

LITE (Laboratory of Information Technologies in Education), URJC

J. Ángel Velázquez-Iturbide, Diana Pérez-Marín, Oriol Borrás-Gené, César Cáceres-Taladriz, Miguel García-Iruela, Marta Gómez-Gómez, Isidoro Hernán-Losada, Raquel Hijón-Neira, Carlos A. Lázaro-Carrascosa, Estefanía Martín-Barroso, Mercedes Martín-Lope, Daniel Palacios-Alonso, Pedro Paredes-Barragán, Maximiliano Paredes-Velasco, Celeste Pizarro, Liliana P. Santacruz-Valencia, Jaime Urquiza-Fuentes, María Zapata-Cáceres

Escuela Técnica Superior de Ingeniería Informática
Universidad Rey Juan Carlos
C/ Tulipán s/n, 28933 Móstoles, Madrid
 {angel.velazquez,diana.parez}@urjc.es

Resumen: Se presenta el grupo de investigación LITE (Laboratory of Information Technologies in Education), de la URJC. Sus actividades pueden clasificarse según dos criterios: ámbito educativo y tecnologías informáticas. Con respecto al ámbito, buena parte de su investigación se ha orientado al aprendizaje de la programación, desde la educación infantil hasta la universidad. Otras áreas son el aprendizaje de los algoritmos, la interacción persona-ordenador y la competencia digital, y el uso de la taxonomía de Bloom. Desde el punto de las tecnologías informáticas, destacan las investigaciones realizadas sobre visualización del software, apoyo a la clase invertida, uso de vídeos, sistemas colaborativos, mecanismos de gamificación, MOOCs y sistemas de experimentación con algoritmos.

Palabras clave: enseñanza de la informática, aprendizaje basado en vídeos, clase invertida, CSCL, gamificación, MOOCs, sistemas de experimentación con algoritmos, visualización del software

Abstract: The research group LITE (Laboratory of Information Technologies in Education), at the URJC, is presented. Its research activities can be classified according to two criteria: educational scope and computing software. With respect to educational scope, a substantial percentage of the research was oriented to programming education, from early childhood education to college education. Other educational fields are the learning of algorithms, human-computer education and digital competency, and the use of Bloom's taxonomy. With respect to software technologies, we may remark software visualization systems, support to flipped-classroom, support to videos, collaborative systems, gamification mechanisms, MOOCs and algorithm experimentation systems.

Key words: computing education, video-based learning, flipped-classroom, CSCL, gamification, MOOCs, algorithm experimentation systems, software visualization

1. Introducción

El grupo LITE (Laboratory of Information Technologies in Education) fue uno de los participantes en un monográfico anterior de esta misma revista sobre grupos españoles que investigan en informática educativa (Velázquez-Iturbide *et al.*,

2012)¹. Presentamos en este nuevo número monográfico una actualización de las actividades del grupo. Dado el carácter multidisciplinar de las publicaciones, las agrupamos en dos secciones, centradas en educación y en informática educativa. Terminamos identificando nuestras líneas de trabajo actuales.

¹ Dado el alto número de miembros de LITE y para evitar excesivas referencias, no se han incluido referencias externas

ni de miembros del grupo incluidas en el monográfico de IE Comunicaciones de 2012.

2. Enseñanza de la Informática

El principal ámbito educativo en el que investiga el grupo LITE es la Informática. Un estudio destacado fue el realizado por un grupo de trabajo de la Sociedad Científica Informática de España (SCIE), coordinado por Ángel Velázquez. Dada la creciente importancia que ha ido adquiriendo el aprendizaje de la Informática en nuestra sociedad, se creó dicho grupo de trabajo con el encargo de elaborar un informe con recomendaciones concretas para España. En resumen, el informe (Velázquez-Iturbide *et al.*, 2018) propugna el reconocimiento oficial de una nueva materia educativa (Informática), cuyo aprendizaje debería ser obligatorio desde la Educación Primaria hasta el Bachillerato. Dicha materia englobaría tanto competencia digital como fundamentos de la ciencia y la ingeniería informáticas. El informe incluía recomendaciones detalladas de contenidos, estructurados en cinco áreas: programación, ordenadores y sistemas operativos, redes e Internet, contenido digital y cooperación, y seguridad.

En esta sección se presentan los diversos trabajos de investigación realizados por el grupo LITE sobre la enseñanza de la Informática, sobre todo de la programación, la algoritmia, la competencia digital y la interacción persona-ordenador.

2.1. Programación

La programación es la materia más importante y más problemática en la formación en Informática. Se ha abordado su aprendizaje en todas las etapas educativas.

2.1.1 Educación Infantil y Primaria

Se realizó un análisis de la situación en España de la educación informática en Educación Primaria, y se diseñó una metodología, basada en el uso de metáforas, para cubrir varios conceptos de programación en Educación Primaria. Los experimentos realizados en colegios al aplicar esta metodología usando programas como PrimaryCode² (desarrollado por el grupo, ver Figura 1) y Scratch han validado la metodología y la capacidad de los niños de mejorar significativamente sus conocimientos de programación (Pérez-Marín *et al.*, 2018).

² <http://www.lite.etsii.urjc.es/tools/primarycode/>

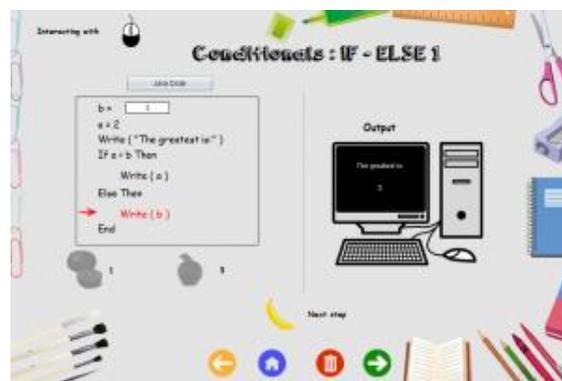


Figura 1. Aspecto de PrimaryCode (fuente: <http://www.lite.etsii.urjc.es/tools/primarycode/>)

También se ha estudiado el uso didáctico de dispositivos tangibles como Makey-Makey en Educación Primaria (Pérez-Marín *et al.* 2019). Del mismo modo, se ha trabajado en el uso de metodologías docentes innovadoras como aprendizaje-servicio para la transferencia a centros de Educación Infantil y Primaria (Santacruz-Valencia *et al.* 2013).

