

Diseño de un juego basado en Interacción Tangible para la enseñanza de Programación

Artola Verónica^{1,2}, Sanz Cecilia¹, Gorga Gladys¹, Pesado Patricia^{1,3}

{vartola, csanz, ggorga, ppesado}@lidi.info.unlp.edu.ar

1 Instituto de Investigación en Informática – LIDI. Facultad de Informática. Universidad Nacional de La Plata

2 Becaria de CONICET (Beca tipo 1)

3 Comisión de investigación científica de la Pcia. De Bs. As. (CIC)

Resumen. En este artículo se presenta un juego educativo basado en interacción tangible, diseñado para ayudar a los alumnos en la comprensión de temas claves de los cursos introductorios de Programación, de las carreras en Informática. El juego (EPIT: Enseñanza de la Programación usando Interacción Tangible) propone la reflexión y el análisis sobre posibles soluciones para la resolución de un problema que involucra el uso de diferentes estructuras de datos, abordadas en los programas de estas asignaturas. Se trata de un juego colaborativo, en los que se trabaja en grupos de 4 alumnos, a los que se les propone atravesar por tres etapas que hacen a la dinámica de la actividad, y en cada una se procura que los alumnos pongan en juego habilidades cognitivas tales como el análisis, comparación, y comprensión, y en particular, en la segunda y tercera etapa, habilidades más orientadas a la colaboración y reflexión conjunta. Se presentan aquí los antecedentes que permitieron la realización de este juego, su diseño y el plan para poder aplicarlo en los próximos meses. Se detallan los resultados esperados y las conclusiones del trabajo.

Keywords: Interacción Tangible, Enseñanza de Programación, Trabajo Colaborativo

1. Introducción

La historia de la interfaz evoluciona diversificándose y especializándose con el tiempo. En la actualidad, existe un abanico de tipos de interfaces muy amplio, y muchas de ellas modifican los modelos de interacción. Así varios autores hablan de nuevos modelos de interacción que conforman nuevos paradigmas para el área de Interacción Persona Ordenador (Bowman, 1997) (Välkkynen, 2003) (Canny, 2006) (Heim, 2007) (Oramas, 2010). La interacción basada en soportes móviles, la basada en interfaces cooperativas y colaborativas, las de soporte *multitouch*, la realidad aumentada, la interacción tangible, etc. Cada una de éstas con sus restricciones y ventajas, se adapta a distintos entornos y escenarios (Ishii, 2006).

El uso de la interacción tangible en escenarios educativos ha ido cobrando importancia, y ha sido foco de estudio a través de diferentes investigaciones (O'Malley, 2004) (Price, 2008) (Marshall, 2007) (Manches, 2009) (Zufferey, 2009) (Guisen, 2011) (Sanz, 2012). La interacción tangible propone un fuerte acoplamiento entre la información digital y su representación tangible. A través de la manipulación física de las representaciones tangibles, la representación digital se altera. Las formas físicas sirven al mismo tiempo como representación y control de sus contrapartes digitales (ver Fig. 1).

En las interfaces tangibles el modelo de interacción es llamado *MCRpd: Model-Control-Representation (physical and digital)*. En este modelo los componentes son los mismos que en el *MVC*¹ pero se presenta al elemento vista, llamado representación, dividido en dos subcomponentes: representación física (*physical*

¹ MVC (Model-View Control). Arquitectura de sistemas de información. Se basa en tres dimensiones principales: *Modelo* correspondiente a la *información*, *Vista* correspondiente a la *presentación* o interacción con el usuario y *Control* correspondiente al *comportamiento*. (Weitzenfeld, 2004).

representations) y representación digital (*digital representations*). *MCRpd* muestra la integración de la representación física y el control (Ullmer, 2001).

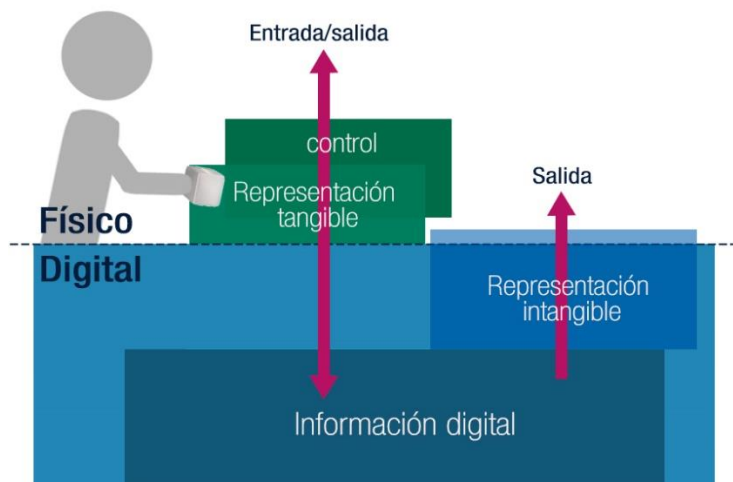


Fig. 1. – Modelo de interacción de las interfaces tangibles (Adaptado de Ishii, 2006).

