la generación de aplicaciones basadas en IT se ha visto fuertemente ligada a especialistas

informáticos. Sin embargo, la creación de este tipo de aplicaciones, requiere de la colaboración de diferentes profesionales con conocimientos expertos en dominios específicos, además de las habilidades de ingeniería e informática propias del desarrollo de sistemas (Gerken, Frechenhäuser, Döner, y Luderschmidt, 2013). Cuando los expertos en alguna disciplina desean implementar una aplicación basada en IT específica para mediar contenidos o tareas de su dominio suelen interactuar con profesionales del ámbito de la Informática. En muchos casos, la poca intervención de los expertos en el dominio dentro del proceso de desarrollo conlleva a frecuentes errores de diseño. Además, para cada cambio en la aplicación, el experto en el dominio debe volver a recurrir al experto técnico (van Herk y cols., 2009). En relación a este problema Tetteroo, Soute y Markopoulos (2013) mencionan que el potencial de las tecnologías IT se puede aprovechar aún más cuando los expertos en el dominio, como maestros y terapeutas, participan en el proceso de desarrollo de sistemas tangibles. Este contexto presenta un conjunto de nuevos retos para los desarrolladores de sistemas IT, orientados a dar nuevas oportunidades de participación a los expertos en el dominio y ofrecer la posibilidad de adecuar las aplicaciones para que se adapten a cada caso. Los desafíos para desarrollar un sistema de

IT, se resumen conceptualmente, en "definir el espacio de diseño" (Garzotto, 2011).

Estos desafíos se enumeran en la siguiente lista: (Tetteroo y cols., 2013):

- Diseñar qué puentes de integración entre el mundo físico y virtual son adecuados en cada caso
- Crear/seleccionar objetos físicos para las interacciones
- Guiar el diseño de las interacciones
- Anticiparse a los usuarios sin formación técnica
- Considerar ámbitos socio-técnicos en el desarrollo: el aprendizaje, la motivación, la colaboración y la organización.

En este contexto es que tienen sentido las herramientas de autor, como mediadoras para la creación de sistemas de IT. Con ellas los expertos en el dominio pueden involucrarse en varias de estas tareas de diseño y desarrollo.

Las herramientas de autor (HA) se definen como programas que permiten al usuario crear sus propias aplicaciones informáticas sin necesidad de recurrir a un especialista informático. Este tipo de aplicaciones ha cobrado especial interés en el ámbito de la Educación al enriquecer las propuestas de enseñanza de los docentes. Habitualmente, este tipo de herramientas trabajan a través de plantillas predeterminadas (Camarda y Minzi, 2012) y, luego de un proceso de compilado de estas plantillas, generan un programa que se puede ejecutar de manera independiente del software que lo generó (Villa y cols., 2002).

El enfoque de las HA trae consigo una transformación del rol del usuario, que asume responsabilidades y tareas tradicionalmente destinadas a los desarrolladores. Las herramientas de

autor resuelven múltiples aspectos de la creación de aplicaciones, orientadas al experto en el dominio, que van desde la especificación de valores de parámetros a la decisión de incluir determinados contenidos y comportamientos (Tetteroo y cols., 2013). En particular, el usuario de una HA para crear aplicaciones IT enfrentará desafíos entre los que se pueden mencionar:

- Inventar metáforas que den forma física a la información digital y determinar qué información se propone representar digital y físicamente. (Shaer y cols., 2004; B. A. Ullmer y Ishii, 2002)
- Seleccionar comportamientos y acciones. El comportamiento de un objeto físico está determinado por su naturaleza y por el contexto de uso. Por lo tanto, para cada objeto de interacción, se requiere seleccionar y definir qué acciones son significativas y en qué contexto. Además, se debe implementar una solución para detectar estas acciones de manera computacional y comunicar esta información a los usuarios (Shaer y cols., 2004; Benford y cols., 2005).
- Determinar los dispositivos de E/S. Entre las tecnologías de entrada de datos para rastrear objetos, y gestos de usuarios en el mundo físico, se encuentran: RFID¹, técnicas de visión por computadora, microcontroladores y sensores. Con respecto a la salida, además de utilizarse pantallas y parlantes existe una variedad de actuadores para crear la salida física (Leds, vibradores, etc.). Como cada una de estas tecnologías requiere un conjunto diferente de dispositivos físicos e instrucciones, la integración y