Para evaluar la idoneidad de las estrategias basadas en juegos como herramientas de aprendizaje de la programación en Educación Primaria, y la evaluación de los conocimientos adquiridos a través de analíticas en tiempo real, se han desarrollado varias herramientas complementarias, un test adaptado a edades tempranas y un videojuego colaborativo (Blue Ant Code, ver Figura 2). Se ha realizado una experiencia con resultados significativos.



Figura 2. Aspecto de Blue Ant Code

Por último, en las etapas de Educación Infantil, Primaria y Educación Especial (Martín *et al.* 2019) también se ha trabajado con dispositivos táctiles como tabletas y pizarras digitales con el programa DEDOS (Roldán-Álvarez *et al.* 2018) resolviendo distintos tipos de actividades personalizadas a cada uno de los niveles. Es destacable el programa “Today I tell”, diseñado especialmente para personas con autismo (Martín *et al.* 2018), que contribuye a la creación de historias y comics en dispositivos táctiles. Este tipo de dispositivos son muy adecuados en etapas tempranas y para alumnos con necesidades educativas especiales, dado que no requieren una motricidad fina y los niños manipulan los objetos con las manos (Roldán-Álvarez *et al.* 2016). Cabe destacar que han recibido varios premios.

2.1.2 Educación Secundaria y Formación Profesional

Dentro del ámbito de la Formación Profesional se comenzó trabajado en el análisis del empleo de juegos serios en el aula, mediante un análisis de la eficacia educativa del empleo de cinco aplicaciones desarrolladas para la enseñanza de la programación, obteniendo resultados positivos. Los alumnos realizaron una evaluación cualitativa de dichas aplicaciones.

Tras los juegos serios se decidió continuar con el estudio del empleo de la gamificación en Formación Profesional, realizando una pequeña experiencia (García-Iruela e Hijón-Neira, 2018) y una propuesta de gamificación de un curso entero. En Educación Secundaria, se ha estudiado el efecto del uso de *apps* educativas para el aprendizaje de la programación, con alumnos de 15-16 años.

2.1.3 Educación Universitaria

A nivel universitario, la principal aportación fue una propuesta para mejorar la comprensión de la ejecución de la recursividad (Velázquez-Iturbide *et al.* 2016). La propuesta consiste en presentar una regla de eliminación de la recursividad lineal mediante la conversión de un esquema general en un fragmento de código iterativo equivalente. El esquema de traducción general, que genera dos bucles, se complementa con los esquemas de varios casos particulares que sólo

necesitan un bucle. La propuesta se evaluó analizando la evolución de los modelos mentales de los alumnos, resultando en un aumento significativo de los modelos mentales viables.

2.2. Algoritmos

La investigación en algoritmos se ha centrado en dos cuestiones no disjuntas: adopción de un enfoque experimental al estudio de los algoritmos de optimización y la identificación de malas concepciones de los alumnos.

La experimentación con la propiedad de optimalidad de los algoritmos de optimización se propuso como forma de proporcionar un método activo al aprendizaje de los algoritmos voraces (Velázquez-Iturbide, 2013a). Se comparaban los resultados de diversos criterios voraces y el alumno debía determinar para cuáles había evidencia experimental de que podían ser criterios óptimos. Este método didáctico se apoya en el uso de las herramientas GreedEx, para los algoritmos voraces, y OptimEx, para algoritmos de optimización cualesquiera (véase sección 3.4).

Dicho método didáctico se ha evaluado con respecto a dos criterios. Por un lado, se ha evaluado su efecto sobre el rendimiento de los alumnos, encontrándose mejoras significativas, sobre todo al nivel de comprensión de la taxonomía de Bloom (Esteban-Sánchez *et al.* 2014). Por otro lado, el método experimental ha permitido identificar diversas malas concepciones de los alumnos sobre conceptos de optimización (Velázquez-Iturbide, 2013b). Su identificación ha permitido realizar intervenciones docentes que han reducido su número y gravedad.

Se han obtenido algunos resultados más generales. En primer lugar, se han analizado las dificultades de los alumnos al resolver prácticas de diversas técnicas de diseño de algoritmos (Velázquez-Iturbide, 2019). Se han identificado 8 malas concepciones, agrupadas en 5 bloques temáticos. En segundo lugar, se ha realizado una revisión de la enseñanza de los algoritmos de optimización mediante la experimentación (Velázquez-Iturbide *et al.* 2015), tratando en profundidad aspectos como métodos de enseñanza, sistemas software educativos y evaluaciones docentes.

2.3 Interacción Persona-Ordenador

La Interacción Persona-Ordenador es una asignatura con dos perfiles diferenciados dependiendo de dónde se ponga el foco, en el proceso de diseño o en el de desarrollo. En la URJC se da prioridad al enfoque de diseño, basándonos en el diseño centrado en el usuario. Para enseñar esta asignatura se está usando una metodología basada en prácticas realistas (Urquiza-Fuentes y Paredes-Velasco, 2017) que mejora la motivación de los estudiantes y donde se investiga mejoras en el trabajo grupal, así como la evaluación de las competencias.

2.4 Taxonomía de Bloom

Se ha estudiado en profundidad el uso de la taxonomía de Bloom para ejercicios de programación. Se realizó una revisión sistemática de la misma (Masapanta-Carrión y Velázquez-Iturbide, 2018), destacando la clasificación de los problemas reportados por diversos autores para su uso, las hipótesis que aventuran sobre sus causas y las soluciones adoptadas para evitarlos.