Koleva et al. (Koleva, 2003) han realizado una distinción entre diferentes tipos de interfaces tangibles en términos de "grado de coherencia" - es decir, proponen analizar si las representaciones físicas y las digitales son vistas como un objeto común que existe en los mundos físico y digital, o son vistas como objetos independientes, aunque interrelacionados temporalmente. El nivel más bajo de la coherencia en su esquema se ve en herramientas de uso general, donde puede haber un objeto físico que se utiliza para manipular cualquier número de objetos digitales. Un ejemplo es el mouse, que controla varias funciones diferentes (por ejemplo, menús, barras de desplazamiento, ventanas, casillas de verificación) en diferentes momentos. En contraste, se pueden tener objetos físicos como dispositivos de entrada donde se dedica cada objeto a llevar a cabo una y solamente una función. El mayor nivel de coherencia (físico-digital de acoplamiento) en el esquema presentado en (Koleva, 2003) es donde existe la ilusión de que las representaciones físicas y digitales son el mismo objeto (O'Malley, 2004).

Entre las propuestas de la interacción tangible, aquellas basadas en superficies activas ya son una realidad habitual, bien en forma de pizarras interactivas o *tabletops* horizontales (Marco et al, 2010).

Las *tabletops* son superficies horizontales aumentadas computacionalmente, también llamadas mesas interactivas. En general, para abordar la interacción tangible sobre una *tabletop* se utiliza una serie de marcadores o fiduciales que, a través de su reconocimiento, provocan determinados eventos y acciones dentro de la aplicación. Un fiducial es una imagen que, situada en la base de un objeto físico permite identificarlo mediante un sistema de detección visual. Este marcador puede dar información como identidad, posición y orientación.

El juego que se presenta en este trabajo se basa en interacción tangible sobre una *tabletop* y utiliza marcadores como forma de detección de los objetos físicos con los que se trabaja.

2. Interacción tangible en la enseñanza de Programación

A lo largo de los años los investigadores han desarrollado diferentes sistemas orientados a ofrecer entornos más sencillos para enseñar a programar, muchos de ellos basados en interacción tangible. Estos sistemas permiten a los usuarios escribir programas mediante el uso de los objetos físicos sin utilizar el teclado y el

mouse. Antecedentes en este sentido se ven representados por la propuesta de Logo de Papert² (Chakraborty, 1999) y los dispositivos de entrada “*Button Box*” y “*Slot Machine*” de Perlman³ (McNerney, 2004). A continuación se describen otros proyectos vinculados a la enseñanza de la programación utilizando interacción tangible.

AlgoBlocks (Suzuki y Kato, 1995) es un proyecto que adopta el uso de bloques como medio de interacción. Cada bloque tiene una semántica especial y los usuarios pueden escribir un programa mediante la conexión de los bloques para dirigir un submarino a través de un laberinto.

Electronic Block (Wyeth y Wyeth, 2001) (Wyeth y Purchase, 2002) utiliza bloques de construcción para programar. Se compone de tres tipos de bloques de construcción: bloques de sensor como entrada, bloques de lógica para llevar a cabo la computación lógica, y bloques de comportamiento como salida. Está diseñado para niños en edad preescolar, por lo que la sintaxis es simple y fácil de manipular.

Tern y *Quetzal* son lenguajes de programación tangible que utilizan bloques que representan semántica específica. Los usuarios conectan los bloques para generar programas. Una vez creado el programa es necesario utilizar de forma manual una cámara para capturar la imagen de la secuencia de los bloques. La imagen se transfiere a una computadora para identificar el programa desarrollado y controlar con él funciones virtuales o un robot real (Horn y Jacob, 2007) (Horn et al, 2008) (Horn et al, 2009)

Wang et al. (2011) han creado *T-Maze*, posteriormente *TanPro-Kit* (Wang, 2013) y *E-blocks* (Wang, 2013). Todos ellos son sistemas de programación tangible que utilizan bloques que al conectarlos generan en tiempo real movimientos dentro de un laberinto. Trabajan con distintos tipos de bloques, bloque de inicio, bloque de fin, bloque de dirección y bloque sensor. Al conectar los bloques se muestra tanto en la pantalla del ordenador como en los propios bloques tangibles, *feedback* indicando si la acción llevada a cabo es correcta.

