



Experiencia m-learning en el aprendizaje de un lenguaje de programación

Autor/res/ras: Pedro María Alcover Garau; Francisco Sánchez Ledesma; Juan Ángel Pastor Franco

Institución u Organismo al que pertenecen: Universidad Politécnica de Cartagena

Indique uno o varios de los seis temas de Interés: (Marque con una {x})

{x} Experiencias de innovación apoyadas en el uso de TIC. Nuevos escenarios tecnológicos para la enseñanza y el aprendizaje.

Idioma en el que se va a realizar la defensa: (Marque con una {x})

Español Inglés

Resumen. El proceso de la enseñanza aprendizaje de los lenguajes de programación presenta ciertas dificultades relacionadas con la falta de formación previa. Además, no es sencillo lograr un avance armónico entre los conocimientos teóricos y el trabajo práctico. Parece conveniente disponer de herramientas complementarias de aprendizaje como el m-learning. Los autores han querido desarrollar una herramienta sencilla, aplicada al aprendizaje de un lenguaje de programación, con objeto de conocer la utilidad de este tipo de herramientas y poder recoger la opinión de los alumnos sobre la ayuda que ésta supone en su proceso de aprendizaje. En este artículo se presenta la experiencia educativa llevada a cabo con alumnos de primer curso de ingeniería, en la asignatura de Informática Aplicada, para reforzar los conocimientos en el aprendizaje de los lenguajes de programación C. Se describen los objetivos iniciales, el diseño de la herramienta; su uso. Se justifica la selección de ejercicios propuestos y, finalmente, se describe el uso de la herramienta y las opiniones de los alumnos y las conclusiones de los autores.

Palabras Claves: Lenguajes de programación, m-learning, aplicación Web.

Abstract. The process of learning programming languages has certain difficulties related to the lack of prior training. Furthermore, it is not easy to achieve a harmonious progress between theoretical knowledge and practical work. It seems appropriate to have additional learning tools such as m-learning tools. The paper describes a simple tool, applied to learning a programming language developed in order to determine the usefulness of these e-learning tools and being able to obtain the views of students on the help offered by the tool for learning process. The tool has been used by first year students of engineering, in the subject of Applied Computer Science, to enhance knowledge in learning programming languages. The paper describes the initial objectives and the design of the tool, as well as the selection of the exercises offered to the students by means of the tool. Finally, the use of the tool and the opinions of students and conclusions of the authors are described.

Keywords: Programming languages, m-learning, Web application.

1. Introducción

El aprendizaje de los lenguajes de programación en los alumnos de primer curso de ingeniería presenta dificultades relacionadas con el escaso desarrollo de ciertas competencias, necesarias para la programación de computadoras, no adquiridas en su formación previa a su llegada a la Universidad [1]. Estos inconvenientes son bien conocidos y han sido identificados por diversos autores [2], [3].

En la docencia de un lenguaje de programación con alumnos sin conocimientos previos, no es sencillo lograr un avance armónico entre teoría y práctica. Surge la conveniencia de desarrollar entornos de aprendizaje complementarios, como el m-learning, que ofrezcan la posibilidad de ejercitarse en la programación en cualquier lugar y momento [4]. Ya Brian Wilson Kernighan insistía en que “*The only way to learn a new programming language is by writing programs in it*”; son muchos los estudios que confirman la necesidad de programar y de la ejercitación continua y constante en la resolución de problemas para lograr aprender a programar [5]-[7].

Mediante herramientas m-learning se pueden crear entornos de aprendizaje donde el estudiante ejercita y fortalece su conocimiento [8]-[10]. En los cursos de programación de computadoras, la adecuada aplicación de las tecnologías modernas tiene una considerable importancia [11]-[15], ya que permite a los estudiantes dedicar suficiente tiempo para resolver ejercicios en horario flexible y con retroalimentación constante.

En este artículo se presenta la experiencia educativa llevada a cabo con alumnos de primer curso de ingeniería, en la asignatura de Informática Aplicada, para reforzar los conocimientos adquiridos sobre el lenguaje de programación C. Una vez realizada la experiencia se ha preguntado a los alumnos sobre la eficacia de la herramienta.

