

***SUCRE4Kids*: El fomento del pensamiento computacional a través de la interacción social y tangible**

Sergio Trilles
Institute of New Imaging Technologies
Universitat Jaume I
Castellón
strilles@uji.es

Carlos Granell
Institute of New Imaging Technologies.
Universitat Jaume I
Castellón
carlos.granell@uji.es

Resumen

Esta comunicación presenta las primeras experiencias del proyecto *SUCRE4Kids* para el fomento de las vocaciones científicas, la promoción del pensamiento computacional y la programación en alumnos de secundaria, ciclos formativos y bachillerato. *SUCRE4Kids* combina el uso de dispositivos inteligentes y componentes electrónicos con la programación visual para la realización de proyectos reales. Lejos de ser una actividad individual, como en la mayoría de iniciativas educativas relacionadas con el aprendizaje de la programación, la novedad estriba en la introducción de aspectos sociales y de interacción tangible en las sesiones con estudiantes. Nuestras observaciones con más de 200 estudiantes revelan que transformar el aprendizaje de la programación en una actividad social y tangible, capaz de captar la curiosidad y atención de los estudiantes, parece un camino a seguir para atraer potenciales estudiantes hacia los estudios de informática o relacionados con competencias STEM.

Abstract

This paper presents the first experiences of the *SUCRE4Kids* project for fostering scientific vocation, the promotion of computational thinking and coding skills in high school students. *SUCRE4Kids* combines smart devices and electronic components with visual block programming for real projects. Far from being an individual activity, as in almost all educational initiatives related to code learning, the novelty resides in the introduction of social aspects and tangible interaction in the sessions with students. Our observations with more than 200 students reveal that transforming the way to learn programming into a social and tangible activity, it can capture the curiosity and attention of students, and it seems a way to follow in order to attract students to computer science or studies related to STEM competencies.

Palabras clave

Pensamiento computacional, vocación científica, promoción programación, interacción social y tangible.

1. Introducción

En un reciente informe sobre la situación de los estudios en informática en Europa [8], España no sale bien parada. En ese informe, se comparan la mayoría de países europeos en cuanto a número de estudiantes en carreras relacionadas con la ciencias de la computación e informática (incluyendo grado de desarrollo de videojuegos. Listado completo en [8, p. 10]), o el porcentaje de mujeres que cursan dichos estudios y grados, entre otros indicadores estadísticos. En estos dos ejemplos, se observa una tendencia decreciente sostenida para los años analizados (2011-2016), lo cual resulta especialmente preocupante si tenemos en cuenta que el ratio de estudiantes en informática por millón de habitantes es de por sí bajo en nuestro país [8]. Lamentablemente, esta situación no ayuda en absoluto a la promoción de los estudios de informática entre estudiantes preuniversitarios, que es el foco del presente artículo.

El informe anterior deja entrever que la demanda y la oferta de profesionales en estudios relacionados con la informática no se ajusta, sobrepasando de largo la demanda a la oferta existente. Lejos de reducirse, ese desajuste se estima que irá en aumento, con el potencial riesgo de convertir la informática en Europa en una profesión de “cuello de botella” en los próximos años. Para 2025, se estima un incremento del 13 % en la demanda de profesionales relacionados con competencias STEM (ciencia, tecnología, ingeniería y matemáticas) [4]. Contrariamente, según la OCDE (Organización para la Cooperación y el Desarrollo Económicos), los estudiantes de secundaria de hoy - que serán los profesionales de mañana - presentan destrezas desiguales en sus niveles de conocimiento en Ciencias y Matemáticas, ocupando España, posiciones medias-

bajas en sus resultados.

Ante este panorama de futuro de nuestra profesión nada halagüeña, urge más que nunca la promoción de los estudios de informática entre los estudiantes pre-universitarios. Es necesario buscar incentivos y vías alternativas que permitan fomentar el pensamiento computacional, la programación y, en última instancia, la vocación científica entre los más jóvenes. Reflejo de esta preocupación, es el nuevo rumbo tomado en algunos países como Reino Unido [17], donde los principios de la programación forman parte del currículo docente desde edades tempranas.

El proyecto *SUCRE4Kids* [15] tiene como objetivo principal el fomento de las vocaciones científicas, y la promoción del pensamiento computacional y la programación en alumnos de secundaria, ciclos formativos y bachillerato. Aparte de los elementos científico-técnicos que definen el proyecto *SUCRE4Kids* (sección 2), la novedad que destacamos en este artículo es la forma en que se ha acercado y presentado *SUCRE4Kids* a los estudiantes, por medio de intervenciones que enfatizan interacciones tangibles y sociales entre los estudiantes. Nuestras observaciones con 206 estudiantes a lo largo de dos años revelan que transformar el aprendizaje de la programación en una actividad social y tangible, capaz de captar la curiosidad y atención de los estudiantes, parece un claro camino a seguir para atraer potenciales estudiantes hacia estudios relacionados con la informática y/o competencias STEM.