¹ RFID (*Radio Frequency Identification*): permite identificar un objeto, en el que hay un

chip, de manera remota mediante una señal de radio.

personalización es difícil y costosa (Shaer y cols., 2004)

- Diseñar cada interacción. Múltiples usuarios pueden interactuar en paralelo con múltiples objetos físicos, o realizar una sola acción a través de múltiples objetos físicos. Es importante en estos casos considerar temas como los puntos de acceso (Hornecker y Buur, 2006), y la coordinación espacial y temporal. (Shaer y cols., 2004)

El desarrollo de una herramienta que aborde estos desafíos presenta un compromiso entre facilidad de uso y la flexibilidad de la herramienta. El balance entre estos dos aspectos se encuentra estrechamente vinculado al tipo de usuario para el cual se crea la herramienta de autor. Además, la herramienta debe proponer capas de abstracción de los aspectos complejos y técnicos para facilitar la tarea del usuario no experto en Informática. Varias de las tareas antes mencionadas serán transparentes para el usuario de la herramienta de autor, gracias a estas capas de abstracción.

2. Herramientas para crear aplicaciones IT en el ámbito educativo

En el ámbito educativo, la variedad de alumnos con diferentes perfiles promueve el interés de los educadores en diseñar diferentes tipos de actividades educativas. En particular, esto se vuelve más frecuente en el contexto de educación especial, donde los requisitos son heterogéneos y cambiantes, debido a las características de cada alumno y la naturaleza de sus necesidades educativas. Así, la tecnología para mediar actividades debe ser altamente flexible, evolutiva y fácil de modificar para abordar el nivel de desarrollo de cada estudiante. Los expertos deberán poder crear actividades educativas que se ajusten a cada caso. En particular, en los casos de aplicaciones tangibles, deberían

tener la posibilidad de actuar como diseñadores, pudiendo crear experiencias de aprendizaje adaptadas a las necesidades específicas y a los objetivos educativos que se proponen (Garzotto y Gonella, 2011).

Los educadores o terapeutas, entonces, deberán poder configurar e intervenir en aspectos tales como (Sanz y cols., 2017):

1. Grados de abstracción de la actividad (representaciones gráficas, de texto o pictogramas)
2. Objetos virtuales y físicos a asociar.
3. Imágenes de fondo
4. *Feedback* en varios niveles.
5. Contenidos de presentación o contextualización de las actividades.
6. Estrategias de entrega de instrucciones/consignas (audio y/o visual).
7. Definiciones sobre cómo y cuándo termina una actividad.
8. Secuencia de actividades y su navegación.

3. Antecedentes de herramientas para crear aplicaciones IT

Se presenta una recopilación de herramientas que ayudan a crear aplicaciones basadas en IT para *tabletops*. Se realizó una revisión sistemática de bibliografía que permite visualizar el estado actual de estas herramientas. Se revisaron un total de 492 artículos en la primera etapa, de los cuales se filtraron 74 para su análisis en profundidad. Finalmente, la selección de artículos contó con 13 artículos que cumplían con los criterios de búsqueda pautados inicialmente (descripción completa de la herramienta o editor de contenidos IT sobre *tabletops*). Cabe aclararse, que también varios artículos se descartaron por presentar la misma herramienta. En la Tabla 1, se listan las herramientas analizadas a partir de esta revisión.

Herramienta	Autor
TUIMS	(Shaer y Jacob, 2009)
ESPranto	(van Herk y cols., 2009)
TUIO	(Kaltenbrunner, 2009)
TLF	(Garzotto y Gonella, 2011)
TUI-VR	(Israel y cols., 2011)
TEC	(Hochstenbach-Waelen y cols., 2012)
ToyVision	(Marco, Cerezo, y Baldassarri, 2012)
CLAY	(Gerken y cols., 2013)
KitVision	(Bonillo Fernandez, 2013)
TULIP	(Tobias, Maquil, y Latour, 2015)
DIY-AT	(Moraiti y cols, 2015)
TouchTokens	(Appert y cols, 2018)
Arcadia	(Kelly y cols, 2018)