Se describen, a continuación, los objetivos perseguidos al diseñar la herramienta; posteriormente se describe su uso y se muestra el lugar donde ha quedado alojada; se justifica, en el siguiente apartado, la selección de ejercicios propuestos; y posteriormente se describe el uso de la herramienta y las opiniones de los alumnos. Por último, se presentan las conclusiones de los autores.

2. Objetivos que perseguían los profesores al diseñar la herramienta

Nuestros alumnos son usuarios habituales de las tecnologías móviles. Prácticamente todos disponen de un Smartphone y de un ordenador portátil; en muchos casos de ambas cosas. Cada vez son más numerosos los que, además, disponen de tableta y la usan de forma ordinaria en clase y en su estudio personal. Son hábiles con la tecnología: más que muchos de sus profesores. En el caso de los Smartphone su limitación principal es el tamaño de su pantalla.

En el mundo académico se insiste en la utilidad del uso de la tecnología móvil en la docencia y en los procesos de aprendizaje. Quizá se ha llegado a un punto de mitificación, donde se confunden deseos con realidades, y donde se pasa fácilmente de verlas como herramientas útiles a percibir las como herramientas necesarias. Los autores piensan que el prestigio de estas tecnologías en el ámbito de la formación

académica se basa en sus posibilidades y no tanto en las capacidades demostradas por las herramientas actualmente disponibles, cuyo uso no está, ni mucho menos, generalizado.

Cuando un docente se plantea la posibilidad de incorporar el m-learning a su trabajo, se encuentra con algunas dificultades: (1) no existe un directorio útil donde localizar una herramienta adaptada a sus necesidades; (2) no resulta en absoluto sencillo desarrollar una herramienta propia; y (3) no hay garantías de que ese esfuerzo del docente o de la institución tenga una efectividad que lo haga rentable.

Los autores han querido desarrollar una herramienta m-learning sencilla, aplicada al aprendizaje de un lenguaje de programación, con objeto de conocer la utilidad de este tipo de herramientas y poder recoger la opinión de los alumnos sobre la ayuda que ésta supone en su proceso de aprendizaje.

3. Descripción de la herramienta

La aplicación permite escribir código en C, y someterlo a un proceso de evaluación. No es un entorno de desarrollo completo, sino limitado, que permite trabajar desde cualquier plataforma móvil sin tener instalado ningún compilador. La implementación de la herramienta (editor, intérprete de C a JavaScript, mensajes de error, base de datos, etc.) ha supuesto un mes de trabajo de un programador experimentado. En la plataforma se puede encontrar una colección de problemas que se resuelven con un pequeño programa en C. Los alumnos pueden ejercitarse en los conceptos trabajados en clase, y obtener evaluación inmediata de ese trabajo. Es necesario que el dispositivo desde donde el alumno trabaja tenga acceso a la red. La herramienta está disponible en <http://moeds.herokuapp.com/>. Cualquiera puede darse de alta como profesor o como alumno vinculado a un profesor.

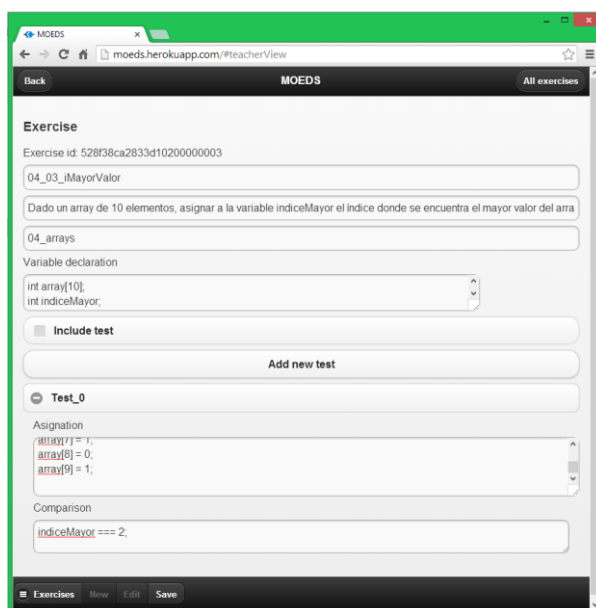


Figura 1. Ejemplo de definición de ejercicio.