Dialando (Smith, 2010) es otro sistema de programación tangible que utiliza controladores Arduino, conectados via *bluetooth* a una *notebook*. El diseño de la interfaz de *Dialando* se basa en la rotación de un disco. El disco representa la dirección que tomará un avatar en pantalla.

Toque (Tarkan, 2010) permite crear programas utilizando como dispositivos de entrada los controles *Wiimote*⁴ y *Nunchuck*⁵. Los programas se basan en recetas con animaciones de un chef en pantalla que se controla de forma dinámica. Incorpora conceptos de programación, tales como eventos, objetos, primitivas, funciones, argumentos y bucles.

Todos estos sistemas buscan proporcionar a los estudiantes un camino para avanzar, en capas, hacia entornos de programación cada vez más auténticos.

² Seymour Papert: (Pretoria, Sudáfrica, 1928) pionero de la inteligencia artificial, creador del lenguaje de programación Logo en 1968. Considerado como destacado científico computacional, matemático y educador. Trabajó con el psicólogo educativo Jean Piaget en la Universidad de Ginebra desde 1959 hasta 1963. <http://www.papert.org/>

³ Radia Perlman : (Portsmouth, Virginia, 1951) es una creadora de software e ingeniera de redes, experta en seguridad, conocida como la Madre de internet. Actualmente trabaja para Intel, Estados Unidos, para la que ha conseguido más de 47 patentes.

⁴ Wiimote (Wii Remote) es el mando principal de la consola Wii de Nintendo. Es capaz de detectar movimiento en el espacio y de apuntar hacia objetos en la pantalla.

⁵ Nunchuk es una expansión para el mando inalámbrico de Wii. Su nombre proviene del Nunchaku, arma de artes marciales, ya que al conectarlo al Wiimote da una apariencia similar a éste.

3. Diseño de un juego basado en IT para la enseñanza de programación

El juego en el que se focaliza este trabajo ha sido planificado a partir de la experiencia de los autores en el área de interacción tangible y educación. En (Artola, 2013) se presentó el desarrollo de ITCol (Interacción Tangible para la Colaboración), un juego de detectives basado en IT, enfocado en vivenciar el trabajo colaborativo, en el marco de un curso de postgrado de la Facultad de Informática de la UNLP. El trabajo fue evaluado mediante un proceso que involucró validar la dinámica del juego, la aplicación informática y el modelo de interacción, y la utilidad de la aplicación en función del objetivo educativo propuesto. Se realizaron numerosas sesiones, se analizaron resultados y se observó que:

- El proceso de colaboración se llevaba a cabo y se veía beneficiado por la interacción tangible con el uso de la *tabletop*, que no dispersaba ni distraía la atención de los participantes, ya que abordaban la tarea de una manera natural, sin que la tecnología les resultara una barrera o elemento de dispersión sino un elemento de integración y de acompañamiento en la tarea.
- El trabajo con objetos reales y la aplicación informática resultaba un aspecto de motivación entre los alumnos.
- La disposición de los participantes alrededor de una mesa invitaba a la colaboración y al diálogo entre los participantes.
- Las habilidades cognitivas puestas en juego a través de la dinámica propuesta, se relacionaron con las necesarias para abordar un proceso colaborativo (comunicación, análisis conjunto, negociación, etc.).

En función de los resultados obtenidos, se consideró la utilización de la plantilla desarrollada para ITCOL como base para el desarrollo de otros juegos educativos orientados a la enseñanza de diferentes temas. En particular, el juego desarrollado aquí reutiliza la plantilla de ITCOL y la readapta para cumplir con los objetivos educativos que se propone.

3.1. Motivación para la creación EPIT

En la cátedra de Algoritmos, Datos y Programas y Programación I, correspondientes a asignaturas del primer año de carreras en Informática de la Facultad de Informática de la UNLP, se observa una dificultad en la comprensión y aplicación de las estructuras de datos más básicas (tales como arreglos y listas simples), que se abordan en estos cursos. En particular, la implementación de programas que hacen uso de operaciones sobre estas estructuras, presentan para el alumno cierta complejidad, ya que son los primeros pasos que está dando en el desarrollo de soluciones algorítmicas. En estos cursos el alumno aborda la resolución de problemas con computadora. Primero se pone el foco en que pueda dominar el uso de estructuras de control, y posteriormente, en que pueda lograr la representación de objetos del mundo real a través de diferentes tipos de datos. Por último, se busca que el alumno pueda construir soluciones integrales en un lenguaje de programación, en donde ponga en juego la selección de las estructuras de control y de las estructuras de datos, pertinentes para representar los diferentes datos involucrados en el problema, y que es capaz de operar sobre estos.