Tras la descripción técnica del proyecto *SUCRE4Kids* (sección 2), describimos la metodología de las sesiones (sección 3). Seguidamente, la sección 4 discute los resultados obtenidos, y se enumeran las conclusiones finales y trabajo futuro en la sección 5.

2. Enfoque *SUCRE4Kids*

SUCRE4Kids combina el uso de dispositivos inteligentes y componentes electrónicos con la programación visual por bloques para la realización de proyectos reales. En esta sección describimos los objetivos didácticos, elementos técnicos y materiales que lo forman.

2.1. Aproximación pedagógica

Desde el punto de vista didáctico, *SUCRE4Kids* promueve dos conceptos para alimentar la curiosidad por la programación. El primero de ellos es familiarizar a los alumnos con la tecnología relacionada con dispositivos electrónicos y con el movimiento “maker”. El bajo coste de estos dispositivos electrónicos, unido a la creatividad y el afán de compartir conocimiento, que es una característica de la cultura maker, está democratizando el uso creativo de esta tecnología para fines comerciales y personales, permitiendo que cualquier per-

sona pueda diseñar y fabricar sus propios dispositivos inteligentes. De forma similar a Cubelets¹, MOSS² o LittleBits³, se da importancia a lo tangible, para que los alumnos puedan tocar y conectar, despertando un interés y conexión con el objetivo final, aprender a programar en definitiva [3].

El segundo concepto se enmarca en las competencias y habilidades propias del pensamiento computacional [18]. El pensamiento computacional engloba las habilidades necesarias para la resolución de problemas mediante competencias como la descomposición, diseño de algoritmos y razonamiento lógico. Los recientes y continuos avances tecnológicos conllevan irremediablemente un cambio en las competencias y habilidades necesarias para el desarrollo de muchas profesiones. Los estudiantes, así como también los profesores [19], deberán conocer y afianzar estas nuevas competencias y habilidades computacionales ya que, de una u otra forma, estarán presentes en las profesiones del futuro que aún están por imaginar.

Cualquier actividad educativa, ya sea basada en tecnología o no, debe diseñarse en concordancia a cómo las personas aprenden. La creación y la experimentación en el aprendizaje definen el constructivismo [1]. Pero si además experimentamos con cosas que son significativas para nosotros, el aprendizaje resulta mucho más placentero. Precisamente, la parte importante del constructivismo está conectada con el interés y las pasiones del estudiante. Con *SUCRE4Kids*, intentamos que los estudiantes experimenten, que “aprendan haciendo” [14], fomentando que las sesiones sean abiertas, participativas, y basadas en el trabajo colectivo. Pero sobre todo buscamos que *SUCRE4Kids* estimule su interés y agrado en el aprendizaje de la programación cuando lo que su programa, lo que se aprende haciendo, es un proyecto que realmente les importa.

2.2. Lo tangible: el maletín *SUCREKit*

El maletín de invención *SUCREKit* (Figura 1) contiene diferentes componentes electrónicos, necesarios para el desarrollo de una serie de proyectos propuestos (sección 2.4), aunque los estudiantes también pueden realizar proyectos que ellos propongan.

La principal diferencia de *SUCREKit* frente otros kits disponibles, es ofrecer una plataforma abierta a nivel de hardware y software, que sea comprensible para profesores y alumnos, y de fácil aplicación a diferentes unidades didácticas independientemente del nivel que tengan los alumnos.

El maletín incluye componentes electrónicos que recrean la partes funcionales de cualquier ordenador: un

¹<https://www.modrobotics.com/cubelets/>

²<https://www.modrobotics.com/moss/>

³<https://littlebits.cc/>

núcleo y dispositivos de entrada y salida (E/S). El núcleo (microcontrolador y buses) está formado por un Arduino y un shield, llamado Grove, que facilita las conexiones de dispositivos de E/S al núcleo. Los sensores actúan de dispositivos de entrada de datos, y los actuadores de salida. Cabe destacar que las E/S pueden ser digitales o analógicas. Concretamente, el maletín *SUCREKit* viene con el siguiente material:



Figura 1: Maletín de invención *SUCREKit*

- Núcleo: 1 Arduino y 1 *Shield Grove*.
- Entrada: 4 sensores (sensor humedad, sonido, luz y proximidad).
- Salida: 2 actuadores (led multicolor y barra de leds).
- Otros: 4 chapas con roles, 9 fichas descriptivas (4 de los sensores, 2 de los actuadores y 3 de los proyectos propuestos) y un cable USB tipo-B.