Para incluir un nuevo ejercicio, el profesor debe (1) dar de alta el enunciado; (2) incorporar opcionalmente la declaración de variables e instrucciones que sean parte de la solución; y (3) también opcionalmente, incluir test de evaluación, que son resultados esperados ante una entrada concreta. En la Figura 1 se puede ver el enunciado 04_03_iMayorValor, que está en la colección de ejercicios 04_arrays. El profesor puede averiguar qué alumnos han intentado resolver algún ejercicio, quiénes lo han conseguido y cuántos errores sintácticos o de algoritmia han cometido.

Al alumno se le ofrecen varias listas o colecciones de ejercicios. El alumno intenta un ejercicio de una lista y somete el código a revisión. Si hay un error sintáctico, la herramienta advierte de su existencia sin indicar su lugar; si el código es sintácticamente correcto, la herramienta indica qué test se han pasado y cuáles no. Cuando todos los test han sido superados, el ejercicio se da por correcto. Si el alumno resuelve todos los ejercicios de una lista, queda indicado en la herramienta. El alumno sabe en todo momento los ejercicios pendientes de una solución correcta definitiva. En la Figura 2 se muestran distintas pantallas de la herramienta vista por el alumno.

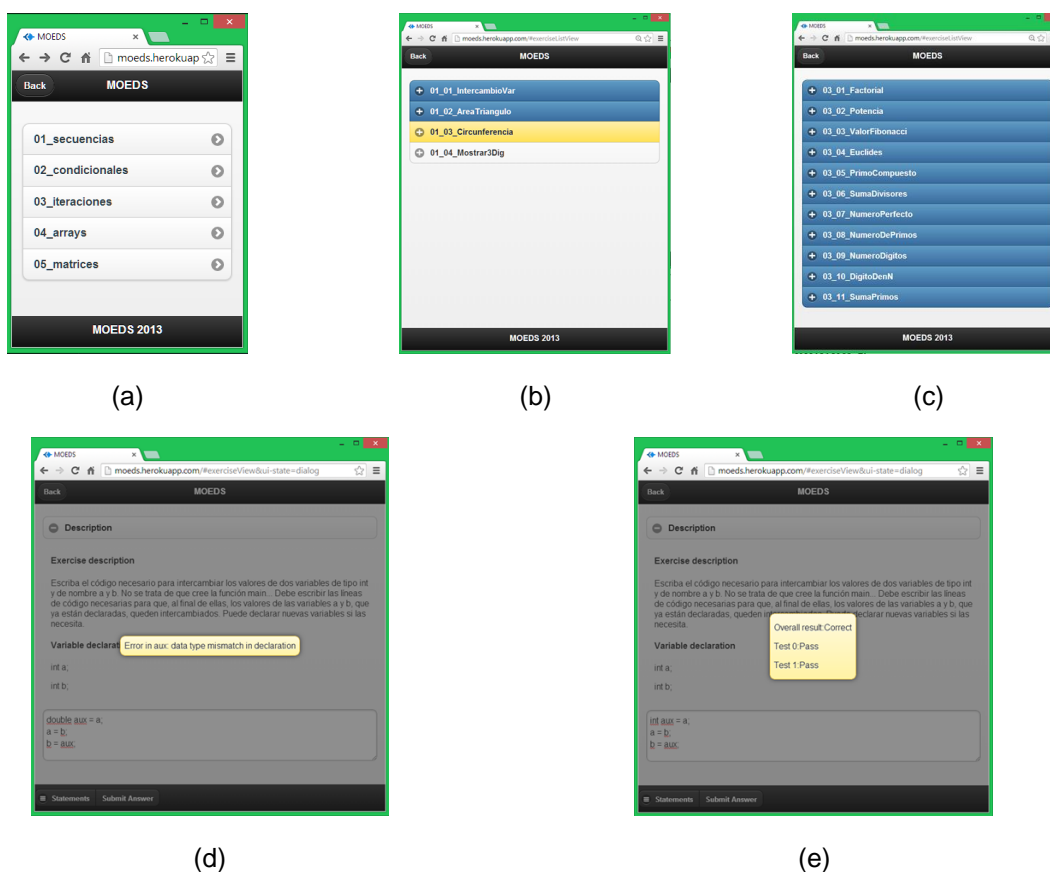