EPIT busca ejercitar al alumno en el análisis de soluciones en forma colaborativa, propone el ejercicio de debatir y negociar cuál de entre varias soluciones es la correcta y por qué. Pone en juego la comprensión de los alumnos de los temas abordados en la asignatura, y habilidades de análisis y colaboración para que el grupo de alumnos pueda arribar a la solución del problema planteado. En la siguiente sección se presenta en detalle la funcionalidad de EPIT.

3.2. Funcionalidad de EPIT

EPIT es una aplicación informática, con la dinámica de juego, en la que pueden intervenir un grupo de alumnos y que funciona sobre una *tabletop*, la cual constituye el espacio de interacción.

Dada la *tabletop* de la que se dispone para el juego, se ha pensado el trabajo para grupos de entre 3 y 4 alumnos, acorde al tamaño de la mesa y a la superficie de interacción disponible.

El juego transita por tres etapas, cada una de las cuáles tiene un propósito específico. A continuación se describe cada una de ellas y su vinculación con el objetivo educativo.

Etapas 1: durante esta etapa los participantes interactúan en la mesa de a uno por vez. Cada integrante del grupo recibe un objeto que representa un módulo (procedimiento o función) que cumple con una tarea específica. Al colocarlo sobre la mesa, EPIT muestra una descripción del objetivo del módulo y una serie de implementaciones posibles para este. El alumno deberá escoger cuál de los códigos es el correcto para el objetivo planteado.

Cuando todos los jugadores hayan seleccionado el código que consideran correcto para el módulo que les tocó, estarán en condiciones de avanzar a la siguiente etapa.

La etapa tiene un tiempo máximo. Si al alcanzar dicho tiempo, el equipo aún no ha terminado, podrá solicitar más tiempo, pero esto degradará la *performance* del equipo.

Etapas 2: el equipo se reúne alrededor de la mesa. La aplicación presenta un problema a resolver.

El objetivo de esta etapa es escoger la solución al problema general de un conjunto de soluciones posibles. Cabe aclararse que sólo una es correcta, el resto tiene errores de diferentes tipos. Además, el equipo debe tomar decisiones respecto a los módulos elegidos individualmente, durante la primera etapa, de manera tal de acordar y negociar si las elecciones han sido adecuadas o los módulos presentan algún tipo de error. Con esto se busca socializar el aprendizaje y que los alumnos tomen decisiones en forma conjunta, como equipo, de manera tal que se ponga en juego la comprensión de los temas involucrados por parte de cada uno de los jugadores, el análisis, y la comunicación y el debate.

La interacción en esta etapa es a través de 3 tipos de objetos físicos:

- **Objetos Solución:**

Estos objetos representan programas principales. Al colocarlos en la mesa se mostrará el código de un programa. El equipo deberá elegir entre todos los objetos solución, cuál es el que representa la solución correcta (Ver Fig. 2).