Posteriormente se analizaron posibles formas de evitar las divergencias de los profesores al clasificar diversos ejercicios de programación en procesos cognitivos (Masapanta-Carrión y Velázquez-Iturbide, 2019). Se midió la eficacia de dos medidas: la presentación de ejemplos de programación para todos los procesos cognitivos y aclarar que resolver un ejercicio puede implicar varios procesos cognitivos. Aunque los profesores experimentaron cierta mejora en sus clasificaciones, en sus clasificaciones tuvieron porcentualmente mayor confianza que acierto. Se concluyó que el aprendizaje de la taxonomía de Bloom requería mucho más tiempo del estimado inicialmente.

3. Informática Educativa

3.1 MOOCs

Los MOOC son el acrónimo en inglés de “cursos abiertos, masivos y en línea”, los cuales permiten compartir el conocimiento generado, principalmente en universidades, acercándolo a cualquier persona que acceda a él mediante un usuario en plataformas específicas virtuales (Coursera, edX, FutureLearn, MiriadaX, etc.) y con la posibilidad de obtener un certificado de pago. Una de las principales críticas a los MOOC es el modelo pedagógico que hay detrás,

desaprovechando el potencial del gran número de estudiantes y con unas bajas tasas de finalización.

El modelo de producción de MOOC de la Universidad Rey Juan Carlos (Becerra-Jimenez *et al.* 2018) ha conseguido integrar diversos elementos para conseguir cursos de alta calidad, como el MOOC “Poténciate con redes sociales”, ganador del Primer Premio de Innovación Educativa MiriadaX 2018. En este MOOC se ha intentado mejorar la interacción y el compromiso o fidelización de los participantes del MOOC mediante nuevos elementos de gamificación y a través del uso de redes sociales como comunidades virtuales de aprendizaje. Como resultado, se han obtenido exitosos grupos de colaboración en Facebook, una mayor implicación del participante y una mejora sustancial de los datos de finalización, duplicando los resultados habituales.

3.2 Clase invertida

El auge del aprendizaje inverso o *flipped learning* como metaestrategia para la aplicación de metodologías activas de aprendizaje ha revolucionado la enseñanza universitaria en los últimos años. Su aplicación en las asignaturas de la rama informática es cada vez mayor, combinando esta técnica con el trabajo colaborativo, el aprendizaje entre iguales, la enseñanza *just-in-time* o la gamificación.

De momento, el contexto de trabajo en esta línea ha sido universitario, tanto a nivel de máster como de grado (Cáceres-Taladriz, 2018). Así, hemos realizado una evaluación en el Máster Universitario en Formación del Profesorado de Educación Secundaria, Bachillerato, FP e Idiomas que imparte la Universidad Rey Juan Carlos. Los alumnos debían aprender a programar páginas web sencillas utilizando los lenguajes HTML, CSS y JavaScript. Con tal fin se utilizó la técnica aula invertida para dar a conocer los contenidos necesarios, combinada con una adaptación de una técnica de aprendizaje cooperativo conocida como puzzle de Aronson, utilizada para realizar una práctica grupal que reflejase los conocimientos adquiridos. Al concluir la experiencia, los alumnos manifestaron su satisfacción respecto a las técnicas empleadas y al aprendizaje adquirido. Además, las calificaciones de las prácticas fueron elevadas (Lázaro-Carrascosa *et al.* 2019).

Por otro lado, se ha estudiado la inclusión de esta metodología en la asignatura de Procesadores de Lenguajes, del Grado en Ingeniería Informática. Los resultados son variados, puesto que se ha mostrado eficaz (mejora significativa de un 17,7% sobre el grupo de control) cuando se usa sobre contenidos prácticos durante un corto espacio de tiempo (2 semanas). Sin embargo, cuando se usa sobre contenidos teóricos a más largo plazo (2 meses) los estudiantes apenas se involucraron en la dinámica de la metodología usando los materiales como apoyo, ya que ellos no cambian su metodología de trabajo. Parece por tanto que se necesita algún tipo de incentivo para involucrar más a los estudiantes en esta metodología que cambie los hábitos de trabajo de los estudiantes (Urquiza-Fuentes, 2019).

3.3 Visualización de software

Se han realizado trabajos en visualización del software en tres ámbitos: programación funcional, procesadores de lenguajes y algoritmos. La línea de trabajo dedicada a la visualización de programas funcionales ha demostrado cómo animaciones de estos se pueden convertir en herramientas educativas eficaces cuando son los estudiantes quienes generan dichas animaciones añadiendo explicaciones de los programas visualizados (Urquiza-Fuentes y Velázquez-Iturbide, 2013).

En lo referente al uso de tecnologías de visualización en la enseñanza de asignaturas de procesadores de lenguajes o de compiladores, se está trabajando tanto en la visualización de analizadores sintácticos como de traductores dirigidos por la sintaxis. La visualización de analizadores sintácticos permite salvar la distancia existente entre los conceptos teóricos en los que se basa y las herramientas de generación de *parsers* con un enfoque eminentemente práctico.

Un aspecto importante son las estrategias de recuperación de errores donde se llegan a crear subárboles sintácticos aislados, en la medida en que no cuentan para proceso posterior de la cadena de entrada. En la Figura 3 se pueden ver dichos subárboles, cuya raíz es un nodo "error" y del que cuelgan nodos erróneamente procesados y *tokens* directamente

ignorados, todos tratados durante el proceso de recuperación. Esta visualización permite a los estudiantes ser más eficientes al crear especificaciones sintácticas que usen este tipo de estrategias (Urquiza-Fuentes *et al.* 2013).

La continuación natural de esta línea de trabajo de la visualización del análisis sintáctico es la siguiente fase de un procesador de lenguajes, la traducción dirigida por la sintaxis. Ello conlleva la inclusión en la visualización de los atributos de los símbolos gramaticales y las acciones semánticas, así como su relación con la especificación del traductor. Los estudios realizados hasta el momento ofrecen resultados prometedores (Nicolalde-Rodríguez y Urquiza-Fuentes, 2018).