Figura 2: Distintas presentaciones de pantalla de la aplicación. (a) Las cinco colecciones ofrecidas a los alumnos. (b) Ejercicios de la primera colección (secuenciales): en la imagen el alumno ha logrado resolver correctamente los dos primeros; no ha realizado ningún intento con el cuarto; y ha intentado el tercero sin conseguir una solución correcta. (c) Aspecto del listado de la tercera colección cuando el alumno ha logrado resolver todos sus enunciados. (d) Mensaje de error sintáctico. (e) Mensaje correspondiente a una solución correcta.

La aplicación ofrece dos formas de introducir las soluciones: mediante un editor de

texto y con una paleta, cuyos detalles se especifican en cajas de diálogo. En la Figura 3 se muestra una vista general de la paleta y algunos ejemplos de su uso. La generación de la paleta supuso gran parte del esfuerzo de desarrollo de la aplicación, y causa de alguna de las limitaciones y restricciones impuestas sobre el código en C; sin embargo, no fue apenas empleada.

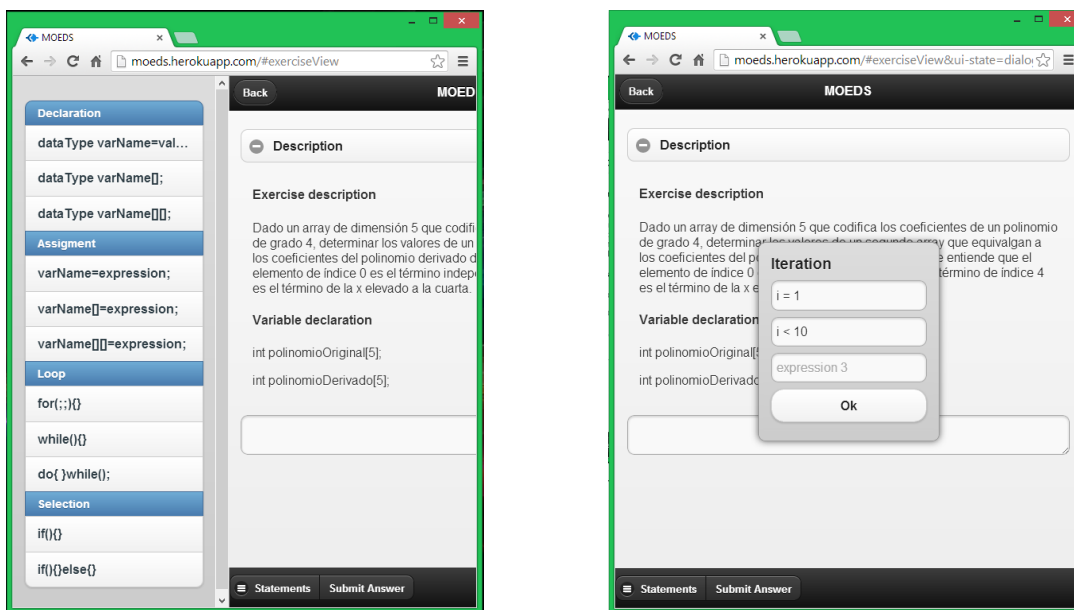


Figura 3: Paleta de edición de sentencias. (a) Vista de la paleta. (b) Vista de la aplicación cuando se ha seleccionado el bloque de iteración for; el código se genera y se inserta en el editor de forma automática.