2.3. Programación visual por bloques

Los bloques, de distinta forma y color, representan conceptos de programación, desde la definición de una variable, condiciones o funciones. De forma interactiva, los bloques se van encajando visualmente, de forma analógica a como se escribe código estándar (anidamiento de instrucciones, etc.), hasta que la estructura final de bloques permite resolver un problema. Algunos destacan el valor de la programación visual en bloques como pasarela a la programación tradicional basada en texto [12].

Existen varias iniciativas relacionadas con la programación visual basadas en bloques [2], como la veterana Scratch [10], MIT App Inventor [9], y el proyecto de código abierto de Google Blockly [6]. En *SUCRE4Kids*, hemos desarrollado una herramienta web de programación visual⁴ a partir de Blockly, creando bloques nuevos para cada uno de los sensores y actuadores incluidos en el maletín (Figura 2).

⁴<http://elcano.init.uji.es/sucrer4kids/>

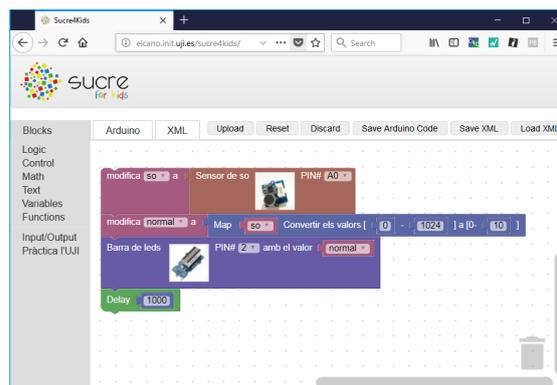


Figura 2: Herramienta de programación visual. Bloques marrones son sensores, bloques lila actuadores.

2.4. Proyectos como objetivo

El objetivo de las sesiones (sección 3) fue que los estudiantes realizaran diferentes proyectos, programando visualmente el circuito ensamblado con los componentes del maletín. Los proyectos propuestos fueron:

- Proyecto “Planta inteligente”: el objetivo es conocer si debemos regar una planta. Mediante un sensor de humedad del suelo para medir la humedad y una barra de LEDs para indicar el nivel.
- Proyecto “Parking inteligente”: recrea el sistema de plazas libres en aparcamientos públicos o de plazas libres en centros comerciales. Mediante un sensor de proximidad, se comprueba si hay un objeto (p.e. coche miniatura) en un sitio (estacionamiento). Se enciende o no un led si la plaza de estacionamiento está ocupada o libre, respectivamente.
- Proyecto “Luz inteligente”: simula una luz inteligente, comprobando el nivel de intensidad de luz en el aula, mediante un sensor lumínico e iluminando un LED cuanto menor es la luz ambiental.
- Proyecto libre: los alumnos tienen libertad para proponer cualquier proyecto que deseen realizar.

3. ¿Qué hemos realizado?

Hemos experimentado con *SUCRE4Kids* durante el programa *Pràctica a l’UJI* para el fomento del grado de informática de la Universitat Jaume I y la captación de futuros alumnos procedentes de centros de formación de secundaria y ciclos formativos. Cada jornada comenzaba con una charla inicial del director del grado a todos los alumnos, para introducirles las ventajas del grado de informática así como las potenciales salidas profesionales. Luego los estudiantes se dividían en dos itinerarios diferentes para la realización de sesiones prácticas.

Cada itinerario constaba de dos sesiones prácticas de

105 minutos con un descanso de 30 minutos entre sesiones. El primer itinerario lo formaban *Introducción a la programación de dispositivos Android: primeros pasos* y *Creación y programación de dispositivos electrónicos con Arduino*, donde experimentamos con *SUCRE4Kids*. En el segundo itinerario, las prácticas fueron: *Introducción a la programación de dispositivos Android: Cazarratones*, y *Aprender a programar con un lenguaje visual un robot móvil*.

En total se realizaron 6 jornadas, de dos sesiones prácticas cada una, durante enero de 2017 y 2018 (finalmente se realizaron 11 sesiones ya que se canceló una en 2018), por las que pasaron 206 alumnos de secundaria, bachillerato y módulos formativos (Cuadro 1). Las sesiones se desarrollan de igual manera independientemente del nivel educativo de los alumnos.