Los trabajos de visualización de algoritmos se han centrado en el sistema SRec de visualización de la recursividad. La contribución más destacada ha sido su ampliación para soportar una metodología de optimización de algoritmos con recursividad múltiple y redundante, como es el caso de los algoritmos de programación dinámica (Velázquez-Iturbide y Pérez-Carrasco, 2017). SRec permite buscar nodos redundantes en un árbol de búsqueda. En caso afirmativo, el usuario puede convertir el árbol de recursión en un grafo de dependencia y posteriormente distribuir sus nodos sobre un array de una o dos dimensiones. El diseño de un algoritmo memorizador o tabulador es inmediato a partir de esta visualización. La Figura 4 muestra sucesivamente el árbol de recursión, el grafo de dependencia generado y el árbol de dependencia colocado sobre una matriz, todo ello para una ejecución del problema del plan óptimo de sedes.

Otra nueva facilidad de SRec es permitir la visualización de un algoritmo con varios juegos de datos. Esta forma de presentar visualizaciones se denomina "múltiples ejecuciones" (Velázquez-Iturbide, 2016), por similitud con otras técnicas de presentación como la de múltiples vistas o múltiples algoritmos.

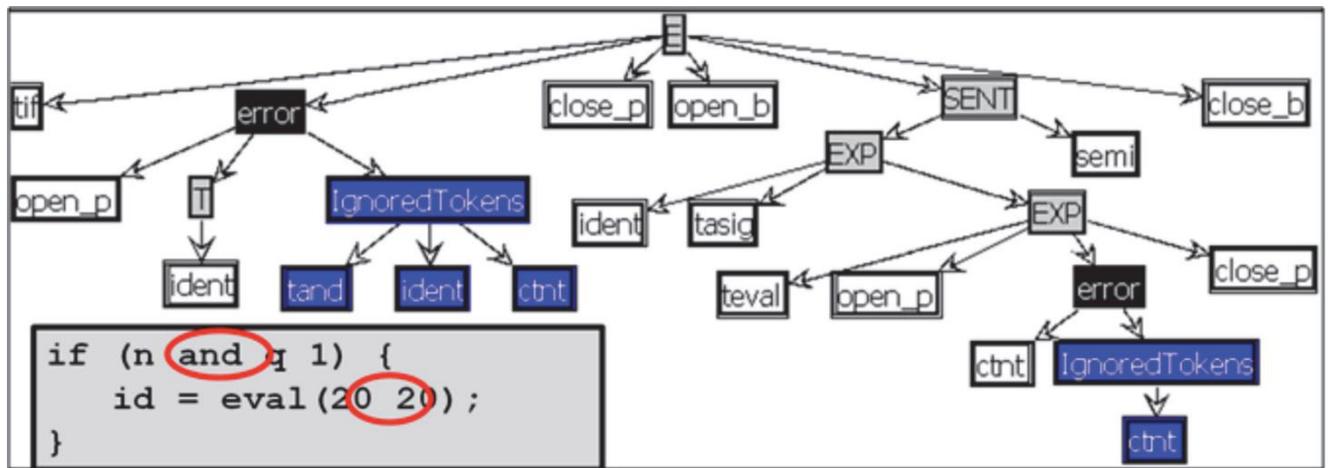


Figura 3. Visualización de estrategias de recuperación de sintáctica, el ejemplo concreto corresponde a la estrategia “modo pánico”, identificando el árbol correctamente procesado, y colgando del nodo “error”, la parte erróneamente procesada y la ignorada

Destaca la realizada evaluación del efecto de SRec sobre la motivación de los alumnos, dentro del marco de la teoría de la autodeterminación (Velázquez-Iturbide *et al.* 2017). Se midieron aumentos significativos de los dos tipos de motivación con mayor determinación, la motivación implícita y la explícita vía regulación identificada. La realimentación proporcionada por los alumnos sobre su satisfacción con SRec confirmaban estos hallazgos.

Un último trabajo, más general, fue un estudio sobre las características de las ilustraciones de las técnicas de diseño en 20 libros de texto de algoritmos (Velázquez-Iturbide, 2013c). El estudio de tipo cualitativo identificó diversos criterios para analizar ilustraciones de 3 técnicas de diseño. Aunque dichos criterios muestran un panorama más amplio que el de las visualizaciones de programas, pueden ser útiles para el diseño de nuevas visualizaciones.

3.4 Sistemas para juzgar algoritmos

Hemos realizado diversas investigaciones con dos tipos de sistemas de valoración de algoritmos: correctores de programas y sistemas de experimentación. La primera clase de sistemas permite juzgar la propiedad de corrección de los algoritmos mientras que los segundos sistemas permiten juzgar la optimalidad y algo la eficiencia.

Con respecto a sistemas correctores de programas, presentamos el sistema FLOP (Llana *et al.*, 2014), cuyas principales características pueden resumirse en una gran flexibilidad para su uso docente, resultado de los criterios de diseño iniciales y de las lecciones aprendidas tras una evaluación de usabilidad. También se realizó una evaluación de la aceptación y de los efectos del sistema corrector Mooshak sobre los alumnos (Rubio-Sánchez *et al.*, 2014). Se encontró que los alumnos lo consideraron adecuado para su función, pero sus mensajes deberían mejorarse para aumentar su aceptación y no había evidencia de que ayudara a los alumnos a reducir la tasa de abandono.

Una segunda línea de investigación ha consistido en desarrollar sistemas que permitan que los alumnos experimenten con la optimalidad de los algoritmos, según el método didáctico esbozado en la sección 2.2. Se han desarrollado dos sistemas: GreedEx (Velázquez-Iturbide *et al.*, 2013) y OptimEx (Velázquez-Iturbide, 2017). GreedEx está concebido para una introducción a los algoritmos voraces, estando limitado a seis problemas, mientras que OptimEx admite algoritmos para problemas cualesquiera. Como consecuencia de esta diferencia, GreedEx incluye visualizaciones y criterios voraces y heurísticos predefinidos, mientras que OptimEx proporciona mayor flexibilidad experimental y da información sobre posibles errores en los algoritmos.