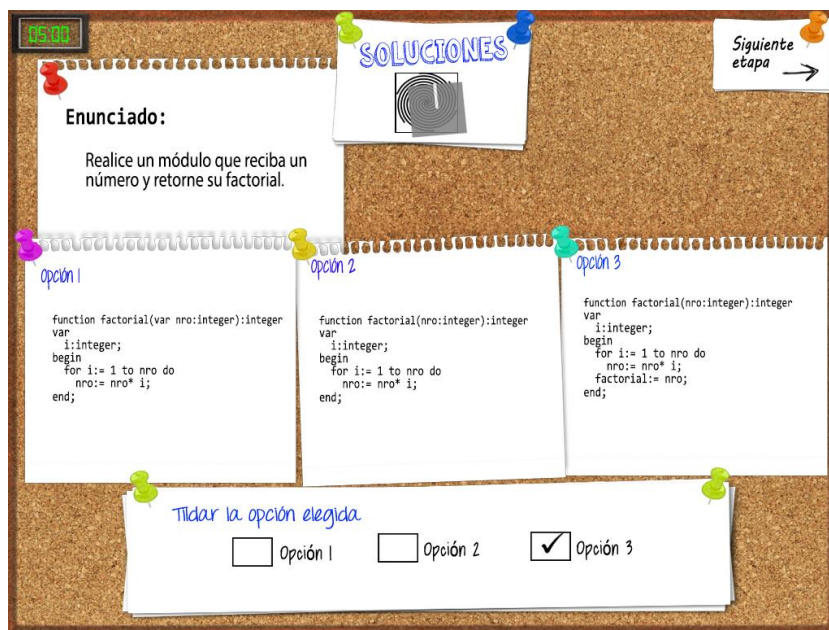


Fig. 2. Ejemplo de la pantalla al colocar un objeto que representa un módulo sobre la mesa.

- **Objetos Error:**

Este tipo de objeto representa categorías de errores que puede tener un código (por ejemplo, errores en la elección de las estructuras de control, errores en el pasaje de parámetros, errores de tipo lógico, etc.). Estos objetos cobran sentido al utilizarlos en conjunto con los objetos solución. Si los alumnos deciden apoyar sobre la mesa un objeto solución y un objeto error, EPIT muestra información/sugerencias acerca de los errores correspondientes a la categoría elegida y la solución en cuestión (representada por el objeto que se apoyó en la mesa). En este caso podría ser que la solución y el objeto error no guarden relación y se les informará esto.

- **Objetos Módulo :**

Son los objetos utilizados en la primera etapa. Esta vez al colocarlos sobre la mesa sólo se muestra el código seleccionado, por el alumno, como el correcto. Sirven a manera de recordatorio para que el equipo pueda conocer lo que cada alumno seleccionó individualmente.

Durante esta etapa los alumnos debe tomar decisiones respecto de: la elección de la implementación de cada módulo de la etapa 1, y la elección de la implementación de la solución general, que utiliza dichos módulos. Para ello deberán debatir la tarea que cada uno ha realizado individualmente, contrastar y analizar las diferentes soluciones del problema general. Si el equipo sospecha de algún tipo de error en las soluciones puede tomar la decisión de relacionar el código con una categoría de error, utilizando los objetos antes mencionados (apoyándolos sobre la mesa). Sin embargo, tiene una cantidad estipulada de relaciones permitidas y al mismo tiempo cada interacción de este tipo degrada la *performance* del equipo.

Al igual que en la etapa 1, se dispone de un tiempo para completar la tarea, que es decidir cuál es la implementación correcta para el problema planteado, y si los módulos que utiliza (elegidos por cada alumno de forma individual) son o no correctos.

Etapa 3:

En esta etapa los alumnos como equipo definen las respuestas al problema planteado en la etapa 2. Deben apoyar el objeto solución sobre la mesa (aquel que consideran que resuelve el problema planteado de forma correcta), y aquellos módulos que consideran que tienen algún tipo de error (Ver Fig. 3)



Fig. 3. La imagen muestra la pantalla correspondiente a la etapa 3. Al igual que en toda la aplicación las áreas de interacción están marcadas con un cuadrado con espiral.

3.3. Caso específico para el juego

Sobre la base de la dinámica de juego antes explicitada, se desarrolló un caso específico. Cabe aclararse que es posible cambiar el caso para abordar diferentes problemas sobre los que realizar el juego. En particular, se decidió enfocar un problema en el que los alumnos tuvieran que utilizar listas simples y vectores, y aplicar operaciones de creación de una lista (agregando siempre el nuevo nodo al final), eliminar

de la lista ocurrencias de determinados valores, insertar elementos en un vector, y eliminar ocurrencias de valores de éste. El problema se conformó en relación a un torneo local de fútbol en el que se busca premiar a los mejores jugadores. Para ello se cuenta con información de los jugadores y de los goles que cada uno convirtió en cada partido.

3.4. Aspectos técnicos de la implementación del juego

La versión prototipo actual de la aplicación fue desarrollada en el entorno *Adobe Flash Builder 4.5* y *ActionScript 3.0 (AS3)*, con entorno de ejecución *Adobe AIR*. La gráfica fue realizada en *GIMP*.

Para la detección de los fiduciales se utilizó la biblioteca *ReactIVision* (Kaltenbrunner y Bencina, 2007). *ReactIVision* se comunica a través del puerto *UDP* utilizando el protocolo *TUIO* (Kaltenbrunner et al., 2005).

Para la configuración del juego se trabajó con un archivo *XML* cuyo esquema ha sido diseñado con la intención de poder generar nuevas actividades.