3.1. Protocolo de las sesiones

Dos profesores participaron en cada uno de las sesiones, pero con roles distintos. El profesor principal (PP, primer autor de la presente comunicación) llevó el peso de la presentación de los contenidos pedagógicos, la descripción y funcionamiento de los materiales y herramientas utilizadas, la explicación y planteamiento de los proyectos guiados y libres, y realizó tareas de apoyo a los grupos durante la realización de los proyectos. El profesor secundario (PS, segundo autor) realizó también tareas de apoyo a los grupos, pero se mantuvo más al margen para llevar a cabo tareas de observación encubierta [11] sin que los estudiantes ni su profesorado fueran advertidos previamente. Este método de investigación social se ajustaba a la naturaleza y condiciones de las sesiones prácticas. La observación encubierta consistió básicamente en tomar notas en papel, y esporádicamente material fotográfico (preservando la privacidad de los menores), sobre las reacciones y comportamiento de los estudiantes.

Antes de comenzar cada sesión, los profesores preguntaban a los homólogos responsables de los estudiantes sobre sus conocimientos previos en programación. Al final de la sesión, y sin estudiantes, se mantenía un breve intercambio informal de impresiones entre los PP/PS y el profesorado de instituto sobre los materiales docentes utilizados y la actitud e interés que habían despertado en los participantes.

3.2. Organización de las sesiones

Cada sesión de *SUCRE4Kids* se dividió en tres secciones diferentes (A-C), en las cuales introdujimos distintos tipos de intervenciones (Cuadro 2).

La sección A consistió en la introducción de contenidos por parte del profesor. Utilizando una presentación y proyector, el PP presentó los objetivos pedagógicos y motivación, los conceptos básicos de pro-

gramación visual, recurriendo al juego del laberinto de programación basado en Blockly⁵ para tareas prácticas. Durante esta sección, los participantes trabajaban de forma individual con un ordenador, siguiendo la presentación disponible mediante acceso al Aula Virtual (Moodle).

A continuación, el PP prosiguió con el bloque de Arduino. Aquí, los participantes formaron grupo de 4 personas, asignando un maletín *SUCREKit* por grupo. El PP describió cada uno de los componentes electrónicos del maletín: placa arduino, *shield Grove*, sensores y actuadores. Los participantes se descargan la librería necesaria localmente y la importa en el Arduino IDE. Finalmente, el PP presentó la herramienta de programación visual y describió cada uno de los bloques de programación relevantes (condicionales, creación y modificación variables, función de mapeo, bloques específicos de sensores y actuadores).

En la sección B se llevó a cabo un proyecto guiado. Primero, cada grupo realizó un proyecto “hola mundo”: encender y apagar un led. Luego, cada grupo realizó el proyecto guiado de una planta inteligente (sección 2.4), que involucraba la conexión de un sensor de humedad de tierra y un actuador de barra de LEDs al Arduino, junto con la programación necesaria para que el circuito funcionara correctamente (Figura 3).

Finalmente, en la sección C, cada grupo realizó un proyecto libre a escoger de entre las propuestas (sección 2.4).



Figura 3: Montaje del proyecto Planta Inteligente por los estudiantes.

El Cuadro 2 resume el tipo de intervenciones y la forma de trabajo utilizadas durante las sesiones. *SUCRE4Kids* nació por una apuesta por lo tangible; los participantes interactúan con los sensores, los conectan físicamente y les dan vida mediante la programación visual. A diferencia de otras iniciativas de programación, centradas en el trabajo individual enfrente del ordenador, la interacción tangible en grupo define *SUCRE4Kids*. Sin embargo, la gran novedad introducida en las sesiones de 2018 es la asignación de roles explícita. Inspirados por el trabajo de codificación

⁵<https://blockly-games.appspot.com/>

Año-sesión	Duración (min)	Num. alumnos	M	F	Perfil alumnos
2017-01	105	20	NA	NA	ciclos formativos - itinerario informática
2017-02	105	22	NA	NA	1 y 2 ESO - cursan informática como optativa
2017-03	95	13	NA	NA	1 y 2 bachiller - cursan informática como optativa
2017-04	95	6	NA	NA	1 bachiller - cursan informática como optativa
2017-05	95	25	NA	NA	1 y 2 bachiller y 4 ESO - cursan informática como optativa
2017-06	90	22	NA	NA	1 bachiller - conocimiento limitados
2018-01	95	20	18	2	2 bachiller - cursan informática como optativa
2018-02	85	19	13	6	4 ESO - cursan informática como optativa
2018-03	90	19	18	1	ciclos formativos - itinerario informática
2018-04	85	19	15	4	1 bachiller - cursan informática como optativa
2018-05	90	21	15	4	1 bachiller - cursan informática como optativa

Cuadro 1: Lista sesiones. Total alumnos: 206; Media alumnos por sesión: 19; Duración media sesión (minutos): 94

Sección	2017		2018	
	Tangible	Social	Tangible	Social
A	Sí	indiv. + grupo	Sí	indiv. + grupo
B	Sí	grupo	Sí	grupo + roles
C	Sí	grupo	Sí	grupo + roles

Cuadro 2: Tipo de intervenciones.

social [16], donde los autores explotan la programación en parejas con roles distintos, nosotros asignamos roles mediante chapas de colores (Figura 1). Sin embargo, definimos roles más generales para cada miembro de un grupo: programador, hardware, coordinador y analista. Como veremos en la sección siguiente (sección 4), con la designación explícita de roles reforzamos considerablemente la interacción social en el trabajo en grupo.