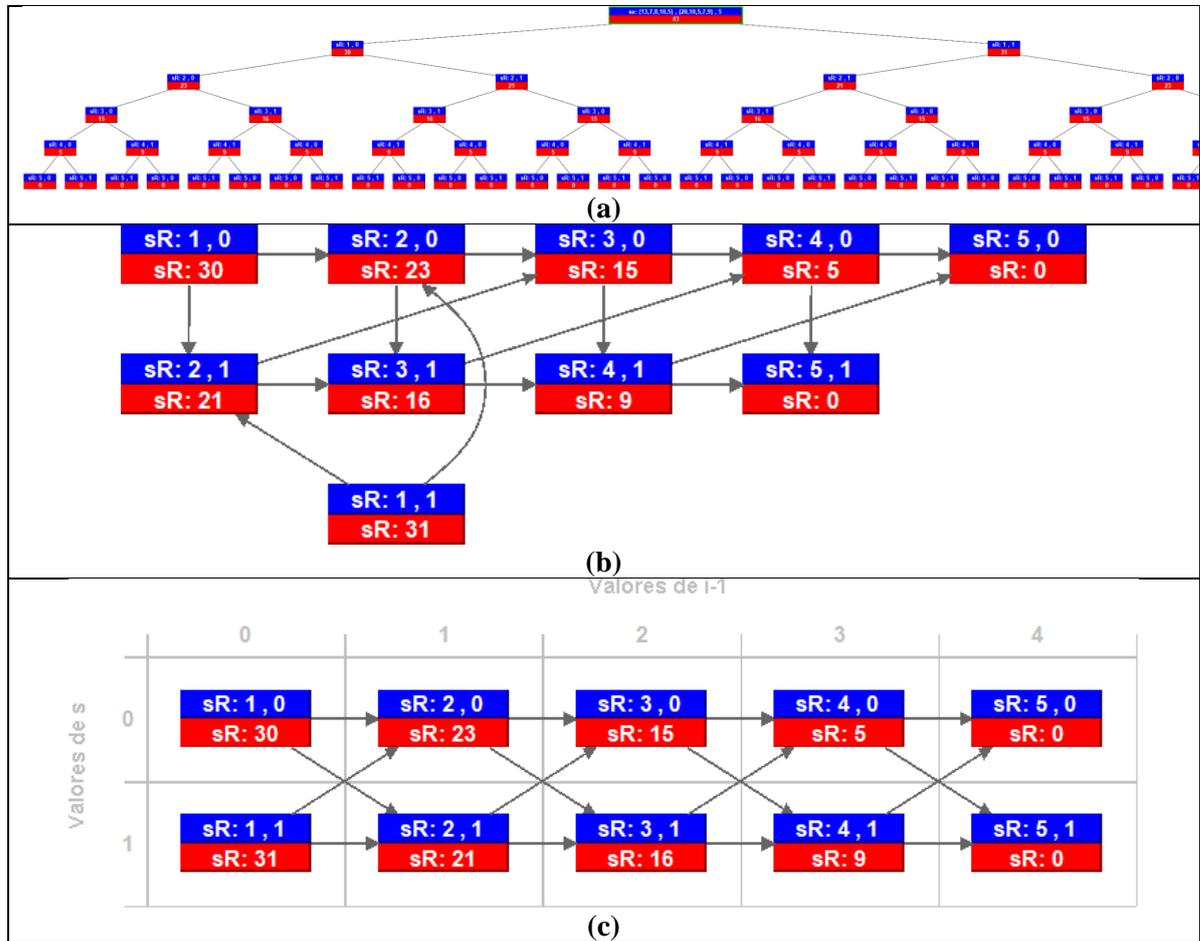


Figura 4. Visualizaciones generadas por SRec, para una ejecución del problema del plan óptimo de sedes: (a) árbol de recursión, (b) grafo de dependencia generado, y (c) grafo de dependencia colocado sobre una matriz mediante correspondencias parámetros-celdas.

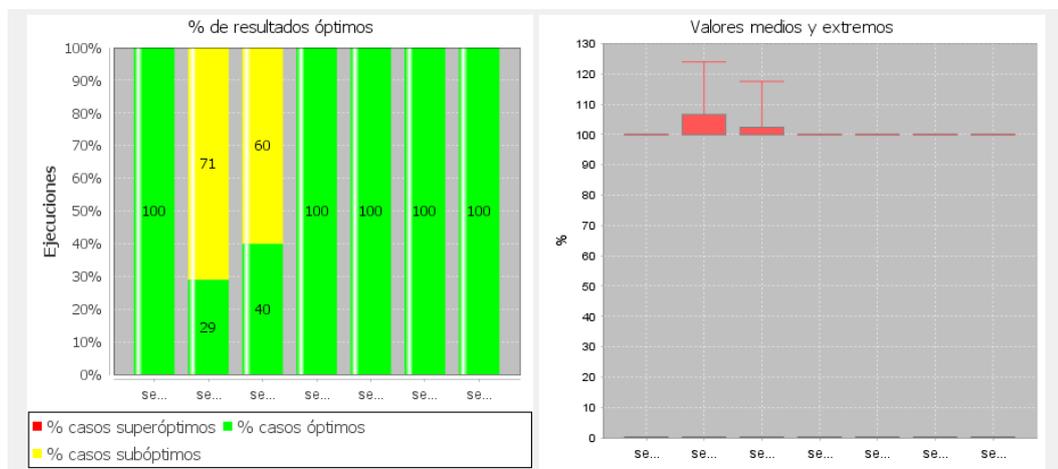


Figura 5. Diagramas generados por OptimEx para 100 ejecuciones del problema del plan óptimo de sedes

OptimEx es más flexible que GreedEx al permitir experimentar con la optimalidad y la eficiencia en tiempo de los algoritmos. En ambos casos se incluyen diagramas para mostrar los rangos y variaciones en los resultados o en los tiempos de ejecución medidos para los algoritmos de optimización a comparar. La Figura 4 muestra el porcentaje de resultados óptimos y las desviaciones de los algoritmos inexactos tras ejecutar 100 juegos de datos con 7 algoritmos distintos que resuelven el problema del plan óptimo de sedes (véase también Figura 5).

3.5 Sistemas colaborativos

Se han desarrollado dos sistemas educativos colaborativos. GreedExCol es una extensión colaborativa de GreedEx (véase la sección anterior) que permite a los alumnos debatir sobre los resultados obtenidos tras ejecutar algoritmos voraces y heurísticos (Debdi *et al.* 2015). Su evaluación permitió medir mejoras significativas en conocimientos y en motivación implícita de los alumnos.