Tabla 1- Herramientas para la creación de aplicaciones IT

Los trabajos analizados indican que en las experiencias llevadas a cabo con las herramientas, los usuarios han logrado alcanzar los objetivos propuestos por sus desarrolladores. Sin embargo, cada uno presenta características diferentes. Se ha observado que en cada proyecto varían los esfuerzos por agregar distintas capas de abstracción en diferentes niveles. Es así que en un extremo, cerca del hardware, se encuentran aquellas herramientas que abstraen la tecnología subyacente pero requieren que un programador resuelva la lógica de la aplicación y en el extremo opuesto, cerca del usuario, se encuentran aquellas que resuelven tanto la detección de objetos físicos como las acciones con ellos, y le dan al usuario la posibilidad de crear actividades sin programar. Entre estos extremos del gradiente, varían las capas de abstracción y las posibilidades que ofrece cada herramienta, siendo proporcional el nivel de abstracción a las limitaciones/facilidades que presenta dicha herramienta. Esto es, cuanto mayor es el nivel de abstracción, más cuestiones de bajo nivel se encuentran resueltas, lo que termina resultando en una menor

flexibilidad para adaptar ciertos aspectos. Para poder organizar las herramientas, se consideran 2 categorías según las capas de abstracción (Figura 1), que se explican debajo:

1. Herramientas con capa de abstracción de la tecnología subyacente a la IT: este tipo de herramientas oculta el hardware y los algoritmos de reconocimiento de marcadores u objetos físicos. Si bien cuentan con facilidades para crear aplicaciones aún es necesario contar con conocimientos de programación para poder llegar a crear una aplicación basada en IT. Por lo tanto, estas herramientas no son consideradas herramientas de autor en este trabajo.
2. Herramientas con capa de abstracción para facilitar el diseño del sistema IT: en este tipo de herramientas no es necesario programar. Solo se asocian eventos con comportamientos de la aplicación. Se continúa teniendo abstracción de la tecnología subyacente.

En el límite de aquellas que facilitan el diseño, se ubican aquellas herramientas que ofrecen facilidades y capas de abstracción, pero aún requieren de algunos conocimientos básicos de programación.

De las herramientas analizadas en la revisión sistemática puede notarse que muchas de ellas no cumplen con ser herramientas de autor por el hecho de no permitir crear actividades, sino que solo permiten configurar aspectos de los objetos físicos con los que se interactúa en la aplicación. Este es el caso de

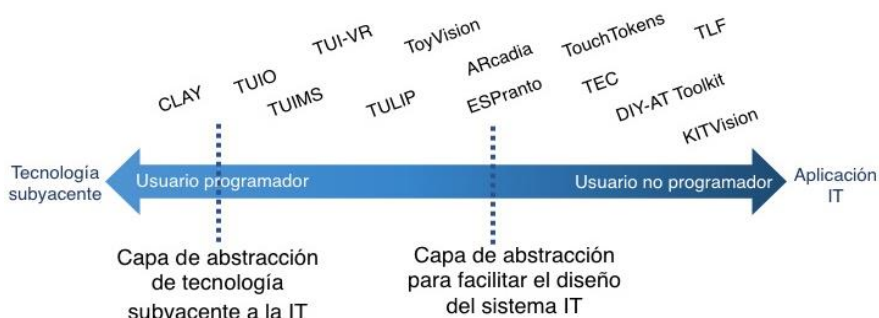


Figura 1 - Herramientas organizadas por nivel de abstracción

TouchTokens y *DIY-AT*. Además, quedan fuera de este grupo las herramientas que están pensadas para ayudar a desarrolladores técnicos a crear aplicaciones IT y que involucran conocimientos avanzados de programación. Estas son: *TUIO*, *TUIMS*, *TUI-VR*, *CLAY* y *TULIP*. En el caso de *ToyVision*, la herramienta presenta una capa de abstracción para la inclusión de objetos activos en una aplicación IT a través de plantillas, pero no incluye el diseño de la actividad en su conjunto, por lo que también queda fuera de este análisis.

Luego de este primer análisis, se pone el foco en las siguientes herramientas a considerar según el objetivo de esta investigación: *ESPranto*, *TLF*, *TEC*, *KitVision* y *Arcadia*.

Con el fin de guiar el análisis de las diferentes herramientas, se determinan características de interés a analizar en relación a las herramientas de autor.