4. ¿Qué hemos aprendido?

4.1. Resultados cuantitativos

En las 11 sesiones realizadas (6 durante 2017 y 5 durante 2018), la sesión mas larga fue de 105 minutos (el máximo establecido) y la más corta de 85 minutos. La duración media fue de 95 minutos. La Figura 4 muestra la distribución porcentual de los tres tipos de secciones para cada sesión. Salvo alguna sesión anómala, el resto sigue un patrón similar: la sección A ocupa aproximadamente el 60 % de la sesión, mientras que las secciones B y C se distribuyen el 40 % restante.

En particular, el tiempo medio dedicado a la sección A fue de 55 minutos, siendo 70 minutos el tiempo máximo dedicado y 45 minutos la sesión con menor dedicación. La parte B, el proyecto guiado, tuvo una duración media de 23 minutos, siendo 40 minutos la sesión con más dedicación y 15 minutos la que menos. Finalmente, a la sección C, el proyecto libre, se dedicaron 15 minutos de media, siendo 25 minutos la sesión con mayor dedicación y 0 minutos en la que menos, debido en este caso a que la mitad de los alumnos llegaron con

retraso a la práctica.

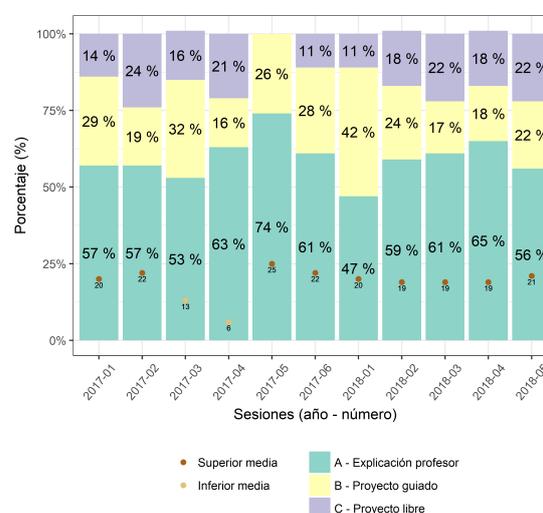


Figura 4: Distribución temporal de las secciones A-C por sesión. Puntos representan número asistentes por sesión (media=19).

Podría pensarse que a menor número de alumnos, las secciones B y C ganarían en importancia. Sin embargo, no se percibe relación alguna en esta dirección, es decir, entre el número de asistentes y la distribución temporal de las secciones. Otros factores como el conocimiento previo de los estudiantes o la variabilidad de los grupos (en cuanto a número de miembro), pueden influir también en la durabilidad de cada sección en una sesión.

4.2. Resultados cualitativos

4.2.1. Observaciones durante explicación profesores (A)

Durante esta sección, se llevaron a cabo dos tipos de actividades principalmente. La primera fue una breve introducción a la programación visual (Blockly) con el juego del laberinto, de forma individual. La segunda

fue una introducción al microcontrolador Arduino, en la que la forma de trabajo fue mixto, iniciando la actividad individualmente pero luego los estudiantes se organizaron en grupos sin definición de roles todavía.

Con respecto a la primera actividad, una observación recurrente en todas las sesiones es que los estudiantes absorbieron rápidamente los conceptos de programación visual, con independencia del grado de conocimientos en programación. Los grupos procedentes de ciclos formativos (2017-01, 2018-03) tuvieron mayores dificultades para llegar a un nivel avanzado del juego del laberinto (mayor nivel, mayor complejidad en cuanto a programación). Con los datos recogidos, no sabemos si esto se debió por falta de interés, por falta de conocimientos previos en programación de los estudiantes de ciclos formativos, o por pura casualidad. En general, parece que les resulta muy intuitivo apreciar la semántica de las construcciones de programación como bucles y/o condicionales a la hora de resolver los niveles de programación del juego del laberinto.

Sin casi explicación y de forma intuitiva, la programación visual y el concepto de codificar y probar de forma instantánea encaja perfectamente durante los primeros pasos durante el proceso de aprendizaje de la programación [12]. La programación visual basada en bloques engancha. En todas las sesiones, apreciamos que a la mitad de los estudiantes aproximadamente les costaba despejarse del juego del laberinto, incluso diez minutos después de haber pasado a la siguiente actividad (introducción al microcontrolador Arduino, sensores y actuadores). Algunos mantenían las dos pantallas a la vez, el juego del laberinto y la presentación sobre Arduino. El nivel extremo es el caso de dos estudiantes (sesión 2017-03) que, terminada la sesión, permanecieron “compitiendo” con resolver la programación de un nivel avanzado del laberinto mientras que el resto de compañeros ya había desalojado el aula.