Por otro lado, MoCAS es un sistema colaborativo diseñado para dar soporte al aprendizaje del ámbito de identificadores a partir del marco educativo CIF para tareas de análisis (Serrano-Cámara *et al.*, 2016). Se evaluó su efecto motivador bajo cuatro condiciones, obteniéndose entre otros resultados que el uso conjunto de CIF y MoCAS aumentaba significativamente la motivación intrínseca, la motivación extrínseca vía regulación identificada, pero también la desmotivación (Serrano-Cámara *et al.*, 2014).

3.6 Gamificación

A nivel universitario se han desarrollado y utilizado sistemas de enseñanza de la programación con juegos interactivos adaptados a los gustos de los estudiantes (Hijón-Neira *et al.*, 2014a; Hijón-Neira *et al.*, 2014b) con resultados muy positivos, integrándolos de forma lúdica en Moodle para su evaluación (Hijón-Neira *et al.*, 2014c). En los últimos años el empleo de mecánicas típicas de los juegos en entornos no lúdicos se ha incrementado debido al efecto motivador y de enganche que producen. En nuestro caso se diseñó el *plug-in* GameMo que permitía la integración de diversos elementos propios de la gamificación en la plataforma Moodle y por lo tanto de una manera sencilla facilitar a los profesores la integración de la

gamificación en su curso. GameMo ha sido probado con éxito y analizado para realizar futuras mejoras (García-Iruela e Hijón-Neira, 2019). Actualmente se continúa esta línea de investigación analizando el efecto de la gamificación en el enganche, la satisfacción y la motivación de los alumnos.

3.7 Aprendizaje basado en vídeos

El aprendizaje basado en vídeos trata de usarlos como medio para mejorar la experiencia educativa de los estudiantes. La plataforma ClipIt (Martín *et al.*, 2015a) permite a los estudiantes ser los creadores de sus propios vídeos y aprender de los vídeos realizados por otros compañeros, así como de las críticas constructivas que reciben de iguales a través de la plataforma. ClipIt obtuvo el tercer premio a la mejor herramienta educativa en el EC-TEL 2014 (Llinás *et al.*, 2014). Se han realizado distintas experiencias en diversas asignaturas donde se ha analizado varias métricas para medir la motivación interna de los grupos (Castellanos *et al.*, 2017) y estudios sobre los criterios de recomendación a usar en este tipo de redes sociales educativas (Martín *et al.*, 2015b).

4. Trabajos futuros

Pueden destacar los trabajos en curso o previstos en varias líneas de investigación. Respecto al desarrollo de la competencia digital docente, existen planes ligados a la reciente aprobación de un nuevo Máster Oficial que se impartirá a partir del curso 2020/2021 en la Universidad Rey Juan Carlos. En esta línea de trabajo se profundizará en la definición detallada de las diferentes dimensiones de la competencia digital. También se pretende acometer formas de comprobar de forma sistemática el alineamiento entre objetivos y resultados de aprendizaje de aprendizaje.

Para la enseñanza de la introducción a la programación se está preparando una nueva versión de PrimaryCode en la que además de los arrays, se incorporen la enseñanza de nuevos conceptos como la recursividad y los accesos a ficheros. Por otro lado, se está desarrollando una aplicación que se puede incluir dentro de la categoría de tutores/correctores. Se trata de una herramienta web para el aprendizaje de expresiones lógicas, dentro de la electrónica digital. La aplicación ha sido evaluada por el alumnado de dos grupos de 3º y otros dos grupos de 4º de ESO. El

cuestionario de usabilidad aplicado tanto a alumnos como a profesores ofreció buenos resultados.

En el área de visualización del software, se sigue investigando en dos líneas: visualización del análisis dirigido por sintaxis y visualización de algunos casos particulares de la recursividad. En este último caso, se busca flexibilizar una aproximación genérica a la visualización de la recursividad con una visualización eficaz de ciertas clases de algoritmos, como son los algoritmos de divide y vencerás o de ramificación y poda. Asimismo, se prevé desarrollar una nueva versión del sistema OptimEx que proporcione mayor flexibilidad en el proceso experimental, recogida más precisa de tiempos de ejecución y mayor soporte a la generación de documentación sobre el proceso y los resultados experimentales.

Por último, dos líneas de Informática Educativa buscan la potenciación de la investigación en nuevas tecnologías. Por un lado, se prevé estudiar el uso combinado de MOOCs y redes sociales digitales como comunidades de aprendizaje, como nueva forma de mejorar el compromiso de los estudiantes con el curso. Por otro lado, se está usando la realidad aumentada en aplicaciones para dispositivos móviles y orientadas a diferentes niveles educativos, con el objetivo de fomentar la enseñanza de conceptos básicos de programación a través de juegos, el pensamiento computacional o la gestión de las emociones.

Agradecimientos

Este trabajo se ha financiado con diversos proyectos de investigación: AlgoTools (TIN2011-29542-C02-01) del Ministerio de Ciencia e Innovación, iProg (TIN2015-66731-C2-1-R) del Ministerio de Economía y Competitividad, y e-Madrid (S2013/ICE-2715) y e-Madrid-CM (P2018/TCS-4307) de la Comunidad Autónoma de Madrid (el proyecto e-Madrid-CM también está financiado con fondos estructurales FSE y FEDER), BlueThinking de la Fundación Orange y e-Integra (TIN2013-44586-R) del Ministerio de Ciencia e Innovación.

Referencias

Becerra-Jimenez, D. Clemente-Sánchez, S. Cáceres-Taladriz, C. y Rosado-Martín, S. URJXC, el

conocimiento abierto como herramienta de transformación digital en las universidades. *Innovación y transformación digital: estrategias y metodologías docentes en Educación Superior*. Dykinson; 2018. p. 119-134. ISBN 978-84-9148-907-8.