1. Actividades: posibilidades que ofrece la herramienta para crear actividades educativas. Por ejemplo, ¿presenta plantillas?, ¿permite crear paquetes con varias actividades?, ¿se habilita la navegación entre actividades?
2. Personalización: refiere a las posibilidades de modificación de

aspectos funcionales y gráficos, con el objetivo de adaptarse más eficientemente a las necesidades de cada docente en particular.

3. Estándares: se analiza la posibilidad de exportar el contenido respetando estándares para el desarrollo de contenidos. Los estándares proporcionan un lenguaje común para que los contenidos pueden comunicarse con distintos entornos tecnológicos (por ejemplo, diferentes tipos de aplicaciones y repositorios en la web).
4. Se encuentra disponible para su uso. ¿Qué licencias tiene?

Además se agrega como característica a evaluar, la tecnología involucrada en las actividades IT. En este caso se considera: técnicas de visión por computadora, RFID, sensores, controladores, etc.

Por otro lado, para el análisis de las herramientas se retoman los criterios presentados en (Sanz y cols., 2017) y descriptos en la sección 2. La información de cada herramienta con respecto a estos aspectos se resume en la Tabla 2 y se describe detalladamente a continuación.

En general, todas las herramientas permiten configurar aspectos que tienen que ver con el diseño y con la

	Actividades		Personalización		Estándares	Disponible	Tecnología			Aspectos configurables						
	Plantilla	Progr. Simple	Gráfico-sonido	Funcionalidad			V. por. comp.	RFID	Adaptable	Tipo de objetos	Relación entre objetos	Imágenes de fondo	Feedback	Contextualización	Instrucciones	Finalizar actividad
ESPranto		X	X	X				X	X	X	X	X	X	X	X	X
TLF	X		X	X				X		X	X	X	X	X		
TEC	X		X	X				X		X		X				X
KitVision	X		X	X		X	X			X	X	X	X	X	X	
Arcadia		X	X	X			X			X	X		X			

Tabla 2 - Herramientas de autor para aplicaciones IT y sus características

funcionalidad como por ejemplo, permiten seleccionar algunos gráficos y sonidos para personalizar el sistema IT (es decir, son herramientas en la categoría: con capa de abstracción para facilitar el diseño del sistema IT).

El caso de ESPranto es una herramienta que se destaca ya que cumple con todas las características de configuración y abstracción presentadas previamente en este trabajo. Dispone de diferentes niveles de diseño orientados a distintos tipos de usuarios. En el nivel más cercano a un usuario no experto en Informática (padres y docentes) se utiliza un editor gráfico con bloques como en Scratch (Resnik y cols, 2009), con el fin de incluir contenido. También, se cuenta con un nivel avanzado de programación en base a macros orientado a expertos en el dominio (por ejemplo, un diseñador de juegos para rehabilitación). Finalmente, para usuarios programadores expertos se les ofrece el *kernel* de programación de ESPranto. Este nivel permite cambiar la tecnología subyacente. Así, la herramienta aún en el nivel pensado para usuarios menos técnicos, presenta la necesidad de algún conocimiento básico en programación. ARcadia es una herramienta que también utiliza un editor gráfico con bloques tipo Scratch, por lo que requiere que el usuario cuente con algunos conocimientos de programación básicos. ARcadia no considera el uso de *tabletops*, sino que aumenta con información digital objetos físicos. Se la consideró para el análisis ya que fue diseñada para la prototipación rápida de aplicaciones IT y esto podría facilitar la participación de expertos en el dominio. TEC es una herramienta basada en el trabajo con una superficie que detecta objetos con RFID, llamada *TagTiles*. Está orientada a la creación de actividades por parte de terapeutas en contextos de rehabilitación. Permite editar, a través de plantillas, sobre qué áreas de la *TagTiles* se realizarán las interacciones y se puede asociar a cada