Sin embargo, no observamos interacción entre los estudiantes. El trabajo individual enfrente del ordenador generaba un silencio casi total en la aula, salvo por breves comentarios entre estudiantes adyacentes para ayudarse entre ellos en la resolución de algún nivel del juego. Lo cual sugiere que, frente al trabajo individual, la programación en parejas podría funcionar de forma natural para potencial el pensamiento computacional y acelerar el aprendizaje de la programación [5].

Con respecto a la segunda actividad, se apreció que los estudiantes no habían visto o tocado Arduino y sensores antes (confirmado por profesores). Tenían respeto al principio y tocaban los componentes del maletín con cautela. Pero ganaron confianza pronto. Aunque estaban en grupo, la interacción no era muy profusa ni “ruidosa”. Cada uno experimentaba y exploraba por su cuenta los componentes del maletín, mientras seguían al mismo tiempo la explicación del PP.

La explicación por parte del PP resulta necesaria porque la inmensa mayoría de los estudiantes nunca habían tenido en sus manos este tipo de dispositivos electrónicos. Los estudiantes examinaban con atención los sensores y los conectaban con cuidado. Percibíamos que les importaba lo que tenían entre manos. Captaron los conceptos de analógico y digital pero había que recordarlo constantemente (PP y PS reiteraron esos conceptos a nivel del grupo) para evitar errores en la conexión de los sensores al Arduino, cuando intentaban conectar un sensor analógico a un pin digital del microcontrolador.

4.2.2. Observaciones durante proyecto guiado (B) y proyecto libre (C)

La asignación de roles creó excitación desde el principio de la actividad y definitivamente captó la atención de los estudiantes de forma decisiva para mantener el interés durante el resto de la sesión. Como el grupo más común era de cuatro personas, se diseñaron cuatro roles para que cada miembro del grupo desempeñara un rol distinto. Excepcionalmente, hubo algunos grupos de 3 estudiantes y un grupo de 5 a lo largo de las 11 sesiones. En el caso de grupos con 3 miembros, el rol de programador y el de analista recaían en la misma persona. En el caso del grupo de cinco, la persona de más asumía el rol de programador (programación en parejas). Se dejó a los alumnos el reparto de roles. No hubo conflictos a este respecto, por lo que PP/PS nunca asignaron explícitamente roles a los miembros del grupo.

En todas las sesiones del 2017 no se tuvo en cuenta la asignación explícita de roles, por lo que el trabajo en grupo y los roles variaron en función del número de integrantes del grupo. Cuando el grupo era de tres estudiantes, resultaba fácil la distribución (implícita) de roles para la ejecución colaborativa del proyecto. El participante que ocupaba la posición central del grupo se sentaba delante del ordenador, y se encargaba de la programación visual con la herramienta web (Figura 2). Los otros dos participantes se situaban uno a cada lado del programador. Uno se encarga de ensamblar los componentes electrónicos, mientras que el otro sostenía las fichas, y aportaba comentarios y detalles técnicos descritos en la fichas para el correcto funcionamiento del montaje y de la programación.

Cuando el grupo estaba formado por cuatro personas, lo cual era lo más habitual, la distribución espacial de los miembros ya no resultaba tan idónea como en el caso de los tres integrantes. Observamos reiteradamente a lo largo de las sesiones del 2017 que al menos un participante se mantenía ausente del trabajo en grupo, posiblemente por la imposibilidad de poder tocar nada y participar activamente en el grupo. También observamos que cuanto un participante tomaba las riendas

de un grupo, acaparaba todas las tareas y decisiones, relegando a los demás miembros a tareas marginales.

Por el contrario, la asignación explícita de roles cambió la forma de colaborar en los grupos en todas las sesiones del 2018. Las chapas arrancaron las primeras risas y caras de sorpresa, pero permitieron identificar y recalcar sin ambigüedad alguna la responsabilidad de cada uno de los miembros del grupo.

En consecuencia, la colaboración e interacción se disparó, hasta el punto de que algunos estudiantes se levantaban libremente y se acercaban a otros grupos para brindar su ayuda en relación al rol que desempeñaban. Es decir, interiorizaron su rol y eso repercutió en una mayor implicación e interés en las tareas que cada uno de los miembros del grupo realizaba. Todos colaboraban porque tenían un rol bien asignado [7]. Las chapas recuerdan continuamente a los miembros del grupo cual es el espacio de participación de cada uno en cuanto a las decisiones y tareas colaborativas de desempeñar. Incluso, algunos participantes llegaron a recriminarse (positivamente) si alguno fallaba o tardaba en realizar la tarea encomendada.