Se trabaja con una *tabletop* a la que se ha denominado: *VisionAR*. La estructura de esta *tabletop* se realizó en fibrofácil y acrílico. Su diseño permite un simple armado/desarmado para facilitar su traslado de un lugar a otro. La Fig. 4 muestra las partes que conforman la estructura de *VisionAr*, que se basa en el modelo *NikVision*⁶.

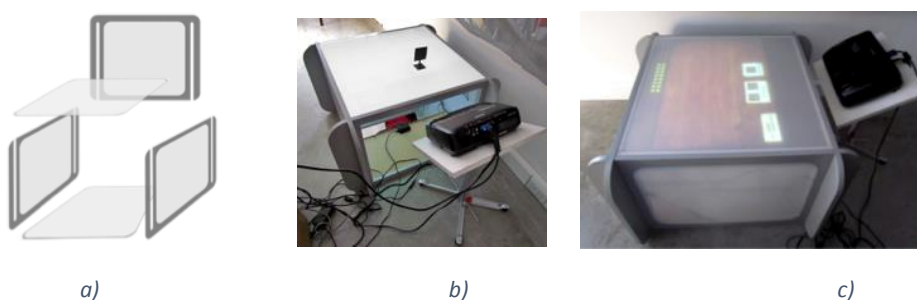


Fig. 4- a) Estructura de VisionAr b) VSIONAR Se observa el proyector, ubicado sobre un atril regulable, y espejo en su interior c) en funcionamiento.

4. Resultados esperados y planificación del caso de estudio

EPIT será utilizado en el marco de un caso de estudio que ayudará a validarlo. Se trabajará en el curso de Programación I antes mencionado, durante el mes de octubre de este año. Se invitará a los alumnos a participar de esta actividad, acordando sesiones de trabajo por cada 4 alumnos. El objetivo será abordar la integración de temas en las semanas de consultas previas al parcial de la materia. Se tiene un estudio de errores frecuentes que comenten los alumnos en la implementación de las operaciones involucradas en el problema planteado en el caso de estudio. Por lo que para cada módulo que se asignará de forma individual a un alumno, se le presentarán la solución correcta, y soluciones distractores con errores vinculados a los vistos en este estudio. Lo mismo se realizará para la solución general de problema, en la que el equipo de alumnos deberá debatir o reflexionar sobre la correcta.

Durante las sesiones se contará con dos observadores que registrarán el proceso a través de un formulario semi-estructurado. Se analizarán aspectos vinculados a:

⁶ NikVision es una tabletop desarrollada por el grupo GIGA AffectiveLab de la Universidad de Zaragoza. NikVision fue diseñada especialmente para ser usada por niños pequeños. La interacción con NIKVision se realiza mediante manipulación de juguetes convencionales sobre la superficie de la mesa.

Dominio de los contenidos: cuáles errores los alumnos son capaces de detectar y cuáles no, formas de razonamientos que proponen, capacidad de análisis que presentan en relación al problema planteado, capacidad para leer el código y entender qué está resolviendo, entre otros.

Proceso colaborativo: cómo se da el proceso en términos generales, y en cuanto a: la comunicación, la capacidad de realizar acuerdos y negociar soluciones, la toma de decisiones en forma conjunta, las estrategias para lograr el objetivo, la organización, etc.

Aplicación y la modalidad de interacción propuesta: se estudiará el funcionamiento de la aplicación en términos de usabilidad, y en cuanto a la interfaz de interacción tangible sobre *tabletop*. Si bien ya se tienen varios resultados en el uso de la interacción tangible sobre *tabletop*, a partir de trabajos previos de los autores (Sanz, 2012), (Baldassarri, 2011), (Artola, 2013), se quiere ampliar la muestra y tomar esta población de jóvenes. Además, se tomarán aspectos concretos de EPIT, por ejemplo, en cuanto a los objetos físicos que se usarán para esta aplicación y su vinculación con los eventos que producen.

5. Conclusiones y trabajos futuros

En este artículo se ha presentado el diseño de un juego llamado EPIT, basado en interacción tangible sobre una *tabletop*. El juego es colaborativo, y se orienta a la enseñanza de Programación, y en particular, se ha planteado un problema relacionado con el uso de estructuras de datos como vectores y listas simples, para llevar adelante un caso de estudio en la materia de Programación I de la Facultad de Informática de la UNLP.