área un *feedback* auditivo. Se pueden seleccionar los objetos físicos con los que se trabajará. El tipo de actividades que se pueden diseñar se basan en apoyar, deslizar, sacar y reubicar objetos cotidianos en/de la superficie de la *TagTiles*. Estas acciones son solicitadas en el marco de los ejercicios diseñados por terapeutas. Si bien no incorpora estándares para entornos tecnológicos, plantea la importancia de promover prácticas relacionadas con la creación y el intercambio de *software*, que son conocidos en la comunidad de *software open-source*, pero aún alejados en la comunidad de la salud y la educación. En el caso TLF, se trata de una herramienta web que posibilita el desarrollo de actividades IT. También permite el diseño a través de plantillas que se corresponden a distintos tipos de actividades. Permite configurar todos los aspectos deseados de cada actividad pero no permite crear secuencias de actividades. Finalmente, KitVision es una herramienta diseñada para que terapeutas desarrollen actividades IT, también a través de plantillas. Permite configurar todos los aspectos de una actividad, pero no una secuencia de ellas. En su lugar, muestra en la mesa todas las actividades creadas para que el usuario pueda darle inicio a cualquiera de ellas. Con respecto a su disponibilidad, es la única de las herramientas evaluadas que se encuentra disponible para su descarga con licencia GNU General Public License version 3.0 (GPLv3).

4. ED-IT. EDitor para aplicaciones educativas IT

A partir del análisis realizado, se han observado algunas áreas de vacancias en relación a las posibilidades que podría ofrecer una herramienta de autor orientada a la creación de actividades educativas basadas en IT. Así, se gestó el diseño de una herramienta de autor llamada ED-IT (**ED**itor para aplicaciones educativas **IT**). Para el

diseño de la herramienta se tomaron como modelo herramientas de autor, que han tenido éxito en la creación de contenidos para el ámbito educativo (Ardora y ExeLearning), para aplicar el mismo modelo de trabajo a través de plantillas. En cuanto a los antecedentes de herramientas para la creación de sistemas IT, se consideraron los casos de TLF, TEC y KitVision, que también se basan en plantillas. Actualmente ED-IT, tiene una primera versión funcional que posibilita crear un proyecto con una secuencia de actividades basadas en IT. Al momento se cuenta con 2 tipos de plantillas para la creación de actividades:

- 1) Plantilla de creación de presentación: permite agregar contenido (texto, imagen, sonido, video) para presentar un tema o una actividad (Ver figura 2)

Figura 2- Plantilla de presentación

- 2) Plantilla de actividad de asociación simple: esta plantilla facilita diseñar actividades donde se asocian objetos físicos a información digital disponible en la superficie de la *tabletop*. Para diseñar estas actividades la plantilla propone dos pasos. En el paso 1, se debe completar un formulario para ingresar el título de la actividad, su consigna, el fondo y otros aspectos generales (por ejemplo, sonido de presentación, etc.). En el paso 2, se deben crear las asociaciones de la actividad. Cada asociación es una interacción que involucra un objeto y un área de la mesa. El usuario debe indicar qué

objeto va en qué área de la mesa, el *feedback* que la aplicación dará al colocar el objeto especificado en esa área y el que dará al colocar cualquier otro objeto (error). El *feedback* puede involucrar imagen y/o sonido. La interfaz para crear las interacciones se diseñó de manera tal que el usuario tenga una representación gráfica de la superficie de la mesa y sobre ella seleccione áreas (con operaciones de clic y arrastre) y asigne imágenes (Figura 3)

Figura 3- Plantilla para agregar interacciones

Cada actividad creada se integra a un proyecto que el usuario puede visualizar en el lateral de la herramienta. En este espacio el docente puede reorganizar la secuencia en que se ejecutarán las actividades. Además, el docente puede configurar las siguientes opciones de un proyecto: sus metadatos, título, disciplina, entre otros.

La publicación de un proyecto para su utilización sobre una *tabletop*, se reduce a un clic. El docente obtendrá de esta manera un archivo zip que contendrá los marcadores cargados, la aplicación IT creada y un pdf con los fiduciales que se deberán colocar debajo de cada objeto.

Además, si el docente lo desea también puede guardar el proyecto en formato .la para poder importarlo luego y continuar su edición, o para compartirlo con la

comunidad (este último aspecto aún está pendiente de implementar).

5. Evaluación propuesta para ED-IT

Al momento se han realizado pruebas informales con usuarios del contexto de los autores. Las pruebas han ido motivando mejoras, principalmente en aspectos de usabilidad. Se ha diseñado una evaluación formal a través de estudios de caso, en el que participarán docentes de diferentes niveles educativos. Dentro de la muestra, se analizan características de perfil que involucran: edad, género, nivel educativo en el que ejercen la docencia, conocimientos previos de IT y de tecnología digital en general.