Durante la actividad del proyecto libre, la mayoría de los grupos (2018) mostraron un actitud más imaginativa a la hora de idear combinaciones de sensores y definir las construcciones (condicionales y bucles) asociadas, en comparación a los grupos de la primera edición (2017). Algunos grupos (todos en 2018) llegaron a configurar y programar circuitos compuestos de distintos tipos de sensores y actuadores, de mayor complejidad que la de los proyectos planteados por el profesorado.

5. Conclusiones y trabajo futuro

La actual comunicación detalla las experiencias con el proyecto *SUCRE4Kids*, el cual tiene como objetivo el fomento de la tecnología y la programación en edades donde hasta el momento no es lo habitual (educación primaria y secundaria) si analizamos los actuales currículos educativos. Tras la preparación de los materiales (maletín de invención *SUCREKit*) y herramientas necesarias (aplicación web de programación visual), la iniciativa fue aplicada durante 2 años alcanzando a más de 200 estudiantes de educación secundaria, en el marco de las jornadas *Pràctica l'UJI* para el fomento del grado de informática de la UJI.

En total se realizaron 11 sesiones programadas de 105 minutos, de los cuales 94 minutos fueron útiles de media. Las sesiones divididas en tres partes diferenciadas, una parte inicial de explicación, una parte para realizar un proyecto guiado y una parte para realizar un proyecto libre. Los tiempos medios en cada parte fueron 55, 23 y 15 minutos respectivamente.

Durante las sesiones se ha manifestado abiertamente

que palpar, tocar e interaccionar con cosas tangibles [3] enfatiza positivamente el compromiso de los alumnos en aprender a programar. La definición y asignación explícita de roles ha marcado claramente la responsabilidad y tarea de cada uno de los miembros del grupo durante el trabajo colaborativo para la consecución del proyecto común. Aunque parece un aspecto menor, en nuestras experiencias la asignación de roles ha resultado una intervención decisiva para promover la interacción social y enfatizar la responsabilidad en la realización de las tareas delegadas a cada miembro del grupo.

Con el objetivo de conocer de primera mano información cualitativa de los alumnos, queda pendiente recabar las opiniones directas de los estudiantes mediante cuestionarios, acompañado de entrevistas semi-estructuradas a alguno de ellos por cada grupo.

Mirando hacia delante, a continuación enumeramos una serie de opciones, a modo de hoja de ruta, para el proseguir del proyecto *SUCRE4Kids*.

- Somos consciente de la barrera existente en ofrecer, en el estado actual, *SUCRE4Kids* a estudiantes de edades inferiores a 12 años. Aunque el maletín facilita la conexión de los diferentes componentes a Arduino mediante el *Shield Grove*, los conceptos de analógico y digital todavía puede suponer una dificultad añadida el realizar las conexiones con escolares de primaria. Una línea de trabajo futuro es facilitar el encaje de sensores y actuadores en Arduino mediante por ejemplo conectores magnéticos, utilizados con éxito en otras iniciativas como *LitteBits* o *Grove Zero*⁶.
- Una de las líneas de mejora es el método de programación visual utilizada. Aún reconociendo las virtudes que tiene la programación visual basada en bloques para principiantes en la programación, nuestra intención es la exploración de otras formas de programación mixtas entre lo visual y basado en texto, y que primen la interacción en tiempo real entre la programación y los resultados obtenidos. En palabras de Bret Victor⁷, “los creadores necesitan una inmediata conexión con lo que están creando”. Es decir, si un desarrollador cambia algo en el código, los efectos de ese cambio tienen que verse inmediatamente. Somers relata magníficamente como la forma actual de programar viola este principio denominado “Inventing on Principle”[13]. El reto es mitigar este problema haciendo que la conexión entre el comportamiento de un sistema y su código sea inmediata. Algunas herramientas ya apuntan en esta dirección, como *Luna*⁸ y *Observable*⁹. Esto es una

⁶<https://bit.ly/2HpliVw>

⁷<http://worrydream.com>

⁸<http://www.luna-lang.org/>

⁹<https://observablehq.com>

línea a explorar en *SUCRE4Kids*, donde el efecto de conectar bloques de una forma u otra, se refleje inmediatamente en el editor para que el estudiante vea las consecuencias del cambio realizado.