Se cree que este juego aporta el escenario educativo en cuestión por varios aspectos:

- a. Los alumnos del primer año no tienen práctica en el trabajo conjunto en pos de buscar soluciones a problemas del área informática. Dado que recién están iniciando su camino en su formación profesional, se considera importante motivarlos a realizar este tipo de actividades para fortalecer sus capacidades analíticas, entender otros puntos de vista o razonamientos, poder exponer sus ideas frente a otros e identificar soluciones correctas y errores.
- b. El acercamiento a los temas de la materia a través de una dinámica de juego permite el trabajo del contenido, desde otra estrategia diferente a la que tradicionalmente se utiliza en el desarrollo de estos cursos. Se espera que colabore en la comprensión de los aspectos a abordar por EPIT, y resulte un complemento a las tradicionales prácticas que se llevan a cabo.
- c. La posibilidad para el docente de tener una mayor observación del proceso de razonamiento de los alumnos y detectar errores de conceptos o de prácticas, en torno al problema planteado. Por ello, se observarán y registrarán las comunicaciones que lleva adelante cada grupo de alumnos, los comentarios, las dificultades, los aciertos y los errores que se abordan en cada caso.
- d. La motivación que puede provocar al alumno de Informática acercarse al uso de aplicaciones novedosas o que no estén a su alcance como es el caso del uso de la interacción tangible sobre una *tabletop*. Se espera que se fortalezca de esta manera su motivación como futuro profesional y que pueda ver e interesarse por proyectos de investigación que se estén llevando a cabo en la Facultad.
- e. El acercamiento a sus docentes profesores desde otro escenario que no sea el tradicional de clase expositiva dialogada, de manera tal de generar mayor comunicación y posibilidad de exponer sus dificultades sobre los temas de la materia.

6. Referencias bibliográficas

Artola V., Sanz, C. y Giacomantone J. (2013). *Interacción tangible en aplicaciones educativas. Diseño e implementación de un prototipo basado en este paradigma de interacción orientado al aprendizaje colaborativo*. Tesis de Licenciatura en Sistemas, Universidad Nacional de La Plata, Argentina.

- Baldassarri S., Marco J., Sanz C., Guisen M.A., De Giusti A.E., Cerezo E. (2011) Interacción Tangible para Desarrollar Competencias Comunicacionales en Educación Especial. XII Congreso Internacional Interacción Persona-Ordenador, ISBN: 978-84-9281-234-9, pp. 341-344. Lisboa, Portugal.
- Bowman D., Hodges, L. F. (1997) "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments". Proceedings of 1997 pp 35–38. Symp. On Interactive 3D Graphics. Providence, RI, Estados Unidos.
- Canny J. (2006) "The Future of Human Computer Interaction". ACM Queue, Vol. 4 No. 6. pp24-32.
- Chakraborty, A., Graebner, R., & Stocky, T. (1999). LOGO.
- Guisen A., Baldassarri S., Sanz C., Marco J., De Giusti A., Cerezo E. (2011). "Herramienta de apoyo basada en Interacción Tangible para el desarrollo de competencias comunicacionales en usuarios de CAA". VI Congreso Iberoamericano de Tecnologías de Apoyo a la Discapacidad (IBERDISCAP 2011). pp. 301-308. V.I. Palma de Mallorca, España.
- Heim S. (2007) "The Resonant Interface: HCI Foundations for Interaction Design." ISBN-13: 978-0321375964
- Horn, M.S., y Jacob, R.J.K., (2007). Designing Tangible Programming Languages for Classroom Use. Proceedings of TEI'07 First International Conference on Tangible and Embedded Interaction.
- Horn, M. S., Solovey, E. T., & Jacob, R. J. (2008). Tangible programming and informal science learning: making TUIs work for museums. In Proceedings of the 7th international conference on Interaction design and children (pp. 194-201). ACM.
- Horn, M. S., Solovey, E. T., Crouser, R. J., y Jacob, R. J. (2009). Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 975-984). ACM.
- Ishii H. (2006). Tangible User Interfaces. CHI 2006 Workshop.
- Kaltenbrunner M., Bencina R. (2007) "ReactIVision: a computer-vision framework for table-based tangible interaction". En TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction, pages 69-74, New York, NY, USA. ACM.
- Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E. (2005) "TUIO - A Protocol for Table-Top Tangible User Interfaces". En Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, Vannes, France.
- Koleva B., Benford S., Kher Hui N., Rodden T. (2003). "A Framework for Tangible User Interfaces". Physical Interaction (PI03) - Workshop on Real World User Interfaces, a workshop at the Mobile HCI Conference 2003. Udine, Italy.
- Manches A., O'Malley, C., Benford, S. (2009). The role of physical representations in solving number problems: A comparison of young children's use of physical and virtual materials. *Computers & Education* 54, 622-640.
- Marco, J., Baldassarri, S. & Cerezo, E. (2010). Bridging the Gap between Children and Tabletop Designers. In Proceedings of the 9th international Conference on interaction Design and Children. España: IDC '10. p. 98-107.
- Marshall, P. (2007). Do tangible interfaces enhance learning?. TEI'07. Baton Rouge, LA, USA
- McNerney, T. S. (2004). From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal and Ubiquitous Computing*,8(5), 326-337.
- O'Malley C. (2004). "Literature Review in Learning with Tangible Technologies". NESTA Futurelab.
- Oramas Mogrovejo J. A. (2010) "Diseño e implementación de un sistema para la manipulación de objetos virtuales por medio de un lenguaje dactilológico". Accedido a través de: <http://www.dspace.espol.edu.ec/handle/123456789/8689> (Consultado en 2013).
- Price S. (2008). A representation approach to conceptualizing tangible learning environments. TEI'08, Bonn, Alemania.
- Sanz C., Baldassarri S., Guisen A., Marco J., Cerezo E., De Giusti A. (2012) ACoTI: herramienta de interacción tangible para el desarrollo de competencias comunicacionales en usuarios de comunicación alternativa. Primeros resultados de su evaluación. VII Congreso de Tecnología en Educación y Educación en Tecnología. TE&ET. ISBN: 978-987-28186-0-9. Pág.226-233. Junín/Pergamino, Buenos Aires, Argentina
- Smith, A. C. (2010). Dialando: Tangible programming for the novice with Scratch, Processing and Arduino.