Borrás-Gené, O. Martínez-Núñez, M. y Fidalgo-Blanco, Á. (2016). New challenges for the motivation and learning in engineering education using gamification in MOOC. *International Journal of Engineering Education*. 32(1B), 501–512.

Cáceres-Taladriz, C. (2018). Flipped Learning y otras metodologías docentes atractivas. *Innovación y transformación digital: estrategias y metodologías docentes en Educación Superior*. Ed. Dykinson. ISBN 978-84-9148-907-8, págs. 87-98.

Castellanos, J., Haya, P. A. Urquiza-Fuentes, J. (2017) A novel group engagement score for virtual learning environments. *IEEE Transactions on Learning Technologies* 10(3), 306-317 doi: 10.1109/TLT.2016.2582164

Debdi, O., Paredes-Velasco, M. y Velázquez-Iturbide, J. Á. (2015). GreedExCol, a CSCL tool for experimenting with greedy algorithms. *Computer Applications in Engineering Education*, 23(5), 790-804. doi: 10.1002/cae.21655

Esteban-Sánchez, N., Pizarro, C. y Velázquez-Iturbide, J. Á. (2014) Evaluation of a didactic method for the active learning of greedy algorithms. *IEEE Transactions on Education*, 57(2), 83-91. doi: 10.1109/TE.2013.2275154

García-Iruela, M. Fonseca, M. J. Hijón-Neira y R. Chambel, T. (2019). Analysis of gamification elements. A case study in a computer science course, 20th International Conference on Artificial Intelligence in Education, AIED 2019, 89-93. doi: 10.1007/978-3-030-23207-8_17

García-Iruela, M. Hijón-Neira, R. (2018). How gamification impacts on vocational training students, 19th International Conference on Artificial Intelligence in Education, AIED 2018, 99-103. doi:

10.1007/978-3-319-93846-2_19

Hijón-Neira, R., Velázquez-Iturbide, C., Pizarro-Romero, C., Carriço, L. (2014a). Serious games for motivating into programming. IEEE Frontiers in Education Conference (FIE), 757-763. doi: 10.1109/fie.2014.7044111.

Hijón-Neira, R., Velázquez-Iturbide, C., Pizarro-Romero, C., Carriço, L. (2014b). Merlin-know, an interactive virtual teacher for improving learning in Moodle. IEEE Frontiers in Education Conference (FIE), 19-26. doi: 10.1109/fie.2014.7043990

Hijón-Neira, R., Velázquez-Iturbide, Á., Pizarro-Romero, C., Carriço L. (2014c). Game programming for improving learning experience. ITICSE 2014 - Proceedings of the 2014 Innovation and Technology in Computer Science Education Conference, 225-230. doi: 10.1145/2591708.2591737

Lázaro-Carrascosa, C., Palacios Alonso, D. y Velázquez-Iturbide, A. (2019). Combinando Flipped Classroom y Puzzle de Aronson en el Máster del Profesorado. International Symposium on Computers in Education (SIIE, 2019). Comunicación aceptada.

Llana, L., Martín-Martín, E., Pareja-Flores, C. y Velázquez-Iturbide, J. Á. (2014). FLOP: A user-friendly system for automated program assessment. Journal of Universal Computer Science, 20(9), 1.304-1.326

Llinás, P., Haya, P. A., Gutiérrez, M. A., Martín, E., Castellanos, J., Hernán, I., Urquiza-Fuentes, J. (2014) Proceedings of the 9th European Conference on Technology Enhanced Learning (EC-TEL 2014). Rensing, C., De Freitas, S., Ley, T., Muñoz-Merino, P.J. (eds.) Open Learning and Teaching in Educational Communities, LNCS vol. 8719, pp 502-505. DOI:10.1007/978-3-319-11200-8_51

Martín, E., Cupeiro, C., Pizarro, L., Roldán-Álvarez, D., Montero-de-Espinosa, G. (2018). "Today I tell" A comics and story creation app for people with autism spectrum condition. International Journal of Human-Computer Interaction, 35(8), 679-601. doi: 10.1080/10447318.2018.1550178

Martín, E., Gertrúdix, M., Urquiza-Fuentes, J., Haya, P. A. (2015a) Student activity and profile datasets from an online video-based collaborative learning experience. British Journal of Educational Technology, 46(5), 993-998. doi: 10.1111/bjet.12318

Martín, E., Hernán-Losada, I., Haya, P.A. (2015b) Comparing social factors affecting recommender decisions in online and educational social network, New Review of Hypermedia and Multimedia, 22:1-2, 6-26. doi: 10.1080/13614568.2015.1027966.

Martín, E., Roldán-Álvarez, D., Haya, P. A., Fernández-Gaullés, C., Guzmán, C., Quintanar, H. (2019). Impact of using interactive devices in Spanish early childhood education public schools. Journal of Computer Assisted Learning, 35(1), 1-12

Masapanta-Carrión, S. y Velázquez-Iturbide, J. Á. (2018). A systematic review of the use of Bloom's taxonomy in computer science education. Proceedings of the 49th ACM Technical Symposium on Computing Science Education, SIGCSE 2018, 441-446. doi: 10.1145/3159450.3159491

Masapanta-Carrión, S. y Velázquez-Iturbide, J. Á. (2019). Evaluating instructors' classification of programming exercises using the revised Bloom's taxonomy. Proceedings of the 24th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2019, 541-547, doi: 10.1145/3304221.3319748

Nicolalde-Rodríguez, D., Urquiza-Fuentes, J (2018). Educational Impact of Syntax Directed Translation Visualization, a Preliminary Study. Proceedings of the 2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). DOI: 10.1109/VLHCC.2018.8506537

Pérez-Marín, D., Hijón-Neira, R., Babelo, A., Pizarro, C. (2018). Can computational thinking be improved by using a methodology based on metaphors and scratch to teach computer programming to children? Computers in Human Behaviour, <https://doi.org/10.1016/j.chb.2018.12.027>

Pérez-Marín, D., Hijón-Neira, R., Romero, A., Cruz, S. (2019). is the use of makekey makekey helpful to teach programming concepts to primary education students? *International Journal of Online Pedagogy and Course Design (IJOPCD)*. IGI-Global 9(2), 63-67.