- Otro de los rumbos en los que se quiere encaminar el proyecto, es pasar de una interacción y colaboración intra-grupal a una inter-grupal. Este nuevo enfoque parte de la idea en que todos los grupos dentro de una sesión colaboren y actúen sincronizados hacia un proyecto común. Esta interacción inter-grupal puede ser de colaboración o de competición. Para ello nuevos usos de la tecnología y conceptos de computación podrían abordarse, como el uso de dispositivos capaces de conectarse directamente a Internet (Internet de las Cosas) con soporte de conectividad WIFI (mediante router local) y dispositivos programables centrales que actúen de pantalla/panel de visualización para proporcionar feedback a los grupos sobre el grado de consecución del proyecto.
- La adopción de nuevas tecnologías y conceptos avanzados supone la posibilidad de proponer nuevos proyectos que hasta el momento no tenía cabida porque no podían desarrollarse técnicamente. Nuestra intención es la de diseñar nuevos proyectos basados en estas nuevas tecnologías pero al mismo tiempo que pongan en valor los tremendos desafíos a los que se enfrenta la sociedad actual como la privacidad, seguridad, cuestiones éticas y, en definitiva, la educación digital.
- Lamentablemente, la diferencia de género es abismal ya en bachiller (Cuadro 1), que se materializa en el escaso porcentaje de mujeres que cursan luego estudios de informática. Políticas urgentes son necesarias para contrarrestar este efecto adverso desde edades tempranas, en primaria, puesto que en secundaria parece ya demasiado tarde.

Agradecimientos

Sergio Trilles ha sido financiando por el programa postdoctoral Val+d de la Generalitat Valenciana (APOSTD/2016/058), y Carlos Granell por el programa Ramón y Cajal (RYC-2014-16913). El código fuente para la generación de los dos cuadros y la Figura 4 está disponible en <https://github.com/cgranell/jenui2018>

Referencias

- [1] Edith Ackermann. Piaget's constructivism, papert's constructionism: What's the difference. *Future of learning group*, 5(3):438, 2001.
- [2] David Bau, Jeff Gray, Caitlin Kelleher, Josh Sheldon, and Franklyn Turbak. Learnable programming: blocks and beyond. *Communications of the ACM*, 60(6):72–80, 2017.
- [3] Paulo Blikstein et al. Project bloks: designing a development platform for tangible programming for children. Technical report, Google, 2016.
- [4] R Castañeda Valle and C Rebolledo Gómez. Panorama de la educación: Indicadores de la OCDE. *Nota del País*, pages 1–11, 2013.
- [5] Jill Denner, Linda Werner, Shannon Campe, and Eloy Ortiz. Pair programming: Under what conditions is it advantageous for middle school students? *Journal of Research on Technology in Education*, 46(3):277–296, 2014.
- [6] Neil Fraser. Ten things we've learned from Blockly. In *Blocks and Beyond Workshop*, pages 49–50. IEEE, 2015.
- [7] Barbara Oakley et al. Best practices involving teamwork in the classroom: Results from a survey of 6435 engineering student respondents. *IEEE Trans. on Education*, 50(3):266–272, 2007.
- [8] Cristina Pereira and Svetlana Tikhonenko. Informatics Education in Europe: Institutions, degrees, students, positions, salaries. Key Data 2011–2016. 2017.
- [9] Shaileen Crawford Pokress and José Juan Dominguez Veiga. MIT app inventor: Enabling personal mobile computing. *arXiv preprint arXiv:1310.2830*, 2013.
- [10] Mitchel Resnick et al. Scratch: programming for all. *Comm. of the ACM*, 52(11):60–67, 2009.
- [11] Jane Ritchie, Jane Lewis, Carol McNaughton Nicholls, Rachel Ormston, et al. *Qualitative research practice: A guide for social science students and researchers*. Sage, 2013.
- [12] R Benjamin Shapiro and Matthew Ahrens. Beyond blocks: Syntax and semantics. *Communications of the ACM*, 59(5):39–41, 2016.
- [13] James Somers. The coming software apocalypse. *The Atlantic*, 9 2017.
- [14] Pedro Juan Sotorrio-Ruiz et al. Aproximación a la técnica “aprender haciendo” para la docencia en microprocesadores. *25 CUIEET*, 2017.
- [15] SUCRE. Sense your context and react for kids. <http://www.sucru.es/>, 2018.
- [16] Olivia L Tabel, Jonathan Jensen, Martin Dybdal, and Pernille Bjørn. Coding as a social and tangible activity. *Interactions*, 24(6):70–73, 2017.
- [17] J Velasco. Niños programadores: para qué sirve la enseñanza de programación en las escuelas. *El diario (Diario de Turing)*, 19, 2014.
- [18] Jeannette M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.
- [19] Aman Yadav, Chris Stephenson, and Hai Hong. Computational thinking for teacher education. *Communications of the ACM*, 60(4):55–62, 2017.