La evaluación se conforma por 2 partes: inicialmente los docentes interactúan con una aplicación IT y luego, en la segunda parte, crean dicha aplicación utilizando la herramienta. Los instrumentos con los que se recabará información son: la observación participante con una grilla de aspectos a observar relacionados a la usabilidad, el instrumento basado en TAM (*Technology Acceptance Model*) (Davis, Bagozzi, y Warshaw, 1989) adaptado por Teo (2009) para medir la intención de uso de la tecnología, y finalmente un *focus group* para información adicional en relación a la percepción de los docente sobre la IT y la herramienta.

6. Conclusiones y trabajos futuros

Se ha realizado un estudio del estado del arte de las herramientas de autor para creación de aplicaciones IT. Se han encontrado trabajos que permiten al usuario abstraer distintas capas del desarrollo de estas aplicaciones. En los trabajos encontrados solo una herramienta se encuentra disponible para su uso (KitVision). Con las vacancias encontradas se diseñó una nueva herramienta de autor, ED-IT, que busca cubrir alguna de ellas. Como trabajo futuro se realizarán evaluaciones sobre la

usabilidad de la herramienta y la intención de uso de los docentes.

7. Bibliografía

- Appert, C., Pietriga, E., Bartenlian, E., y González, R. M. (2018). Custom-made tangible interfaces with touchtokens. En *Proceedings of the 2018 international conference on advanced visual interfaces* (pp. 15:1–15:9). New York, NY, USA: ACM. doi: 10.1145/3206505.3206509
- Benford, S., Schnädelbach, H., Koleva, B., Anastasi, R., Greenhalgh, C., Rodden, T., y cols (2005). Expected, sensed, and desired: A framework for designing sensingbased interaction. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1), 3–30
- Bonillo Fernandez, C. (2013). Desarrollo de una herramienta para el diseño y ejecución de actividades enfocadas a ancianos con el tabletop nikvision (Final Degree Project). Departamento de Informática e Ingeniería de Sistemas (DIIS) Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza.
- Camarda, P., y Minzi, V. (2012). *Primaria digital, aulas digitales móviles, manual general introductorio*. Bs. As. Ministerio de Educación de la Nación.
- Davis, F. D., Bagozzi, R. P., y Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. *Management science*, 35(8), 982–1003.
- Garzotto, F., y Gonella, R. (2011). An open-ended tangible environment for disabled children's learning. En *Proceedings of the 10th international conference on interaction design and children* (pp. 52–61). New York, NY, USA: ACM. doi:10.1145/1999030.1999037