Roldán-Álvarez, D., Martín, E., García-Herranz, M., Haya, P. A. (2016). Mind the gap: Impact on learnability of user interface design of authoring tools for teachers. *International Journal of Human-Computer Studies*, 94, 18-34. DOI: 10.1016/j.ijhcs.2016.04.011.

Roldán-Álvarez, D., Martín, E., Haya, P. A., García-Herranz, M., Rodríguez-González, M. (2018). DEDOS: An authoring toolkit to create educational multimedia activities for multiple devices. *IEEE Transactions on Learning Technologies*, 11(4), 493-505. doi: 10.1109/TLT.2017.2788867

Rubio-Sánchez, M., Kinnunen, P., Pareja-Flores, C. y Velázquez-Iturbide, J. Á. (2014). Students perception and usage of an automated programming assessment tool. *Computers in Human Behavior*, 31, 453-460. doi: 10.1016/j.chb.2013.04.001

Santacruz Valencia, L., Pérez Marín, D., Gómez Gómez, M. y Hijón Neira, R. (2013). ApS como metodología de transferencia de aprendizajes en TIC de la Universidad a los centros escolares. *Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación (TE & ET)*, 11, 46-53. Recuperado de <http://sedici.unlp.edu.ar/handle/10915/32352>

Serrano-Cámara, L. M., Paredes-Velasco, M., Alcover, C. M^a y Velázquez-Iturbide, J. Á. (2014). An evaluation of students' motivation in computer-supported collaborative learning of programming concepts. *Computers in Human Behavior*, 31, 499-508. doi: 10.1016/j.chb.2013.04.030

Serrano-Cámara, L. M., Paredes-Velasco, M., Velázquez-Iturbide, J. Á., Alcover, C. M^a y Castellanos, M^a. E. (2016). MoCAS: A mobile collaborative tool for learning scope of identifiers in programming courses. *International Journal of Engineering Education*, 32(2B), 969-981

Urquiza-Fuentes, J. (2019). Experiencia de clase invertida en la asignatura de Procesadores de Lenguajes. *Competencia Digital Docente: Una perspectiva de futuro en la Educación Superior*. En imprenta.

Urquiza-Fuentes, J., Velázquez-Iturbide, J. Á. (2013). Towards the Effective Use of Educational Program Animations: the Roles of Student's Engagement and Topic Complexity. *Computers & Education*, 67, 178-192. DOI: 10.1016/j.compedu.2013.02.013

Urquiza-Fuentes, J., Almeida-Martínez, F. J., Pérez-Carrasco, A., Velázquez-Iturbide, J. Á. (2013). Improving students' performance with visualization of error recovery strategies in syntax analysis. *Journal of Research and Practice in Information Technology* 45(3), 237-250.

Urquiza-Fuentes, J., Paredes-Velasco, M. (2017). Investigating the effect of realistic projects on students' motivation, the case of human-computer interaction course. *Computers in Human Behavior* 72: 692-700. DOI: 10.1016/j.chb.2016.07.020

Velázquez-Iturbide, J. Á. (2013a). An experimental method for the active learning of greedy algorithms. *ACM Transactions on Computing Education*, 13(4), artículo 18. doi: 10.1145/2534972

Velázquez-Iturbide, J. Á. (2013b). Identification and removal of misconceptions on optimization concepts underlying greedy algorithms. *Journal of Research and Practice in Information Technology*, 45(3/4), 203-217

Velázquez-Iturbide, J. Á. (2013c). Using textbook illustrations to extract design principles for algorithm visualizations. En W. Huang (Ed.), *Handbook of Human Centric Visualization*, Springer Science+Business Media, 227-249. doi: 10.1007/978-1-4614-7485-2_9

Velázquez-Iturbide, J. Á. (2016a). A «multiple executions» technique of visualization. *Proceedings of the 21st Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2016*, 59-64. doi: 10.1145/2899415.2899451

Velázquez-Iturbide, J. Á. (2016b). GreedEx and OptimEx: Two tools to experiment with optimization

algorithms. *International Journal of Engineering Education*, 32(3A), 1.097-1.106

Velázquez-Iturbide, J. Á. (2019). Students' misconceptions of optimization algorithms. *Proceedings of the 24th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2019*, 464-470. doi: 10.1145/3304221.3319749

Velázquez-Iturbide, J. Á. et al. (2012). LITE (Laboratory of Information Technologies in Education), URJC. *IE Comunicaciones*, 15, 103-112.

Velázquez-Iturbide, J. Á. et al. (2018). Informe del grupo de trabajo SCIE/CODDII sobre la enseñanza preuniversitaria de la informática: SCIE. Recuperado de <http://www.scie.es/informatica-ensenanza-no-universitaria/>

Velázquez-Iturbide, J. Á., Castellanos, M^a. E. e Hijón-Neira, R. (2016). Recursion removal as an instructional method to enhance the understanding of recursion tracing. *IEEE Transactions on Education*, 59(3), 161-168. doi: 10.1109/TE.2015.2468682

Velázquez-Iturbide, J. Á., Debdi, D. y Paredes-Velasco, M. (2015). A review of teaching and learning through practice of optimization algorithms. En R. Queirós (Ed.), *Innovative Teaching Strategies and New Learning Paradigms in Computer Programming* (pp. 65-87). IGI Global.

Velázquez-Iturbide, J. Á., Debdi, D., Esteban-Sánchez, N. y Pizarro, C. (2013). GreedEx: A visualization tool for experimentation and discovery learning of greedy algorithms. *IEEE Transactions on Learning Technologies*, 6(2), 130-143. doi: 10.1109/TLT.2013.8

Velázquez-Iturbide, J. Á., Hernán-Losada, I. y Paredes-Velasco, M. (2017). Evaluating the effect of program visualization on student motivation. *IEEE Transactions on Education*, 60(3), 238-245. doi: 10.1109/TE.2017.2648781

Velázquez-Iturbide, J. Á. y Pérez-Carrasco, A. (2017). Using the SRec visualization system to construct dynamic programming algorithms. *International Journal of Engineering Education*, 33(2B), 908-917