Al ser web, la aplicación se hace independiente del terminal y de su OS. Además, el código del alumno se compila y ejecuta en su propio terminal. Eso elimina cualquier problema de seguridad en el servidor (código malicioso, acceso mediante punteros a distintas zonas de memoria, etc.), pero ha obligado a elaborar un mini compilador de C hacia JavaScript, que en esta primera versión tiene bastantes limitaciones:

- **Tipos de dato:** sólo **char**; **int**, y **double**. No se pueden usar modificadores de tipo. Sólo se puede trabajar en base decimal.
- **Operadores:** Salvo el operador signo negativo, no puede hacerse uso de ningún operador monario (tampoco incremento ++ o decremento --). No pueden utilizarse operadores a nivel de bit, ni operadores compuestos. No puede hacerse uso del operador coma (,). Tampoco puede hacerse uso del operador ternario (? :).
- **Expresiones.** No se puede combinar, en una expresión, literales, variables o expresiones de distinto tipo. No se admite el casting.
- **Sentencias:** no se admite el uso de la estructura **switch**, ni de las sentencias **break**, **continue** o **goto**.
- **Funciones:** No se admite el uso de función alguna.

Aunque esta lista de restricciones es importante, no imposibilitan la resolución de los ejercicios. Estas restricciones han supuesto una incomodidad para los alumnos, y han causado algunos errores sintácticos.

4. Selección de ejercicios

Se ha elaborado una lista de problemas tipo, de complejidad creciente, clasificados según las distintas estructuras de control y de datos necesarios para solventarlos. De todos ellos, se ha incluido en la aplicación un subconjunto: aquellos que más fácilmente podían solucionarse con el subconjunto de C soportado por la herramienta. Cada uno de los problemas incorpora una dificultad nueva, o una variante típica. En cualquier caso, son problemas clásicos en un curso introductorio a la informática. Las colecciones de ejercicios que se han usado en la herramienta son:

1. Ejercicios de solución secuencial, sin necesidad de estructuras de control ni de datos.
2. Ejercicios que se solucionan con estructuras condicionales: secuencia de condicionales independientes; árboles de condicionalidad; estructuras híbridas de condicionalidad.
3. Ejercicios sin arrays que se solucionan con estructuras de iteración: Repetición n veces de una operación; Repetir una operación mientras que se cumpla una condición; iterar hasta encontrar ejemplo o contraejemplo, o hasta terminar una lista; iteraciones anidadas; etc.
4. Ejercicios con arrays: recorrido simple con un índice de un array, donde cada valor es independiente de los otros; recorrido y procesado de un array donde cada valor depende de otros anteriormente calculados; recorrido de un array con más de un índice, etc.
5. Ejercicios con matrices: recorrer una matriz con un solo índice; anidamiento en el recorrido de matrices; etc.

La herramienta no ofrece colecciones exhaustivas de problemas, sino conjuntos seleccionados y restringidos, donde se ha intentado abarcar el mayor número de soluciones tipo posibles. Por ejemplo, se puede preguntar cuántos dígitos tiene un entero; o preguntar cuáles dígitos son; o averiguar si un dígito determinado está presente en un número dado; o mostrar cuántas veces aparece ese dígito en el número. Son ejercicios aparentemente similares, pero cuando se buscan sus soluciones se descubre que cada cuestión lleva a una solución sensiblemente distinta.

5. Presentación de resultados

El total de alumnos en la asignatura y en el presente curso son 167. De ellos, 75 han usado alguna vez la herramienta y han logrado resolver correctamente al menos un ejercicio. De entre ellos, 23 han usado de forma muy significativa esta herramienta (han resuelto 25 ó más de los 36 ejercicios propuestos). La Figura 4 muestra el porcentaje de alumnos que, en cada franja de notas, han usado la herramienta de forma significativa.

Se ha confeccionado un cuestionario con siete secciones y un total de 42 preguntas. De los 23 alumnos que han usado la herramienta de forma significativa, 18 han contestado a las preguntas del cuestionario. Vamos a centrar el análisis de la evaluación de la herramienta en las opiniones de estos 18 alumnos.