- Gerken, K., Frechenhäuser, S., Dörner, R., y Luderschmidt, J. (2013). Authoring support for post-wimp applications. En *Ifip conference on human-computer interaction* (pp. 744–761).
- Hochstenbach-Waelen, A., Timmermans, A., Seelen, H., Tetteroo, D., y Markopoulos, P. (2012). Tag-exercise creator: Towards end-user development for tangible interaction in rehabilitation training. En *Proceedings of the 4th acm sigchi symposium on engineering interactive computing systems* (pp. 293–298). New York, NY, USA: ACM. doi: 10.1145/2305484.2305534
- Hornecker, E., y Buur, J. (2006). Getting a grip on tangible interaction: A framework on physical space and social interaction. En *Proceedings of the sigchi conference on human factors in computing systems* (pp. 437–446). New York, NY, USA: ACM. doi: 10.1145/1124772.1124838
- Israel, J. H., Belaifa, O., Gispén, A., y Stark, R. (2011). An object-centric interaction framework for tangible interfaces in virtual environments. En *Proceedings of the fifth international conference on tangible, embedded, and embodied interaction* (pp. 325–332).
- Kaltenbrunner, M. (2009). reactivation and tuio: A tangible tabletop toolkit. En *Proceedings of the acm international conference on interactive tabletops and surfaces* (pp. 9–16). New York, NY, USA: ACM. doi: 10.1145/1731903.1731906
- Kaltenbrunner, M., y Echtler, F. (2018, junio). The tuio 2.0 protocol: An abstraction framework for tangible interactive surfaces. *Proc. ACM Hum.-Comput. Interact.*, 2(EICS), 8:1–8:35. doi: 10.1145/3229090
- Kelly, A., Shapiro, R. B., deHalleux, J., y Ball, T. (2018). Arcadia: A rapid prototyping platform for real-time tangible interfaces. En *Extended abstracts of the 2018 chi conference on human factors in computing systems* (pp. D314:1–D314:4). New York, NY, USA: ACM. doi: 10.1145/3170427.3186535
- Marco, J., Baldassarri, S., y Cerezo, E. (2013). Toyvision: a toolkit to support the creation of innovative board-games with tangible interaction. En *Proceedings of the 7th international conference on tangible, embedded and embodied interaction* (pp. 291–298).
- Marco, J., Cerezo, E., y Baldassarri, S. (2012). Modeling tangible game controls with toyvision. En *Proceedings of the 13th international conference on interaction person-computer* (pp. 19:1–19:2). New York, NY, USA: ACM. doi: 10.1145/2379636.2379655
- Marco, J., Oakley, I., Cerezo, E., y Baldassarri, S. (2013). Designing and making a tangible tabletop game with toyvision. En *Proceedings of the 7th international conference on tangible, embedded and embodied interaction* (pp. 423–426)
- Moraiti, A., Vanden Abeele, V., Vanroye, E., y Geurts, L. (2015). Empowering occupational therapists with a diy-toolkit for smart soft objects. En *Proceedings of the ninth international conference on tangible, embedded, and embodied interaction* (pp. 387–394). New York, NY, USA: ACM. doi: 10.1145/2677199.2680598
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. B. (2009). Scratch: Programming for all. *Commun. AcM*, 52(11), 60-67.
- Sanz, C. (2014). Material de estudio del curso "tecnología informática. Evolución y aplicaciones" de la carrera de especialización en tecnología informática aplicada en educación.

- Sanz, C., Gorga, G., Artola, V., Moralejo, L., Salazar Mesía, N., Archuby, F. H., Pesado, P. M. (2017). Estilos y paradigmas de interacción persona-ordenador y sus posibilidades para el escenario educativo. En XIX workshop de investigadores en ciencias de la computación (wicc 2017, itba, Bs. As).
- Shaer, O., y Jacob, R. J. (2009). A specification paradigm for the design and implementation of tangible user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 16(4), 20
- Shaer, O., Leland, N., Calvillo-Gamez, E. H., y Jacob, R. J. (2004). The tac paradigm: specifying tangible user interfaces. *Personal and Ubiquitous Computing*, 8(5), 359–369.
- Teo, T. (2009). Modelling technology acceptance in education: A study of pre-service teachers. *Computers & Education*, 52(2), 302–312.
- Tetteroo, D., Soute, I., y Markopoulos, P. (2013). Five key challenges in end-user development for tangible and embodied interaction. En *Proceedings of the 15th acm on international conference on multimodal interaction* (pp. 247–254). New York, NY, USA: ACM. Descargado de doi: 10.1145/2522848.2522887
- Tobias, E., Maquil, V., y Latour, T. (2015). Tulip: A widget-based software framework for tangible tabletop interfaces. En *Proceedings of the 7th acm sigchi symposium on engineering interactive computing systems* (pp. 216–221). New York, NY, USA: ACM. doi: 10.1145/2774225.2775080
- Ullmer, B. A., y Ishii, H. (2002). *Tangible interfaces for manipulating aggregates of digital information* (Tesis Doctoral no publicada). Citeseer
- Van Herk, R., Verhaegh, J., y Fontijn, W. F. (2009). Espranto sdk: An adaptive programming environment for tangible applications. En *Proceedings of the sigchi conference on human factors in computing systems* (pp. 849–858). NY. USA: ACM. doi: 10.1145/1518701.1518831
- Villa, M. G., Morales, A. M. F., Martínez, J., Valenzuela, P. P. M., Saorín, S. M., Marín, A. R. C., y Del Baño, J. V. (2002). Herramientas de autor e integración curricular: “las aventuras de topy”, una aplicación multimedia para el desarrollo de la comunicación alternativa y aumentativa en el aula. *Las nuevas tecnologías en la respuesta educativa a la diversidad*.