- Suzuki H. & Kato H. (1995) **Interaction-level support for collaborative learning: AlgoBlock—an open programming language**. Proceeding CSCL '95 The first international conference on Computer support for collaborative learning. pp 349-355. L. Erlbaum Associates Inc. Hillsdale, NJ, Estados Unidos.
- Tarkan S., Sazawal V., Druin A., Golub E., Bonsignore E.M., Greg W., Atrash Z. 2010. Toque: Designing a Cooking- Based Programming Language For and With Children. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Atlanta, Georgia, USA, April 10 - 15, 2010). CHI '10. 2417-2436
- Ullmer B., Ishii H. (2001). "Emerging frameworks for tangible user interfaces", in Carroll, J. ed. Human Computer Interaction in the New Millenium, Allison. Wesley, 579-601.
- Välkkynen P., Korhonen I. et al. (2003) "A user interaction paradigm for physical browsing and near-object control based on tags". Physical Interaction (PI03) - Workshop on Real World User Interfaces", collocated with Mobile HCI Conference 2003. Udine (Italy).
- Wang, D.L., Zhang, C., Wang, H.A. (2011). T-Maze: a tangible programming tool for children. In Proceedings of the 10th International Conference on Interaction Design and Children (Ann Arbor, USA, June 20-23, 2011). IDC '11. ACM, New York, NY, 127-135.
- Wang, D., Qi, Y., Zhang, Y., & Wang, T. (2013). TanPro-kit: a tangible programming tool for children. In Proceedings of the 12th International Conference on Interaction Design and Children (pp. 344-347). ACM.
- Wang, D., Zhang, Y., & Chen, S. (2013). E-Block: A Tangible Programming Tool with Graphical Blocks. Mathematical Problems in Engineering, 2013.
- Weitzenfeld A., (2004) Ingeniería de software orientada a objetos con UML, Java e internet. Cengage (Thomson) Learning (ISBN 970-686-190-4)
- Wyeth P. y Wyeth G., (2001) "Electronic blocks: tangible programming elements for preschoolers," in Proceedings of the International Conference on Human-Computer Interaction (INTERACT '01), pp. 496–503, 2001.
- Wyeth P. y Purchase H. C., (2002) "Tangible programming elements for young children," in Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI '02), pp. 774–775, ACM Press, 2002.
- Zufferey G., Jermann P. Lucchi A., Dillenbourg, P. (2009) "TinkerSheets: Using Paper Forms to Control and Visualize Tangible Simulations" In Proceedings of the Third International Conference on Tangible and Embedded Interaction (TEI'09), pp. 377-384
- Schweikardt, E. and Gross, M.D. The robot is the program: interacting with roBlocks. In *Proc. Conference on Tangible and Embedded Interaction TEI'08*, (2008).