En la primera sección del cuestionario, los alumnos evalúan el funcionamiento e interfaz de la herramienta. En general piensan que el diseño es bueno y que la herramienta funciona correctamente: no se bloquea, tiene respuesta inmediata, es sencilla de usar, etc. Consideran que es de fácil uso en cualquier momento y lugar (9 sobre 10) Echan en falta mejores mensajes de información cuando la aplicación encuentra errores sintácticos. Entre los aspectos negativos destacan la falta de utilidad de la paleta de instrucciones (1,7 sobre 10), y que los alumnos consideran que la herramienta suscita escaso interés para ser utilizada y que no es entretenida (5,4 y 5,3 sobre 10 respectivamente).

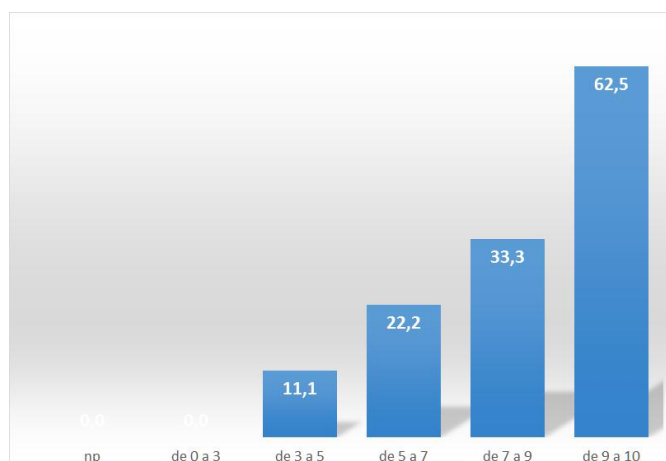


Figura 4. Porcentaje del uso masivo de la herramienta en cada franja de notas del examen de evaluación de los temas correspondientes

En la segunda sección los alumnos han opinado sobre la utilidad de la herramienta para aprender los conceptos de la asignatura. La opinión es que la herramienta es moderadamente útil (6,3 sobre 10), pero en la misma medida también opinan que podrían haber realizado los ejercicios con su ordenador, en cualquier otro momento y lugar sin hacer uso de la herramienta. Los alumnos han valorado especialmente útil las colecciones de problemas, no la plataforma móvil.

En la tercera sección se estudiaba el tiempo de uso de la herramienta. La mayor parte de los alumnos ha trabajado con ella más de 10 horas, y el 50% afirma que ya se sentía cómodo con ella a partir de la segunda hora.

En la cuarta sección se ha evaluado cómo los ejercicios han ayudado a comprender los conceptos o a afianzarlos. La opinión de los alumnos es bastante positiva, especialmente en el manejo de las condiciones de permanencia en las iteraciones y en el modo en que se han de recorrer los arrays. Los alumnos valoran especialmente bien la secuenciación de los ejercicios (casi un 8 sobre 10). Ninguna de las valoraciones de esta sección se refiere a la aplicación en sí, sino a los ejercicios presentados en ella. Es la sección que recoge mejor valoración: los alumnos valoran el contenido de la aplicación más que la plataforma.

En la quinta sección del cuestionario se evalúa la utilidad de la aplicación para motivar el estudio y como herramienta de aprendizaje. Los alumnos no consideran que la herramienta sea valiosa en sí misma, aunque estiman que es un buen complemento al resto de actividades de aprendizaje. Opinan que el tiempo invertido

en ella está bien empleado, aunque la valoración no es muy alta (6,11 sobre 10). Se incentivó el uso de la aplicación concediendo, a los alumnos que resolvieran con ella todos sus ejercicios, un punto de más en la prueba escrita que evaluaba los conceptos trabajados en la aplicación. De los 80 alumnos que se han dado de alta, 17 han resuelto todos los ejercicios; el resto ha utilizado la herramienta casi exclusivamente como una colección útil de problemas para su estudio personal, al margen de la interfaz de trabajo que la herramienta ofrecía.

En la sexta sección los alumnos informaron sobre su acceso a la aplicación. Casi ninguno dispone de Tablet, pero prácticamente todos disponen de un Smartphone. Sin embargo, el uso de la aplicación se ha realizado fundamentalmente desde equipos portátiles (laptops): 549 accesos desde Smartphone frente a 8140 accesos desde laptop. Los alumnos han trabajado fundamentalmente desde sus casas y, en menor medida, desde el campus de la universidad. No han trabajado en desplazamientos, o espacios de descanso, ni en otros momentos.

Por último, en la última sección del cuestionario, se pedía la opinión global a cada alumno. Las ideas más repetidas entre los muchos comentarios registrados son las siguientes:

- Críticas a las distintas limitaciones de la aplicación: limitaciones sintácticas; excesiva brevedad y falta de concreción de los mensajes de error; el hecho de que no pudiera almacenarse en disco las versiones previas de las soluciones introducidas por el alumno. También se ha criticado las limitaciones del editor de código, la brevedad de los enunciados, y el no poder disponer de las soluciones para cotejar con el código propuesto.
- Al hacer una valoración global, en general los alumnos han calificado muy positivamente la aplicación. Destacan la utilidad de disponer de una herramienta que ofrece una colección de ejercicios ordenada en temas y complejidad, accesible desde cualquier terminal y cualquier sitio y momento, con la posibilidad de obtener una corrección inmediata a las soluciones propuestas.
- Los alumnos han agradecido explícitamente el esfuerzo de los profesores en el desarrollo y puesta en marcha de la aplicación.

6. Conclusiones

No se ha pretendido hacer una valoración o abrir debate sobre enseñanza presencial *vs on line*. No cuestionamos la validez de ninguno de los dos modelos: desde luego, cada uno de ellos tiene su ámbito y su clientela. Tampoco queremos hacer una valoración sobre las herramientas m-learning en general: es evidente que tienen sus muchas aplicaciones útiles para el alumno: videos, transparencias, animaciones, etc. En la formación universitaria presencial este tipo de herramientas puede jugar un valioso papel de complemento o apoyo a la labor de enseñanza-aprendizaje. En la formación on line, posiblemente su papel sea fundamental, y posiblemente los Centros Universitarios no puedan prescindir de ellas si pretenden ser competitivos.

No podemos sacar conclusiones generales sobre m-learning, o sobre las aplicaciones en general sólo a la luz del análisis de una aplicación concreta aquí presentada, pero

sí podemos sacar alguna conclusión sobre el uso del m-learning para el aprendizaje de un lenguaje de programación del estilo del C.

Cuando el alumno ha podido elegir la plataforma, la decisión ha sido abrumadoramente mayoritaria en la elección del laptop: 8140 accesos, frente a sólo 549 desde Smartphone (94% de accesos por laptop). La programación exige muchas herramientas que deben estar simultáneamente disponibles, y la pantalla de un móvil, e incluso de la de una Tablet, no dan para ello. Quienes trabajamos en la programación nos procuramos pantallas (a veces más de una) de muchas pulgadas.

Las principales limitaciones en la interfaz gráfica han venido exigidas por la pretensión de que fuera una aplicación válida para trabajar desde smartphones. Se ha priorizado el m-learning sobre el *learning*.

Los alumnos se han quejado de las limitaciones de la aplicación: restricciones en C, en los editores, en los mensajes de error, etc. Están acostumbrados a trabajar con un IDE de verdad, y al ofrecerles algo limitado se sienten defraudados. Toda aplicación e-learning debe gozar de un nivel mínimo de calidad y funcionalidad para que se vea como útil: debe ser comparable al IDE mismo, u ofrecer alguna prestación valiosa que el IDE no incluya. En nuestro caso la corrección automática, por ejemplo, ha sido muy bien valorada por los alumnos. Lograr una aplicación con esa calidad y prestaciones no es nada sencillo, y requeriría muchas jornadas de trabajo de un programador experimentado: va más allá de lo que podría hacer un equipo docente/investigador en su trabajo diario. Y además, habría que ofertarla para dos plataformas diferentes (Smartphone y laptop), para que las prestaciones del laptop no se vieran comprometidas por limitaciones del Smartphone.

La principal aportación que la herramienta ha supuesto para los docentes es la realimentación inmediata respecto del trabajo del alumno. Desde luego, hubiera sido un trabajo ingente para los profesores corregir manualmente tantos ejercicios para tantos alumnos (80 han hecho uso de la aplicación, 8689 entregas realizadas). Además, hemos recibido una información exacta, puntual y concisa sobre errores más frecuentes y en qué ejercicios los alumnos hallan mayor dificultad para llegar a una solución correcta. Esa información ayuda enormemente a adaptar la docencia al real aprendizaje del alumno que trabaja. Esa información y esa ayuda sí merecen, a nuestro juicio, la merma que supone trabajar con las limitaciones que la herramienta exige.

Bibliografía y Referencias.

- [1] Lira Tavares, O.; Menezes, C.S.; Nevado, R.A., “Pedagogical architectures to support the process of teaching and learning of computer programming” FIE, 2012, pp. 1 – 6. DOI: 10.1109/FIE.2012.6462427
- [2] Jenkins T. “On the difficulty of learning to program.” 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences, 2002, p. 53–58.

- [3] McCracken M, Wilusz T, Almstrum V, Diaz D, Guzdial M, Hagan D, et al. "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students." ITiCSE-WGR '01, New York, USA: ACM Press; 2001, p. 125.
- [4] Hunter, G. ; Livingstone, D. ; Neve, P. ; Alsop, G., "Learn Programming++: The Design, Implementation and Deployment of an Intelligent Environment for the Teaching and Learning of Computer Programming" IE, 2013 9th, pp. 129–136. DOI: 10.1109/IE.2013.46.
- [5] Menezes, C. S., Tavares, O. L., Nevado, R. A., Cury, D. "Computer supported cooperative systems to support the problem solving-a case study of learning computer programming." Proceedings of the FIE 38th, 2008. v1, p.1–6.
- [6] Polya, G., "The Art of Solving Problems", Editora Interciência, 1st ed., 1978.
- [7] Ribeiro, L.R.C.; Mizukami, M.G.N. "Problem-based learning (PBL) in higher education: the model of the faculty of engineering and the environment built at the University of Newcastle". COBENGE 2004, Brasília, 2004.
- [8] Anderson, T., & Elloumi, F. (eds.). "Theory and Practice of Online Learning." http://cde.athabasca.ca/online_book/pdf/TPOL_book.pdf (Athabasca University, October, 29, 2008).
- [9] Dietinger, T. (2003). "Aspects of e-learning environments." Unpublished doctoral dissertation, Graz University of Technology, Austria.
- [10] Wentling, L.T., Waight, C., Gallaher, J., Fleur, J.L., Wang, C., & Kanfer, A. "E-learning: a review of literature." <http://learning.ncsa.uiuc.edu/papers/elearnlit.pdf> (Knowledge and Learning Systems Group, University of Illinois, May, 06, 2008).
- [11] A. Cartelli, "Teaching in the Knowledge Society: New Skills and Instruments for Teachers." London, U.K.: Information Science, 2006.
- [12] S. Petrina, "Advanced Teaching Methods for the Technology Classroom." London, U.K.: Information Science, 2007.
- [13] T. Boyle, C. Bradley, P. Chalk, R. Jones, and P. Pickard, "Using blended learning to improve student success rates in learning to program," J. Educ. Media, vol. 28, no. 2–3, pp. 19–30, 165–178, 2003.
- [14] S. Hadjerrouit, "Towards a blended learning model for teaching and learning computer programming: A case study," Inform. Educ., vol. 7, no. 2, pp. 181–210, 2008.
- [15] S. Hadjerrouit, "A blended learning model in java programming: A design-based research approach," in Proc. Comput. Sci. IT Educ. Conf., 2007, pp. 283–308.